Unit - 5
Computer Assisted Audit Techniques
INTRODUCTION TO CAAT

LEARNING OBJECTIVES
The learning objectives of the CAAT module are:

- Understand how to use office automation software for performing various tasks as relevant to services provided by chartered accountant in areas of accounting, assurance and compliance.
- How to use CAAT/SQL queries for data analysis as required.
- How to review controls implemented at various levels/layers such as: Parameters, user creation, granting of access rights, input, processing and output controls in enterprise applications.

1.1 INTRODUCTION
Auditors deal with information in myriad ways encompassing the areas of accounting, assurance, consulting and compliance and most of this information is now available in electronic form. This is true not only in case of large and medium enterprises but even in small enterprises. In case there are enterprises who have still not adapted the digital way, then it is an opportunity for Auditors to help such enterprises to ride the digital wave. Hence, it has become critical for Auditors to understand and use information technology as relevant for the services we provide. It is rightly said: “one cannot audit data which is flying in bits and bytes by using the ancient method of riding on a horse back”. We are living in a knowledge era where the skill sets are keys to harnessing the power of technology to be effective as knowledge workers. Computer Assisted Audit Techniques (CAATs) refers to using technology for increasing the effectiveness and efficiency of auditing. CAATs enable auditors to do more with less and add value through the assurance process which is more robust and comprehensive. This chapter provides an overview of the process, approach and techniques which could be used across various technology platforms and in diverse enterprises.

1.2 THE ALL-ENCOMPASSING ELECTRONIC DATA
A great blessing in ancient times was: “May you live in exciting times”. Indeed, we are living in exciting times without even being aware of it. We are experiencing how technology innovations are making our life and living simpler by bridging global boundaries and bringing global information on our finger tips. For enterprises as well as professionals, the question is no longer what technology can do for us but what we can do with technology. The question “do I need to use technology” is no longer relevant instead the relevant decision is about “how do I use technology to remain relevant”.

Information technology is all pervasive and more so as the government and regulatory agencies also are using technology platform to provide services to citizens and compelling information to be filed in electronic form. The government at all levels has drawn up ambitious plans to implement e-Governance initiatives to improve speed, access and transparency of services. The Information Technology (IT) Act 2000 with IT Amendment Act 2008 and IT rules 2011 provide the regulatory framework and mechanism for recognizing electronic records and electronic transactions thereby facilitating ecommerce and also identifying cybercrimes and providing penalties and compensation for them. Hence, we can expect IT usage to only keep growing in the near future impacting all areas of life more so in our work as professionals.

1.3 AUDITORS AND CAATs

As auditors, we come across computers and communication technology as the most common denominator among our clients, both large and small. Further, we use computers and communication technology for providing services to our clients. In today’s complex and rapidly changing technology environment, it is important to master the right techniques which could be used across enterprises and across various technology platforms. Typical of a IT environment are the speed of processing, large capacity of storage, lack of the paper based trails, the radically different way of information processing, the ease of information access, internal controls being imbedded and the ever-present risk of failure of IT and loss of data. All these factors make it imperative for auditors to harness power of technology to audit technology environment by taking into consideration the risks, benefits and advantages. CAATs empower Auditors with the key survival techniques which effective used in any IT environment. CAATs are not specialist tools designed for use by specialist IT auditors but these are common techniques which can be easily mastered to audit in a computerized environment for statutory audit, tax audit and internal audit as also for providing consulting services.

1.4 AUDITORS AND CAATs

CAATs are tools for drawing inferences and gathering relevant and reliable evidence as per requirements of the assignment. CAATs provide direct access to electronic information and empower auditors not only to perform their existing audits more efficiently and effectively but also facilitate them in knowing how to create and execute new type of IT related audit assignments. CAATs provide a mechanism to gain access and to analyze data as per audit objective and report the audit findings with greater emphasis on the reliability of electronic information maintained in the computer system. There is higher reliability on the audit process as the source of the information used provides and greater assurance on audit findings and opinion. CAATs are available in specific general audit software designed for this purpose but the techniques of CAATs can be applied even by using commonly used software such as MS Excel and by using query/reporting features of commonly used application software. CAATs can be used to perform routine functions or activities which can be done using computers, allowing the auditors to spend more time on analysis and reporting. A good understanding of CAATs and know where and when to apply them is the key to success. ICAI has published a guidance note on CAAT and publication titled: “Data Analysis for Auditors” which may be referred for more details.

1.5 NEED FOR CAATs

In a diverse digital world of clients’ enterprises, the greatest challenges for an Auditor is to use technology to access, analyze and audit this maze of electronic data. CAATs enable auditors to move from the era of ticks of using pencil or pen to the era of clicks by using a mouse. CAATs will help
auditors to change focus from time-consuming manual audit procedures to intelligent analysis of data so as to provide better assurance to clients and also manage audit risks. Some of the key reasons for using CAATs are:

1. Absence of input documents or lack of a visible paper trail may require the use of CAATs in the application of compliance and substantive procedures.
2. Need for obtaining sufficient, relevant and useful evidence from the IT applications or database as per audit objectives.
3. Ensuring audit findings and conclusions are supported by appropriate analysis and interpretation of the evidence.
4. Need to access information from systems having different hardware and software environments, different data structure, record formats, processing functions in a commonly usable format.
5. Need to increased audit quality and comply with auditing standards.
6. Need to identify materiality, risk and significance in an IT environment.
7. Improving the efficiency and effectiveness of the audit process.
8. Ensuring better audit planning and management of audit resources.

1.6 OBTAINING AUDIT DATA

In most cases where CAATs are used, it becomes necessary to obtain copy of data in their original format for independent analysis. The data has to be obtained in commonly accepted format. It is important to understand the format in which the data is stored in the application which is being audited. If the data is a native format which is not readable by audit software, then it is necessary to use the reporting feature of application software and export this data to commonly recognizable format of audit software. For example, auditor may not be aware of the data structure/tables of a software developed through a vendor by the client. In such case, auditor may have to study the reporting features and use the export feature to get the data in the required format. It is very important to educate the client about the need to obtain copy of the data as required for audit. Based on the audit scope and relevant audit environment, auditor may have to finalize the required approach for getting the data for audit. This may include installing audit software on client system or using the application software for audit as feasible.

1.7 KEY STEPS FOR OBTAINING DATA

1. Discuss with client about the requirement of raw data for audit and issue a request letter for getting the requested data in specified form as per the audit objectives.
2. Discuss with the IT personnel responsible for maintain data/application software and obtain copies of record layout and definitions of all fields and ensure that you have an overall understanding of the data. The record layout should describe each field and provide information about the starting and ending positions and the data type (numeric, alphanumeric, character, etc.).
3. Print sample list of the first 100 records in the data file and compare this to a printout of the obtained data to confirm they are correct.
4. Verify data for completeness and accuracy by checking the field types and formats, such as identifying all records with an invalid date in a date field.

5. Obtain control totals of all the key data and compare with totals from the raw data to ensure all records have been properly obtained. This can be performed by importing the data in audit software and reviewing the statistics of all the key fields.

1.8 KEY CAPABILITIES OF CAATs

CAATs refer to using computer for auditing data as per audit objectives. This requires understanding of the IT environment and most critically the core applications and the relevant database and database structure. CAATs could be used by using the relevant functionalities available in general audit software, spreadsheet software or the business application software. However, broadly the key capabilities of CAATs could be categorized as follows:

1. **File access:** This refers to the capability of reading of different record formats and file structures. These include common formats of data such as database, text formats, excel files. This is generally done using the import/ODBC function.

2. **File reorganization:** This refers to the features of indexing, sorting, merging, linking with other identified files. These functions provide auditor with an instant view of the data from different perspectives.

3. **Data selection:** This involves using of global filter conditions to select required data based on specified criteria.

4. **Statistical functions:** This refers to the features of sampling, stratification and frequency analysis. These functions enable intelligent analysis of data.

5. **Arithmetical functions:** This refers to the functions involving use of arithmetic operators. These functions enable performing re-computations and re-performance of results.

**Precautions in using CAATs**

CAATs have distinct advantages for Auditors and enable them to perform various types of tests. However, it is important to ensure that adequate precautions are in taken in using them. Some of the important precautions to be taken by Auditors are:

1. Identify correctly data to be audited
2. Collect the relevant and correct data files
3. Identify all the important fields that need to be accessed from the system
4. State in advance the format the data can be downloaded and define the fields correctly
5. Ensure the data represent the audit universe correctly and completely.
6. Ensure the data analysis is relevant and complete.
7. Perform substantive testing as required.
8. Information provided by CAATs could be only indicators of problems as relevant and perform detailed testing as required.
1.9 STEP BY STEP METHODOLOGY FOR USING CAATs

CAATs are very critical tools for Auditors. Hence, it is important to formulate appropriate strategies to ensure their effective use. Some of the key strategies for using CAATs are:

1. Identify the scope and objectives of the audit. Based on this, auditor can decided about the need and the extent to which CAAT could be used.
2. Identify the critical data which is being audited as per audit scope and objectives.
3. Identify the sources of data from the enterprise information system/application software. These could be relating to general ledger, inventory, payroll, sundry debtors, sundry creditors.
4. Identify the relevant personnel responsible for the data and information system. These personnel could be from the IT department, vendors, managers, etc.
5. Obtain and review documents relating to data/information systems. This should provide information about data types/data structures and data flow of the system.
6. Understand the software by having a walk-through right from user creation, grant of user access, configuration settings, data entry, query and reporting features.
7. Decide what techniques of CAATs could be used as relevant to the environment by using relevant CAAT software as required.
8. Prepare a detailed plan for analyzing the data. This includes all the above steps.
9. Perform relevant tests on audit data as required and prepare audit findings which will be used for forming audit report/opinion as required.

1.10 EXAMPLES OF TESTS PERFORMED USING CAATs

CAATs can be used for compliance or substantive tests. As per the audit plan, compliance tests are performed first as per risk assessment and based on the results of the compliance tests; detailed compliance tests could be performed. Some examples of tests which can be performed using CAATs are given below:

1. **Identify exceptions:** Identify exceptional transactions based on set criteria. For example, cash transactions above Rs. 20,000.
2. **Analysis of Controls:** Identify whether controls as set have been working as prescribed. For example, transactions are entered as per authorised limits for specified users.
3. **Identify errors:** Identify data, which is inconsistent or erroneous. For e.g.: account number which is not numeric.
4. **Statistical sampling:** Perform various types of statistical analysis to identify samples as required.
5. **Detect frauds:** Identify potential areas of fraud. For example, transactions entered on weekdays or purchases from vendors who are not approved.
6. **Verify calculations:** Re-perform various computations in audit software to confirm the results from application software confirm with the audit software. For e.g.: TDS rate applied as per criteria.
7. **Existence of records:** Identify fields, which have null values. For example: invoices which do not have vendor name.
8. **Data completeness:** Identify whether all fields have valid data. For example: null values in any key field such as date, invoice number or value or name.

9. **Data consistency:** Identify data, which are not consistent with the regular format. For example: invoices which are not in the required sequence.

10. **Duplicate payments:** Establish relationship between two or more tables as required. For example duplicate payment for same invoice.

11. **Inventory obsolescence:** Sort inventory based on data of purchase or categories as per specified aging criteria or period and identify inventory which has become obsolete.

12. **Accounts exceeding authorized limit:** Identify data beyond specified limit. For example, transactions entered by user beyond their authorized limit or payment to vendor beyond amount due or overdraft allowed beyond limit.

### 1.11 Analytical Review Procedures

The various standards on auditing highlight need for acquiring the required skill-sets to audit in an IT environment and using relevant techniques. Many of the requirements of the auditing standards can be complied by adapting them for use in an IT environment as required. For example: Standard on Auditing (SA) 520 Analytical Procedures states:

A1. Analytical procedures include the consideration of comparisons of the entity’s financial information with, for example:

- Comparable information for prior periods.
- Anticipated results of the entity, such as budgets or forecasts, or expectations of the auditor, such as an estimation of depreciation.
- Similar industry information, such as a comparison of the entity’s ratio of sales to accounts receivable with industry averages or with other entities of comparable size in the same industry.

A2. Analytical procedures also include consideration of relationships, for example: Among elements of financial information that would be expected to conform to a predictable pattern based on the entity’s experience, such as gross margin percentages.

Most of the analytical procedures can be performed in an IT environment using CAATs which makes the audit process much more effective and efficient.

### Summary

CAATs enable auditors to use computers as a tool to audit electronic data. CAATs provide auditors access to data in the medium in which it is stored, eliminating the boundaries of how the data can be audited. As auditors start using CAATs, they will be in a better position to have a considerable impact on their audit and auditee as more time is spent on analysis and less time on routine verification. It is important to understand the client IT environment and chart out which techniques of CAAT could be used. Initially, time needs to be invested in this Endeavour but once the audit plan is prepared based on the IT environment as per audit scope, re-use becomes easier. However, the audit plan and tests need to be updated based on changes in the IT environment as relevant. Using CAATs provides greater assurance of audit process to the auditor and also to the auditee. The key to using CAAT is recognizing the need, learning how to use CAATs and using them in practical situations.
ICAI has issued guidelines on CAAT and the various assurance standards highlight the importance of using CAATs as relevant for audit. ISACA has also issued standards and guidelines on auditing and CAATs. Some of the key aspects from these standards and guidelines are given below.

2.1 NEED FOR USING CAATs

As entities increase the use of information systems to record, transact and process data, the need for the auditors to utilize tools to adequately assess risk becomes an integral part of audit coverage. The use of computer-assisted audit techniques (CAATs) serves as an important tool for the auditor to evaluate the control environment in an efficient and effective manner. The use of CAATs can lead to increased audit coverage, more thorough and consistent analysis of data, and reduction in risk.

CAATs include many types of tools and techniques, such as generalized audit software, customized queries or scripts, utility software, software tracing and mapping, and audit expert systems. CAATs may be used in performing various audit procedures including:

- Tests of details of transactions and balances
- Analytical review procedures
- Compliance tests of general controls
- Compliance tests of application controls

CAATs may produce a large proportion of the audit evidence developed on audits and, as a result, the auditor should carefully plan for and exhibit due professional care in the use of CAATs.

2.2 KEY FACTORS TO BE CONSIDERED IN USING CAATs

When planning the audit, the IS auditor should consider an appropriate combination of manual techniques and CAATs. In determining whether to use CAATs, the factors to be considered include:

- Computer knowledge, expertise, and experience of the IS auditor
- Availability of suitable CAATs and IS facilities
- Efficiency and effectiveness of using CAATs over manual techniques
- Time constraints
- Integrity of the information system and IT environment
- Level of audit risk
2.3 CAATs PLANNING STEPS

The major steps to be undertaken by the auditor in preparing for the application of the selected CAATs include the following:

- Set the audit objectives of the CAATs, which may be included in the terms of reference for the exercise.
- Determine the accessibility and availability of the organization’s IS facilities, programs/systems and data.
- Clearly understand composition of data to be processed including quantity, type, format and layout.
- Define the procedures to be undertaken (e.g., statistical sampling, recalculation, confirmation).
- Define output requirements.
- Determine resource requirements, i.e., personnel, CAATs, processing environment (the organization’s IS facilities or audit IS facilities).
- Obtain access to the organization’s IS facilities, programs/systems and data, including file definitions.
- Document CAATs to be used, including objectives, high-level flowcharts and run instructions.

2.4 AUDIT EVIDENCE AND CAATs

Audit is primarily said to be the process of collecting and evaluating audit evidence as per audit objectives. Based on the scope and objectives of audit, auditor can obtain the audit evidence by:

- Inspection
- Observation
- Inquiry and confirmation
- Re-performance
- Recalculation
- Computation
- Analytical procedures
- Other generally accepted methods

2.5 CAATs DOCUMENTATION

Work papers

The step-by-step CAATs process should be sufficiently documented to provide adequate audit evidence. Specifically, the audit work papers should contain sufficient documentation to describe the CAATs application, including the details set out in the following sections.

Planning

Documentation should include:

- CAATs objectives
Computer Assisted Audit Techniques

- CAATs to be used
- Controls to be exercised
- Staffing and timing

**Execution**

- Documentation should include:
  - CAATs preparation and testing procedures and controls
  - Details of the tests performed by the CAATs
  - Details of inputs (e.g., data used, file layouts), testing periods, processing (e.g., CAATs high-level flowcharts, logic) and outputs (e.g., log files, reports)
  - Listing of relevant parameters or source code

**Audit Evidence**

Documentation should include:

- Output produced
- Description of the audit analysis work performed on the output
- Audit findings
- Audit conclusions
- Audit recommendations

In audits where CAAT is used, it is advisable that the audit report includes a clear description of the CAATs used in the objectives, scope and methodology section. The description of CAATs used should also be included in the body of the report, where the specific finding relating to the use of CAATs is discussed. This description should not be overly detailed, but it should provide a good overview for the reader.

**2.6 AUDIT TEST USING CAATs**

If the data to be audited is available in electronic form, then CAATs could be used for:

- Inquiry and confirmation – identifying accounts for which external confirmation is to be obtained. Request letters for confirmation of balances can be printed using CAAT software.

- Re-performance: The processing of transactions done by the application software can be re-performed and the resultant data can be compared to verify correctness and completeness. For example: Postings of transactions to personal ledger can be re-performed using the original transaction data base and compared with classified transactions as per ledgers.

- Re-calculation: All the computations which were done electronically by the application software used in the enterprise can be independently validated by re-performing the computations. For example, Tax deducted at source or VAT charged on sales, interest computation, etc. can be re-computed in CAAT software and validated with the computed totals from the original application software to confirm correctness of processing of transactions.
Computation: using CAAT software, it is possible to compute totals to confirm correctness. For example, the VAT payments made for the year can be total in CAAT software to compare with the total payments as per VAT returns. The interest debited can be computed and compared with actual debit to interest a/c for the year.

Analytical procedures: Based on the data available in electronic format, various analytical procedures can be performed by comparing and relating various aspects of financial and non-financial information.

2.7 AUDIT SAMPLING

Auditor has to design and select an audit sample and evaluate sample results. Appropriate sampling and evaluation will meet the requirements of ‘sufficient, reliable, relevant and useful evidence’ and ‘supported by appropriate analysis. Auditor should consider selection techniques that result in a statistically based representative sample for performing compliance or substantive testing.

When using either statistical or non-statistical sampling methods, auditor should design and select an audit sample, perform audit procedures, and evaluate sample results to obtain sufficient, reliable, relevant and useful audit evidence. Audit sampling is defined as the application of audit procedures to less than 100 percent of the population to enable the IS auditor to evaluate audit evidence about some characteristic of the items selected to form or assist in forming a conclusion concerning the population.

Statistical sampling involves the use of techniques from which mathematically constructed conclusions regarding the population can be drawn. Non-statistical sampling is not statistically based, and results should not be extrapolated over the population as the sample is unlikely to be representative of the population.

Design of the Sample

When designing the size and structure of an audit sample, IS auditors should consider the specific audit objectives, the nature of the population, and the sampling and selection methods.

Auditor should consider the need to involve appropriate specialists in the design and analysis of samples.

Selection of the Sample

There are four commonly used sampling methods. Statistical sampling methods are:

- **Random sampling** — Ensures that all combinations of sampling units in the population have an equal chance of selection

- **Systematic sampling** — Involves selecting sampling units using a fixed interval between selections, the first interval having a random start. Examples include Monetary Unit Sampling or Value Weighted selection where each individual monetary value (e.g., Rs. 1) in the population is given an equal chance of selection. As the individual monetary unit cannot ordinarily be examined separately, the item which includes that monetary unit is selected for examination. This method systematically weights the selection in favour of the larger amounts but still gives every monetary value an equal opportunity for selection. Another example includes selecting every ‘nth sampling unit
Non-statistical sampling methods are:

- **Haphazard sampling** — The IS auditor selects the sample without following a structured technique, while avoiding any conscious bias or predictability. However, analysis of a haphazard sample should not be relied upon to form a conclusion on the population.

- **Judgmental sampling** — The IS auditor places a bias on the sample (e.g., all sampling units over a certain value, all for a specific type of exception, all negatives, all new users). It should be noted that a judgmental sample is not statistically based and results should not be extrapolated over the population as the sample is unlikely to be representative of the population.

Auditor should select sample items in such a way that the sample is expected to be representative of the population regarding the characteristics being tested, i.e., using statistical sampling methods. To maintain audit independence, the IS auditor should ensure that the population is complete and control the selection of the sample. For a sample to be representative of the population, all sampling units in the population should have an equal or known probability of being selected, i.e., statistical sampling methods.
LEARNING OBJECTIVES

- To gain understanding on Importing different file formats into IDEA.
- To understand on how to generate field statistics for the database.
- To understand formatting Data.

There are many audit software available in the market. However, for the purpose of learning CAATs, we will be using IDEA software in this chapter and also for performing exercises in the lab. Students may refer to the ICAI publication titled: “Practical application of CAAT–case studies” for more examples and details of CAATs using IDEA Software.

3.1 IMPORTING DATA

Function Description

Important Assistant brings the selected file or files into IDEA database management system. User-friendly Import assistant guides user through a series of steps and instructions for importing the file into the Software. All the functionalities of IDEA can be performed only when the file is available within IDEA. Hence, the first step in data analysis is ensuring that the files to be audited are in selected format acceptable in IDEA and are imported into IDEA.

Let us assume you have an Excel or Access file and you want to perform certain idea functionalities on it, then it is important to import these files into IDEA. IDEA facilitates user to import external files in different formats like Access, Excel, dbase or other ODBC/DSN formats into IDEA database.

We are explaining below step by step how to import a file into IDEA. If you have access to IDEA Server, you can also import files to IDEA Server. For now, you will have to import a file to your Working Folder. A Working Folder is simply a folder that contains the IDEA databases that you wish to analyze. It is recommended that all the files related to each audit or investigation be stored in a separate folder or directory to simplify file management and housekeeping.

Note: The sample files which are used for explaining how to perform various functionalities of idea are available in the C:\Program Files\IDEA\UserFiles\Tutorial\ when IDEA is installed with default settings. You can perform exercises using the relevant sample data files. Please check with your lab instructor about the location of these files. We will be using tutorial/sample files for explaining different functionalities of IDEA in this booklet.
You can import files of different format into IDEA. However, for this exercise, we are giving below an example for importing an Access file into IDEA.

**Step by Step Procedure for Importing Data into IDEA**

**Location**
- File>Import Assistant >Import to IDEA
- Alternatively, on the Operations toolbar, click the Import to IDEA button. Import Assistant dialog box appears as shown in Fig 3.1.1.

![Fig. 3.1.1 Import Assistant Location](image)

**Step 1: Select the Format**

![Fig. 3.1.2 Import Assistant Step 2](image)
(a) In the Import Assistant Dialog Box, select Microsoft Access from the list.

(b) Click the **Browse** button next to the File name box to select the Microsoft Access database you want to Import.

(c) Navigate to and select `C:\Program Files\IDEA\UserFiles\Tutorial\Customer.MDB`.

(d) Click **Open**.

(e) The **Select File** dialog box closes and the selected file name and path appear in the File namebox in the **Import Assistant** dialog box.

(f) Click **Next**.

The **Microsoft Access** dialog box appears.

**Step 2: Select Tables as shown in Fig 3.1.3**

![Fig. 3.1.3 Import Assistant Step 3](image)

(g) In the **Select tables** box, select **Database1**.

**Note:** If you import a Microsoft Access file that contains more than one table, you may simultaneously import multiple tables by selecting the associated check boxes. However, any options you select will apply to all imported tables.

(h) All Character fields will be imported with a length of 255 characters unless changed. This is not likely to be the underlying Character field length. Therefore, leave the **Scan records for field length** check box selected. Also, accept the default value in the **Scan only** box in order to scan 10,000 records to determine the maximum field length.

(g) Accept the default output file name (**Customer**), and then click **OK**. When the file is imported, the database name becomes filename-tablename. In this case, the file you imported becomes an IDEA database called **Customer-Database**.
Step 3: Result - Customer-Database

On clicking Ok, the Customer Database is imported into IDEA. The imported database is opened in the Database Window. In the File Explorer Window the imported database is highlighted as shown in Fig 3.1.4.

![Customer Database](image)

**Fig. 3.1.4. Customer Database is imported into IDEA**

**Exercises:**

1. Please perform the following exercises in your lab.
2. Please import excel file following the above steps.
3. Please import text delimiter file and notice the different steps.
4. Please import excel file using ODBC option.

### 3.2 IMPORTING DATA

Click on field statistics after importing the files and understand the nature of data which is imported.

**Function Description**

You can use the Export Database task to create an external file from an IDEA database so that you can use the data in other applications, such as a spreadsheet package. IDEA exports data in a number of text, database, spreadsheet, and mail merge formats.
You can use the Export Database task to create an external file from an IDEA database so that you can use the data in other applications, such as a spreadsheet package. You can also use Copy and Paste to incorporate portions of text or sections of database into other Windows applications. IDEA also supports drag and drop into any other OLE2 container application, such as Microsoft Excel. IDEA exports data in a number of text, database, spreadsheet, and mail merge formats.

**Step by step process for exporting files from IDEA.**

**Location**

- File>Export Database

![Fig. 3.2.1 Export Database Location](image)

**Database Used**

- Customer-Database1
Export Dialog Box

(a) In Records to Select, on selecting All will select the entire records. On selecting Range allows the user to select the Starting and Ending record number.

(b) By default the path is set to C:\Users\Saranya\Documents\IDEA\Samples\Customer-Database1.

(c) In Export Type, user can select the format in which the current file has to be exported and also name the resultant exported file.

(d) In Filename allows the user to select the path to which the file has to be exported.

(e) On clicking Fields, we can select or unselect the fields that have to be exported.

(f) During Exporting process, condition or criteria can be applied using the criteria button. On clicking this button, Equation Editor Dialog box is opened which facilitates the user to write in query or condition.

(g) Clicking Ok, exports the active file into given desired format.

Exercise:
1. Export actual idea file to different formats such as: HTML, Excel, text delimited file and check the results.

3.3 SUMMARIZATION

Learning Objectives:
To total the sales transaction by INVOICENO to produce a list of outstanding Sales as well as to identify the number of INVOICENO and the Sales per INVOICENO.

Function Description
Summarization accumulates the values of Numeric fields for each unique key. For example,
summarizing an Accounts Payable database by account number (the key) and totalling invoice amounts produces a database or Results output of outstanding liabilities by supplier. The Summarization task provides:

- A list of unique items (keys) in the database
- The number of records for each key
- Totals of one or more numeric fields for each key

You may select up to eight fields to summarize by or if using one field you can use the Quick Summarization option. If using multiple fields, the Summarization task works faster and is more efficient on a database that has been sorted.

You may select any number of Numeric fields to total. The resultant database provides a list of unique key items and the number of records (NO_OF_RECS) for each key from where you can drill down to display the individual records for the key. You can select additional field data to be selected from fields from the first occurrence or the last occurrence.

The summarization result grid is limited to 4,000 rows and does not display results beyond 4,000 rows. If you expect your result will have more than 4,000 rows, you must choose to create a database.

**Step by Step Procedure for summarization**

Ensure that Sales Transactions is the active database and the Data property is selected in the Properties window.

**Location**

- Analysis > Summarization

(a) In the Summarization Dialog Box, select the following:
- Fields to summarize: INVOICENO
Numeric fields to total: SALES

(b) Click **Fields**. The **Fields** dialog box appears. Note that no fields are selected. This stops unnecessary information from being included in the summarized database.

(c) Click **OK** to return to the **Summarization** dialog box.

Select the **Use Quick Summarization** check box. The Use Quick Summarization check box may be selected as a faster means to summarize your database. However, Quick Summarization may only be used if the database has no more than 4,000 unique keys. In addition, Quick Summarization allows you to select only one field to summarize.

There are two types of output from the **Summarization** task:
- Summarization database
- Summarization result

Note that as with most tasks in IDEA, you may apply a criterion to the task, for example; only summarize transactions for a specified period. As with all other tasks where you can apply a criterion, if you apply the criterion to the database using the **Criteria** link in the Properties window, the criterion equation appears in the **Criteria** text box on the task dialog box. However, you may enter a new criterion or modify an existing one using the Equation Editor as shown in Fig 3.3.2.

(d) In the **Statistics to include** area, accept the default selections of **Sum**.

(e) Click **OK**.

View the resultant database and note the following fields:
- **INVOICENO** - List of unique INVOICENO
- **NO_OF_RECS** - Number of records for each INVOICENO
- **Sales_SUM** - Total Sales of the transactions for each INVOICENO
Also note the number of records (989) on the Status bar displayed in the below figure. Note that IDEA automatically creates an Action Field link to the parent database (Sales Transactions). It allows you to display the records from the Sales Transactions database by clicking on a value (in blue) in the NO_OF_RECS field as shown in Fig 3.3.3.

![Fig. 3.3.3 Summarization Result](image)

Exercises:

1. Use the sample employees file in the sample database of idea and perform summarization on branch and total on salary. Click on the hyperlink of the number of records and check the result.
2. Use the sample employees file in sample database of idea and perform summarization on country and total on salary and check the result.

### 3.4 STATISTICS

#### Learning Objectives

Understanding to view the field statistics for the Numeric fields in the active database.

The statistics is used for reconciling totals, obtaining a general understanding of the ranges of values in the database, and highlighting potential errors and the area of weakness to focus subsequent tasks.

#### Function Description

The Field Statistics property provides statistical information about all Numeric, Date, and Time fields within the active database. The field statistics are available and displayed for all records in the database, with any applied criteria ignored.

By default, the Field Statistics window displays the statistics for Numeric fields. Ensure that Customer-Database is the active database and the Data property is selected in the Properties window.

#### Step by Step Procedure for Statistics

**Location**

- In the Properties window, click Field Statistics.
On clicking Field Statistics in property window, IDEA displays below message box. Click **Yes** to generate field statistics for all fields.

By Default By default, the **Field Statistics** window displays the statistics for Numeric fields. In this case, field statistics appear for all the available numeric fields. To view field statistics for the Date fields and Time fields in a database, click **Date** and **Time** in the **Field Type** area of the **Field Statistics** window. In the current database, there are no Date or Time fields.

If the database contained more than one Date, Numeric, or Time field, multiple date, numeric, or Time field statistics would appear together on the same screen for easy comparison of values.
Statistics for Numeric Fields

As shown in the above image, active database contains six numeric fields and the statistics for all six numeric fields are displayed in Field Statistics window which allows easy comparison between them.

Study the field statistics for the all Numeric fields. Note in particular:

<table>
<thead>
<tr>
<th>Field Type</th>
<th>COUNTRYNO</th>
<th>UNIT_PRICE</th>
<th>QTY</th>
<th>PROD_CODE</th>
<th>AMOUNT</th>
<th>PAYMENT_AMT</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Records</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
</tr>
<tr>
<td># of Zero Items</td>
<td>0</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Average Value</td>
<td>8.08</td>
<td>8.08</td>
<td>90.07</td>
<td>9627.42</td>
<td>710.36</td>
<td>710.36</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>3</td>
<td>3.84</td>
<td>0</td>
<td>9598</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>10</td>
<td>9.99</td>
<td>3496</td>
<td>9951</td>
<td>27793.20</td>
<td>27793.20</td>
</tr>
</tbody>
</table>

Below two Images displays statistics for Date and Time fields for current active database.

Statistics for Date Fields
Statistics for Time Fields

Fig. 3.4.5 Time Statistics

Exercise:
- Click on field statistics for sample employees and understand profile of data.
- Click on field statistics for sample-inventory and understand profile of data.

3.5 SAMPLING

Learning Objectives
- To draw a number of records with fixed interval for testing.
- To select a random sample of records for testing.
- To extracts a random sample with a specified number of records from each of a series of bands.

Function Description
Sampling in IDEA is broadly statistical and probability-based.

The probability-based sampling techniques are: Systematic, Random and Stratified Random.

The statistical sampling techniques are: Attribute, Classical Variable and Monetary Unit sampling.

We will be covering examples of systematic and random sampling only in this training. Students may try other forms of sampling in the lab exercise.

Systematic Record Sampling
Systematic Record Samplings a method to extract a number of records from a database at equal intervals to a separate database. It is often referred to as interval sampling.
There are two methods of determining the sample:

- Entering the number of records, in which case IDEA computes the interval size.
- Entering the selection interval, in which case IDEA computes the number of records.

IDEA calculates the above parameters on the number of records in the database and defaults to the first to last records. However, we can extract the sample from a range of records, if required.

**Step by Step Procedure for performing Systematic Sampling**

Ensure that **Sales Transaction-Sales Trans** is the active database and the **Data** property is selected in the **Properties** window.

**Location**

- **Sampling >Systematic.**
- Alternatively, click the **Random Record Sampling** button on the **Operations** toolbar. The **Random Record Sampling** dialog box appears as shown in Fig 3.5.1

![Fig. 3.5.1 Systematic Sampling](image-url)
Step 1: Number of Records

(a) In the **Number of records to select** box, enter 100.
(b) Accept the random number seed provided by IDEA. IDEA uses the random number seed to start the algorithm for calculating the random numbers. If a sample needs to be extended, then entering the same seed but with a larger sample size produces the same original selection plus the required additional records.
(c) Accept the defaults in the **Starting record number to select** and the **Ending record number** to
   (a) **Select** boxes.
   (b) IDEA sets the defaults as the first and last records; in this case 1 and 1000.
(d) In the **File name** box, enter **Systematic Sales**.

Step 2: Selection Interval

(a) In the **Number of records to select** set to 100.
(b) Enter **10** for the **Selection interval**.
(c) Enter **1** for the **Starting record number to select**.
(d) Enter **1000** for the **Ending record number to select**.
(e) Enter **Systematic Sales** in the **File name**.
(a) In Selection Interval tab page, enter 10 for selection interval. This means it picks every 10th records from 1 to 1000 records. Total records picked are 100.

(b) Click Ok.

Step 3: Result of Systematic Sampling

![Systematic Result](image)

**Fig. 3.5.4 Systematic Result**

(a) As displayed in the result, every 10th record is picked from the active database and total numbers of records are 100.

**Random Record Sampling**

Random Record Sampling is a commonly used method of sampling. With it we enter the sample size as well as the range of records from which the sample is to be extracted to a separate database. Then, using a random number seed, IDEA generates a list of random numbers and selects the appropriate records associated with these numbers.

**Step by Step Procedure for performing Random Sampling**

Ensure that Sales Transaction-Sales Trans is the active database and the Data property is selected in the Properties window.
Location

- **Sampling > Random.**
- Alternatively, click the Random Record Sampling button on the Operations toolbar. The Random Record Sampling dialog box appears as shown in Fig. 3.5.5

![Fig. 3.5.5 Random Sampling Location](image)

![Fig. 3.5.6 Random Sampling](image)

(a) In the Number of records to select box, enter 10.
(b) Accept the random number seed provided by IDEA. IDEA uses the random number seed to start the algorithm for calculating the random numbers. If a sample needs to be extended, then entering the same seed but with a larger sample size produces the same original selection plus the required additional records.
(c) Accept the defaults in the Starting record number to select and the Ending record number to Select boxes.
IDEA sets the defaults as the first and last records; in this case 1 and 999.
(d) Leave the Allow duplicate records check box unselected.
(e) In the File name box, enter Random Sales.
(f) On clicking Ok total 10 records are extracted from active database.
Random Sampling Result

Exercises:
1. Perform sampling on different sample files available in sample database.
2. Perform stratified sampling using one of the tables in the sample database.

3.6 STRATIFICATION

Learning Objectives

To stratifying the data from the file into bands and gaining the profile of the data. The data can be stratified based using the numeric, date or character field to select a random sample of records for testing.

Function Description

The process of stratification involves creating bands based on ranges of values (normally from the minimum to the maximum values of one or more fields) and accumulating the records from the database into the appropriate bands. By totaling the number of records and value of each band, you can gain a profile of the data in the database. You can then investigate any deviations from expected trends. You may have up to 1,000 stratification bands. The stratification analysis is also useful for determining high and low cut-off values for testing exceptional items.

A numeric stratification analysis can also be created for each unique value or key in a field by selecting that field from the Group by drop down list. For example, you could produce a profile of sales for each salesperson. This can potentially create an extremely large volume of output however the maximum number of groups that will be displayed in the result is 80. If there are more than 80 groups, only the first 80 are displayed. Therefore, there is an option to specify low and high cut-off values to restrict output. Only groups whose total value of transactions is between the specified range are output.

To include all items in the stratification analysis, the bands should start less than the minimum value and the upper band greater than the maximum value of all fields.

Date and Character stratification are different than Numeric Stratification in the sense that different fields are totaled to the one used for banding.
Step by Step Procedure for performing Numeric Stratification

Ensure that Sales Transaction-Sales Trans is the active database.

Location
- Analysis > Stratification.

(a) In the Field to stratify box, select SALES.
(b) In the Fields to total on box, select SALES.
(c) Group By USERID.
(d) Specify the stratification bands:
  - Change the increment to 10,000.
  - Click in the < Upper Limit text box of the first row. Note the text box is filled with the value 10,500.
  - Click the second row of the spreadsheet area. This automatically fills with 10,500 to 20,500
  - Highlight the next three rows of the spreadsheet area to take the range to 50,500.
Ensure the **Create result** check box is selected. In the **Result name** box, enter **Stratification Sales**.

(e) Click **OK**.

(f) The **Numeric Stratification** result output for the **Sales Transactions** database becomes active and appears as a link in the **Results** area of the **Properties** window. Note that there are 6 accounts in the first band ( \( \geq 500 \) and \( < 10,500 \)).

![Fig. 3.6.3 Stratification Result](image)

**Exercises**

- Use sample-employees file and perform numeric stratification on salary with various bands.
- Use sample-employees and perform character stratification based on name.

### 3.7 SORTING

**Learning Objectives**

- To create a new database in which its records are physically sorted in a specified order.

Ensure that **Sales Transaction-Sales Trans** is the active database.

**Location**

- Data>Sort.
Function Description

The fields you select to sort the records by are known as keys. A sort order may contain up to eight keys. When creating a sort order, the most significant field is selected first (primary key), followed by the next most significant field, and so on down to the least significant field (secondary keys).

With the Sort task, a new database is created with the records in the sequence of the key. This new database is a child database and appears in the File Explorer below the main database. Once you have sorted a database, IDEA displays the records in the database in the sort order and updates the list in the Indices area of the Properties window.

Step by Step Procedure for performing Sorting

Click on the Field list button (down arrow) on the Field column of the first row of the dialog grid to display a list of all fields in the database. Select the required field.

The Direction column displays Ascending order for each field selected. To change the order for a field, click the Direction column to activate the Direction list button.

To add further fields to the sort order, click the next available row in the Field column and select the field and its direction as above. A maximum of eight fields may be entered. Note: If required, you can delete a key from the sort order by selecting the required row and then clicking Delete Key. Here DATE field is selected in descending direction.
(e) In File name box enter **Sorted database**.
(f) Click **OK**. Date field is sorted in descending direction as shown in the below figure.

**Sorted-Database**

![Sorted database](image-url)
Exercises:
1. Use Sample-Employees and sort on city.
2. Use Sample-Employees and sort on First Name, City and Salary.
3. Duplicate Detection and Gap Detection

### 3.8 DUPLICATE DETECTION

**Learning Objectives**
- To test the validity of invoices.
- To test for duplicate invoice numbers.

**Function Description**
IDEA includes two key functions to identify exceptions, irregularities, anomalies and errors.

These are: Duplicate Detection and Gap Detection.

These functions assist the user to sift through large volumes of data and help pin-point specific duplicate entries or specific missing entries. These also help the user obtain an assurance on all the data reviewed by it. The duplicate or missing items identified can be taken up for testing after running the respective duplicate and gap tests within IDEA.

Duplicate Detection and Gap Detection tests are standard passing tests which are run on every database right at the inception prior to detailed excepting and analytical testing within IDEA. The tests do not require much querying experience and resemble plug-n-play tests. Both tests run largely on formatted sequential data fields like Invoice Number, Purchase Order Number, Cheque Number, etc.

**Step by Step Procedure for Duplicate Detection**

Ensure that **Sales Transaction-Sales Trans** is the active database.

**Location**
- Analysis > Duplicate Key > Detection.

![Fig. 3.8.1 Identify Duplicates Location](image)
(a) The **Duplicate Key Detection** dialog box appears.

![Fig. 3.8.2 Duplicate Key Detection](image)

(b) Leave the **Output Duplicate Records** option selected.

(c) Click Fields and select **INVOICENO**, **INVDATE**, and **USERID**.

![Fig. 3.8.3 Fields](image)

(d) Click **Key**. The **Define Key** dialog box appears.

(e) In the **Field** column, select **INVOICENO** and leave the direction as **Ascending**.

![Fig. 3.8.4 Define Key](image)
(f) Click OK to return to the **Duplicate Key Detection** dialog box.

(g) In the **File name** box, enter **Duplicate**.

(h) Click OK to run the task.

**Duplicate Result**

(a) In the resultant database of 8 transactions with DATE, INVOICENO and USERID duplicate values are investigated.

(b) Duplicate file is opened as current active database.

(c) The given result is sorted based on INVOICENO field in ascending direction.

---

**Fig. 3.8.5 Duplicate Result**

**Exercises:**

1. Use Sample-Employees file and find out duplicates based on name.
2. Use Sample-Employees and identify duplicates based on fields: First Name and Name.
3. Use Sample-Employees file and identify duplicates based on Address.

---

**3.9 GAP DETECTION**

**Learning Objectives**

- To test for completeness.
- To test for gaps in the invoice number sequence.

Ensure that **Sales Transaction** is the active database.

**Location**

- Analysis > Gap Detection.
Step by Step Procedure for Gap Detection

(a) In the Field to use list box, select INVOICENO.
(b) In the Output area, ensure the Create result check box is selected.
(c) In the Result name box, enter Gap Detection Sales.
(d) On clicking Ok, the result database contains the gaps occurring From: INVOICENO, To: INVOICE and Number in the active database.
Exercise:
1. Use Sample-Bank Transactions file and identify gaps based on date.

**3.10 AGING**

Learning Objectives
- To age a selected database from a particular date for up to six specified intervals. These intervals can be days, months, or years.
- To age the outstanding debts at the yearend in order to determine provisions required against bad debts.

Function Description
Aging function presents aged summaries of data. This summary may be based on the current date or a specified cutoff date. Use the Aging task to age a selected database from a particular date for up to six specified intervals. These intervals can be days, months, or years. For example, you can age the outstanding debts at the yearend in order to determine provisions required against bad debts.

The most common use of the Aging task is with Accounts Receivable or Debtor Ledgers. However, also consider using Aging on inventory databases (date of last movement) or on short-term loan databases.

The Aging task optionally produces:
- A detailed aging database
- A key summary database
- A Results output

Ensure that **Sales Transaction-Sales Trans** is the active database.
Location

- Analysis >Aging.

**Fig. 3.10.1 Aging Location**

Step by Step Procedure for performing Aging:

(a) Enter the Aging date to 2012/06/27 i.e yyyy/mm/dd format.

(b) Set the Aging field to use to DATE.

(c) Set the Amount field to total to SALES.

(d) Aging interval in Days. The other options are Months and year.

(e) The Aging task optionally produces:
   - A detailed aging database.
   - A key summary database.
   - A result output.

(f) Select only Create result option.

(g) In the Name box enter Aging Sales.

(h) On clicking Ok, the below output is displayed.
3.11 DATA EXTRACTION

Learning Objectives

- To perform an extraction to identify accounts where the new credit limit has been exceeded.

Function description

Extract selected data from a file for further investigation for creating a new file of logically selected records. For example: you can use Direct Extraction to perform a single extraction on a database, or up to 50 separate extractions with a single pass through the database.
Step by Step Procedure for Data Extraction

Ensure that **Sales Transaction** is the active database.

### Location

- **Data > Extractions > Direct Extraction.**

#### Fig. 3.11.1 Direct Extraction Location

![Direct Extraction Location](image1)

#### Fig. 3.11.2 Direct Extraction

(a) In the **File Name** column, replace the default file name with **Sales Greater than 20,000**.

(b) Click the **Equation Editor** Button, and then enter the equation **SALES >= 20000**.

(c) Click the **Validate and Exit** button to return to the **Extract to File(s)** dialog box.

(d) On clicking Ok creates a new child database with name **Sales Greater than 20,000** which contain records where Sales is greater than 20,000.
Record Extraction Result

Exercises:

- Use Sample-Suppliers file and extract supplier details where supplier no. is from 30000 to 90000.
- Use Sample-Suppliers file and extract supplier details where country is “Chile”.

3.12 BENFORD’S LAW

Learning Objectives

To compare the data with the data pattern predicted by Benford’s Law analysis.

Function Description

Benford’s Law states that digits and digit sequences in a data set follow a predictable pattern. The Benford’s Law task generates a database, and optionally a Results output, that you can analyze to identify possible errors, potential fraud, or other irregularities. If artificial values are present in the selected database, the distribution of the digits may have a different shape, when viewed graphically, than the shape predicted by Benford’s Law.

A Benford’s Law analysis is most effective on data:

- Comprised of similar sized values for similar phenomena.
- Without built-in minimum and maximum values.
- Without assigned numbers, such as bank accounts numbers and zip codes.
- With four or more digits.

Step by Step Procedure for using Benford’s Law function

Ensure that Sales Transaction is the active database.
Location

- Analysis > Benford’s Law.

Fig. 3.12.1 Benford’s Law Location

The analysis counts digit sequences of values in the database and compares the totals to the predicted result according to Benford’s Law. Non-zero digits are counted from left to right and values below 10 are ignored.

Fig. 3.12.2 Benford’s Law

Steps for Benfords Law

(a) Select the field to analyze. In the Field to analyze drop-down list, select the Numeric field for which you want to analyze with Benford’s Law.

(b) Select the number type. In the Include Values area, select the check box for the required number type (Positive or Negative).

(c) Optionally, specify the upper and lower boundaries. The upper and lower boundaries define the acceptable range of values where the actual result can appear. Select the Show boundaries check box to include the boundaries in the Results output and resultant database.

(d) Select the required analysis types. In the Analysis Type area, select the required analysis types, and then accept or change the associated database file names.

(e) Optionally, create a Results output.
(f) On Clicking Ok, we get results for Benford’s First Digit, Second Digit, First Three Digit and First Two Digit.

(g) In the below image, Benford First Digit result is displayed. To view Second, Third and Fourth Digit result, click on the IDEA files explorer.

![Fig. 3.12.3 Benford’s Law First Digit Result]

**Exercise:** Use Sample-Bank Transactions file and perform Benford’s Law function on amount.

### 3.13 CONSOLIDATION OF DATA

**Learning Objectives**

- To summarize data and create a report based on many calculations.
- To define how your data is displayed and organized.

**Function Description**

Consolidation is the process of combining values from several ranges of data. Data can be consolidated by Pivot Table.

**Step by Step Procedure for reporting**

Ensure that **Sales Transaction** is the active database.

**Location**

- Analysis>Pivot Table.

![Fig. 3.13.1 Pivot Table Location]
In the **Result name** box, enter a name for the **Pivot Table Results** output.

![Fig. 3.13.2 Pivot Table](image)

Click **OK**. The **Pivot Table Field List** dialog box appears. It contains the list of available fields in the database, with their field types in adjacent brackets.

![Fig. 3.13.3 Pivot Table Field List](image)

(a) Create the table by dragging fields to summarize from the **Pivot Table Field List** dialog box to the following areas: **Drop Column Fields Here**, **Drop Row Fields Here**, **Drop Data Items Here** and **Drop Page Fields Here**:

(b) You can drag the same field more than once into the data area creating a new statistic each time - sum, amount, and so on.

(c) You can add multiple fields to the Column and/or Row area. When more than one field is added to a column or row area the data in the rightmost header column or lowest header row is grouped by data in the preceding row or column.

(d) You can also select a field from the **Pivot Table Field List** dialog box, select the area of the pivot table to which you want to add the selected field from the drop-down list, and then click **Add To**. Once a field has been added to the table, it is highlighted in the **Pivot Table Field List** dialog box.

(e) For every field added to the pivot table, its sort order is added to the **Indices** area of the **Properties** window. You may delete these indices from the **Indices** area without affecting the pivot table.

(f) From the **Pivot Table Field List** dialog box, click **Close**.
Pivot Table Result

![Pivot Table Result](image)

Exercises:
1. Use Sample-Employees file and perform pivot table function with branch as row and salary as column name.
2. Use Pivot table to find out salary as per country/branch and total of salary.

3.14 EQUATION EDITOR

Learning Objectives

- To create formulae and equations in IDEA.

Functions

Use functions to perform more complex calculations and exception testing. You can use them for date arithmetic, text searches, and some statistical operations. They are very similar in style and operation to functions found in other software packages such as Microsoft Excel, Lotus 1-2-3, and dBASE. Each function calculates a result based upon the parameters passed to the function. Parameters are passed in parentheses.

![Equation Editor](image)
Select the type of function you require from the following category list to view the details for the associated functions:

- All - Complete list of all functions.
- Character - Text manipulation or conversion
- Numeric - Calculations and statistics
- Date and Time - Date and time calculations and conversions
- Matching - Matching of multiple and similar items
- Conditional - IF statements to select different items
- Financial - Financial calculations
- Custom - User-defined functions

**Entering Functions in Equation Editor**

Entering a function involves entering the function name along with any necessary parameters. Functions are entered in the **Equation area** of the **Equation Editor**.

**Entering the Function Name**

To enter the function name in the **Equation area**, expand the appropriate function category and then double-click the required function name. Alternatively, if you know the name of the function, you can type it directly into the **Equation area**. As you type the function name, the **Equation Editor** provides a list of possible function names.

**Entering the Parameters**

Parameters may be complete equations with references to other functions, such as nested functions, or simple constants. Parameters are separated by a list separator. The list separator is defined in your Windows Regional Settings and is usually a comma or a semi-colon. Most functions require a specific number of parameters. A syntax error occurs if you provide the wrong type of parameter to a function.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>EXPLANATIONS</th>
<th>EXAMPLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Represents text or a character expression. A character expression can contain characters or a Character data field.</td>
<td>“Smith” “123456” “Joe” + “Bloggs”</td>
</tr>
<tr>
<td>Number</td>
<td>Represents a numeric expression. A numeric expression can contain numeric constants or Numeric data fields.</td>
<td>123.45 ACCNT BAL QTY*COST</td>
</tr>
<tr>
<td>Time</td>
<td>Represents a time expression. It is displayed in the grid as HH:MM:SS. A time expression can contain a time constant expressed as a string, a numeric constant, a Time field or a Numeric field. Numeric values are interpreted as the number of seconds.</td>
<td>“23:59:59” TRANS_TIME</td>
</tr>
<tr>
<td>Date</td>
<td>Represents a date expression. A date expression is a character string that is eight characters long and contains a valid date in YYYYMMDD format.</td>
<td>“19960131” TRANS_DATE</td>
</tr>
</tbody>
</table>
3.15 REPORTING

Learning Objectives

- To generate a report for the current active database.

Function Description

You can use the reporting feature to generate various types of reports from IDEA.

Step by Step Procedure for reporting


Report Assistant

Accept all defaults and select the Allow headings to span multiple lines check box. Click Next.

![Fig. 3.15.1 Report Assistant](image)

Step 1: Heading Step

Modify the report headings as follows, and then click Next:

- Date – Invoice Date
- INVOICENO – Invoice Number
- NAME – User Name
- SALES – Sales
- USERID – User Id
Step 2: Define Breaks Step

(a) During this step, define if and where control breaks (required for totals in reports) are required.
(b) Sequence the database records in the following order:
   - INVOICENO: Ascending

(c) Click Next.

Note: IDEA displays the records in the report in the order of the index. IDEA displays the index description in the Indices area of the Properties window once you have completed the report.
Step 3: Report Breaks Step
Create a break and total the Sales field for each INVOICENO. Select the options as in the image below, and then click Next.

![Fig. 3.15.4 Report Assistant - Report Breaks](image)

Step 4: Grand Totals Step
Create grand totals for the Sales field only, set the font style to bold, and then click Next.

![Fig. 3.15.5 Report Assistant -Grand Totals](image)

Step 5: Header/Footer Step
Enter/select the following information, and then click Finish.

- **Print cover page**: Select this check box.
- **Title**: Sales Transactions.
- **Comments**: Ordered by User Id and date.
- **Prepared by**: Enter your name or initials.
- **Header**: Enter the name of your organization.
- **Date/Time**: Accept the defaults unless you have particular preferences.

**Note**: The options you have selected effect how the report is printed. The name entered into the Prepared by field appears on reports accessed via the Print Preview of a Results output.
On clicking finish, you can preview the report generated.

### 3.16 FIELD MANIPULATION

**Learning Objectives**

To view field definitions, add or delete fields, change field properties such as field name and type.

**Function Description**

You can use this to modify the fields as required. Please note that making changes to a field through **Field Manipulation** may cause any output based on that field (results, drill-downs, indices, views, etc) to appear incorrect or become invalid. Results may be made valid again by returning the settings to what they were when the result was created. To avoid this, instead of changing a field definition, append a new field to the database with the required definition. For example, instead of changing the type of a field from Character to Numeric, create a new Virtual Numeric field using @Val.

Select **Data>Field Manipulation**. Field Manipulation Dialog Box is opened.

- Add a field
- Click **Append**.
- Enter the field definition:
• **Field Name**: A unique field name up to 40 characters in length that does not contain spaces or special characters.

• **Type**: Click the **Type** text box to display the type options, and then select the required type.

• **Length**: The total length of the field in characters. IDEA automatically determines the length for Editable Numeric fields.

• **Decimal**: If applicable, enter the number of decimal places. If the decimal is implied, enter 0. The maximum number of decimals places allowed is six.

• **Parameter**: Click the **Parameter** field to invoke the **Equation Editor**.

• **Tag Name**: If you have **Smart Analyzer** installed, click the **<No tag>** link to add a tag.

• **Description** (optional): A brief description of the field. The maximum length of the description is 256 characters.

• **Click Ok**

• **Delete a field**

• **Click in the row corresponding to the field to be deleted.**

• **Click Delete.**

• **Click Yes** to confirm the deletion.

**Note**: As a security feature, by default, native fields cannot be deleted

**Change a field type**

Click the **Type** text box to display the type options, and then select the required type.

**Notes:**

- Virtual field types can only be changed to other Virtual field types. For example, a Virtual Character field could be changed to a Virtual Date field.

- Field types cannot be changed for Editable, Boolean or Multistate fields; however the data within this field in the database can be changed.

**Change a field name**

Click in the **Field Name** cell of the field name to be changed and enter the new name.

**Exercise:**

Use Sample-employees file and modify different field and understand the impact.
### Annexure - Using Audit software: Some Practical Examples

These are samples of analysis which can be performed using audit software for various types of audits/reviews.

<table>
<thead>
<tr>
<th>Category</th>
<th>Type of Analysis</th>
<th>CAAT Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creditors</td>
<td>• Payment terms</td>
<td>• Extraction, Date difference</td>
</tr>
<tr>
<td></td>
<td>• Credit Control Policy</td>
<td>• Extraction, Date difference,</td>
</tr>
<tr>
<td></td>
<td>• Aging of creditors</td>
<td>• Aging, date difference</td>
</tr>
<tr>
<td></td>
<td>• Discount policy</td>
<td>• Stratification, Summarization,</td>
</tr>
<tr>
<td></td>
<td>• Authorised suppliers</td>
<td>• Join, Relation, Sort,</td>
</tr>
<tr>
<td></td>
<td>• Authorization</td>
<td>• Filter, Extract, Summarization,</td>
</tr>
<tr>
<td></td>
<td>• Accounting</td>
<td>• Sort, Total, Count, Summarization,</td>
</tr>
<tr>
<td></td>
<td>• Supplier classification on purchases</td>
<td>• Sort, Relation, Join, Summarization, Sort</td>
</tr>
<tr>
<td></td>
<td>• Analysis of purchases by period</td>
<td>• Aging, MIS, Summarization, Count, Total</td>
</tr>
<tr>
<td>Tax Audit</td>
<td>• Identify cash loan\deposits&gt;20000</td>
<td>• Sort, Filter, Extract, Export, Index, analytical tool Sort, Filter, Extract,</td>
</tr>
<tr>
<td></td>
<td>• Identify cash payment &gt; 10000</td>
<td>• Export, analytical tool</td>
</tr>
<tr>
<td></td>
<td>• Review of TDS compliance</td>
<td>• Join, Relation, Sort, analytical tool</td>
</tr>
<tr>
<td></td>
<td>• Analysis of Inventory</td>
<td>• Aging, MIS, Summarization, Count, Total</td>
</tr>
<tr>
<td>Financial audit</td>
<td>• Review of Authorization</td>
<td>• Filter, Extract, Summarization,</td>
</tr>
<tr>
<td></td>
<td>• Review of discount policy</td>
<td>• Stratification, Summarization,</td>
</tr>
<tr>
<td></td>
<td>• Compliance with tax rates – sales tax, excise duty, etc</td>
<td>• Join, relation, filter, extract, sort, count, Total, Compare files</td>
</tr>
<tr>
<td></td>
<td>• Verification of financial accuracy</td>
<td>• Summarization, Sort, Statistics, Total, Summarization</td>
</tr>
<tr>
<td></td>
<td>• Aging of debtors</td>
<td>• Aging, MIS, Periodicity check</td>
</tr>
<tr>
<td>Internal Audit</td>
<td>• Overall statistical analysis</td>
<td>• Statistics, Benford’s Law, Summarization</td>
</tr>
<tr>
<td></td>
<td>• Identification of exception items</td>
<td>• Benford’s Law, Filter, Gap detection</td>
</tr>
<tr>
<td></td>
<td>• Duplicate payment for invoices</td>
<td>• Identify duplicates, Sequence,</td>
</tr>
<tr>
<td></td>
<td>• Debtors outstanding &gt; credit period</td>
<td>• Join, Relation, Aging, Summarization,</td>
</tr>
<tr>
<td></td>
<td>• Age-wise analysis of debtors</td>
<td>• Aging, MIS, Summarization, Statistics</td>
</tr>
<tr>
<td></td>
<td>• Age-wise analysis of inventory</td>
<td>• Aging, MIS, date difference, total, count</td>
</tr>
<tr>
<td>Purchases</td>
<td>Payroll</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------</td>
<td></td>
</tr>
<tr>
<td>• Duplicate payments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Invalid vendors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Duplicate invoices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Invalid purchases</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Payments without receipt of goods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Inflated prices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Excess quantities purchased</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Join, Relation, Sort, Summarization, Filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Join, Relation, Sort, Summarization, Filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Join, Relation, Compare files, Filter, Exceptions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Filter, Sort, Stratify, Join, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Filter, Sort, Stratify, Join, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Join, Relation, Summarization, Sort, Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Re-computation using analytical tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Identify duplicates, Sequence, Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Re-computation using analytical tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Re-computation using analytical tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Join, Relation, Summarization, Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Join, Relation, Filter, Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Summarization, Statistics, MIS, Join, Relation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### QUESTIONS

1. Computer Assisted Audit Techniques (CAATs) refers to using _________ for increasing the effectiveness and efficiency of auditing.
   A. Technology
   B. Standards
   C. Documentation
   D. Systematic Process

2. Which of the following statements pertaining to CAAT is true?
   A. CAATs are specialist tools designed for use by specialist IT auditors
   B. CAATs refer to common techniques which can be easily mastered to audit in a computerized environment.
   C. CAATs cannot be used for compliance audit.
   D. CAATs can be used for Information systems audit only.

3. Which of the following statements pertaining to CAAT is incorrect?
   A. CAATs are tools for drawing inferences and gathering relevant and reliable evidence.
   B. CAATs provide direct access to electronic information
   C. CAATs provide a mechanism to gain access and to analyze data
   D. CAATs techniques are available in general audit software only.

4. Which of the following is a key reason for using CAATs?
   A. Availability of input documents
   B. Availability of visible paper trail.

---

Aging | Overdue A.R. and A.P. | Favorable credit terms | Inventory turnover rates | Dormant accounts | Records with future, blank or otherwise invalid dates | Items past a cutoff date | Contracts awarded before contract date | Transactions outside of billing period | Length in days of various activities | Aging, MIS, date difference | Aging, MIS, date difference, | Stratify, Filter, Summarization, | Identify Gaps, date difference | Sort, Filter, Sequence, Identify Gaps, Date difference, Statistics, Count | Date difference, Aging, Filter, | Date difference, Aging, Filter, Summarization, Statistics | Aging, Filter, Summarization, Statistics | Date difference, Aging, Filter, Summarization,...
C. Need to access information from systems having different IT environments.
D. Need to perform audit in lesser time.

5. Which of the following is a not a benefit of using CAATs?
   A. Increased audit quality and comply with auditing standards.
   B. Identify materiality, risk and significance in an IT environment.
   C. Identifying fraud in audit area.
   D. Ensuring better audit planning and management of audit resources.

6. For effective use of CAATs it is necessary to:
   A. Understand programming language of the software being audited.
   B. Obtain source code of the software being audited.
   C. Obtain copy of data in their original format for independent analysis.
   D. Audit data is retained in the original database format.

7. Which of the following step is important for validating the correctness of data obtained for audit?
   A. Issue a letter for getting the requested data in specified form as per the audit objectives.
   B. Obtain copies of record layout and definitions of all fields.
   C. Ensure that auditor has an overall understanding of the data.
   D. Compare the imported in audit software with control total of key data.

8. File access in CAATs refers to the capability of:
   A. Reading of different record formats and file structures.
   B. Conversion of data to common formats of data
   C. Using the import/ODBC function.
   D. Reviewing access controls of application software.

9. File reorganization feature of CAATs refers to:
   A. Functions provide auditor with an instant view of the data from different perspectives.
   B. Indexing, sorting, merging, linking with other identified files.
   C. Using of global filter conditions to select required data based on specified criteria.
   D. Use of sampling, stratification and frequency analysis.

10. Which of the following statement pertaining precautions in using CAATs is incorrect?
    A. Collect the relevant and correct data files.
    B. Identify all the important fields that need to be accessed.
    C. Ensure the data represents audit universe correctly and completely.
    D. Information provided by CAATs is clear indicator of problems and no further testing is required.
11. Which of the following is the first step in using CAATs?
   A. Identify the sources of data from the enterprise information system/application software.
   B. Based on scope and objectives of audit, decide about the need and extent to which CAAT could be used.
   C. Identify the critical data which is being audited as per audit scope and objectives.
   D. Identify the relevant personnel responsible for the data and information system.

12. Which of the following is the final step in using CAATs?
   A. Obtain and review documents relating to data/information systems.
   B. Decide what techniques of CAATs could be used as relevant to the environment
   C. Prepare a detailed plan for analyzing the data.
   D. Perform relevant tests on audit data as required and prepare audit findings and include in audit report/opinion.

**ANSWERS**

1. A  
2. B  
3. D  
4. C  
5. C  
6. C  
7. D  
8. A  
9. B  
10. D  
11. B  
12. D

**REFERENCES**

Below are sample list of references:

www.icai.org
www.isaca.org
www.auditnet.org
www.caseware-idea.org
www.acl.com
www.theiia.org
LEARNING OBJECTIVES

- Using advance extraction options for performing complex data analytics
- Understanding use of Benford’s Law in audit
- Understanding advance @ functions
- Learning Macro and IDEA Script

4.1 EXTRACTING SELECTED RECORDS

IDEA offers various options under “Extract Group” to select and extract records from databases and perform further analysis and/or to save the extracted records as a separate database.

Following options are provided in IDEA to perform extraction under analysis menu:

1. Direct
2. Key Value
3. Top Records
4. Indexed
5. Duplicate Key Detection
6. Gap Detection

4.1.1 DIRECT

Direct function is used to extract records by writing one more criteria’s and allows to save result in new database. With direct function we can provide one or more criteria to extract records.

Step by step procedure for using direct function

Open database ‘sales register’. Total records in the file are 999 and extract records having INV_ AMOUNT more than 10000.
Step 1: Analysis > Direct

Step 2: Records to extract:
- All – select this option to apply criteria on entire databases
- Range - select this option to apply criteria on range of records and not on entire database

Step 3: Database order: Criteria can be defined on indexed or sorted database by using this option.

Step 4: File Name: Give new file name

Step 5: Click on equation editor icon to define the criteria

Step 6: Define the criteria on equation editor screen INV_AMOUNT > 10000 and click on validate and exit option or press Ctrl+U
Step 7: Click on OK

Fig. 4.1.1.3 – Define criteria

Fig. 4.1.1.4 – Save file

Fig. 4.1.1.5 – Result saved in new database
Step 8: A new file meeting the criteria is created with 896 records out of 999 having invoice amount more than 10000

Note: Any Criteria defined in the equation editor can be saved by using Ctrt+S and it will save in Equations.ILB folder of the active project. Same can be reused as it is or edited for further references.

Exercises: 1. Extract records of sales of more than or equal to 15000 in East region
Exercises: 2. Extract records of mobile phones sold to foreign customer

4.1.2 KEY VALUE

Helps to extract records using values in a key. This function is used to extract records for a key in the entire database in a separate file. Key can be a region, product, zone, customer type, etc.

Step by step procedure for using Key Value function

Open database ‘sales register’. Total records in the file are 999.

Extract records based on key being ‘Product’

Step 1: Analysis > key Value

Step 2: Click on … besides ‘Existing keys:’ option

Fig. 4.1.2.1 Using Key Value

Step 2: Click on … besides ‘Existing keys:’ option
Step 3: Select product from the drop down option and click on OK

Step 4: Idea will extract all the key values and will pop up under group found section.

Step 5: Under output section, provide file name under ‘prefix for each database’

Step 6: Click on create a separate database for each unique key, will extract all the records for each unique key in a separate database.

Step 7: Use criteria tab to define condition for extraction of records key wise.
Fig. 4.1.2.4 Key wise databases created

Key value wise databases (product wise in our example) created and saved under master file sales register

Exercises: 1. Extract records based on key value region for south and east region
Exercises: 2. Extract records based on EXEC_NAME for INV_AMOUNT exceeding 25000

4.1.3 TOP RECORDS:

Helps to extract specified number of top or bottom records for each value of key in a separate database.

Step by step procedure for using Top Records function

Open database ‘sales register’. Total records in the file are 999.

Extract 3 top records INV_AMOUNT wise for each of the customer type

Step 1: Analysis > Top Records
Step 2: Number of records to extract - type 3
Step 3: Type: select Top records. Selecting bottom records option here will extract bottom records for the given conditions
Step 4: Top Records for: select INV_AMOUNT
Step 5: Under Group (optional): select Customer Type
Step 6: File name – type Top 3 invoices customer wise
Step 6: Enter condition to apply the function on selected type of records

Exercise 1: extract bottom 5 invoices for each executive
Exercise 2: extract top 3 invoices above Rs15,500/- for each region

4.1.4 INDEXED
Uses an existing index to narrow down the scope of extraction. This function can be used along with criteria to get better results. It combines the capabilities of criteria and range extraction.

Step by step procedure for using indexed function
Open database ‘sales register’. Total records in the file are 999.
Extract records of INV_AMOUNT >= 15000 having invoice date between 3rd April to 10th April, 2014
Step 1: Analysis > Indexed
Step 2: Field- select INVOICE_DATE
Step 3: Value is - => 20140403 (YYYYMMDD)
Step 4: and - <= 20140410
Step 5: Criteria – INV_AMOUNT > 15000
Step 6: File Name - Invoices above 15K between 3 to 10 April. 16 records extracted as an output

![Index function options](image1)

**Fig: 4.1.4.1 – Index function options**

![Output of indexed function](image2)

**Fig: 4.1.4.2 – Output of indexed function**

Exercise 1: Extract invoices for west region having invoice value between 8000 to 48000

Exercise 2: Extract invoices for product refrigerator having invoice date between 21st March to 28th March 2014

### 4.1.5 DUPLICATE KEY

IDEA includes two tasks for finding duplicate records:

**Duplicate Key Detection**: Identifies all records where the fields in the key are identical.

**Duplicate Key Exclusion**: Similar to Duplicate Key Detection, except that an additional field must be different.

The Duplicate Key Detection task is used to identify duplicate items in a database. Duplicate Key Detection can be used to identify duplicates within a single field, such as duplicate invoice numbers, or duplicates in a combination of up to eight fields.

Duplicate Key Detection can also be used to identify non-duplicate keys (unique) by selecting the **Output Records without duplicates** option instead of the default Output Duplicate Records option.
The Duplicate Key Exclusion task is used to identify duplicate items in a database where a further specified field is different, such as duplicate purchase order numbers for different suppliers. Duplicate Key Exclusion can be used to identify duplicates within a single field, such as duplicate invoice numbers, or duplicates on a combination of up to eight fields.

**Step by step procedure for Duplicate Key Detection**

The director of accounting is worried that several invoices may have been issued on multiple occasions. Verify if there have been any multiple invoices issued.

Step 1: Ensure that **Sales Transactions-Database** is the active database and that the **Data** property is selected in the **Properties** window.

Step 2: On the **Analysis** tab, in the **Explore** group, click **Duplicate Key** and then click **Detection**. Alternatively, from the Quick Access Toolbar, click the **Duplicate Key Detection** button.

![Fig: 4.1.5.1 Duplicate Key Detection dialog box.](image)

Step 3: Click **Key**. The **Define Key** dialog box appears.

Step 4: Select **INV_DATE** in **Ascending** order and then select **CUSTOMER_NO** in **Ascending** order.

![Fig 4.1.5.2 Define Key dialog box.](image)

Step 5: Click **OK**.

Step 6: Accept the default selection of the **Output duplicate records** option.

Step 7: In the **File name** field, enter **Duplicate Customer Sales on Same Day** as in Fig 4.1.5.3.

![Fig 4.1.5.3 Output duplicate records](image)
Step 8: Click OK.

The new database, containing all transactions with sales to the same customer more than once on the same day, is created and opened in the Database window.

Step 9: Ensure that **Duplicate Customer Sales on Same Day** is the active database and that the **Data** property is selected in the **Properties** window.

Step 10: On the **Analysis** tab, in the **Tasks** groups, click **Re-run**.

Step 11: In the Duplicate Key Detection dialog box, click **Key**.

Step 12: Add **QTY** as the third key.

Step 13: Click **OK**.

Step 14: In the Duplicate Key Detection dialog box, enter **Refined Duplicate List** as the file name.

Step 15: Click **OK**.

The new database of potential duplicates is displayed.

**Step by step procedure for searching Non-Duplicate records**

While Duplicate Key Detection allows you to search for more than one occurrence of a record, IDEA allows you to extract records that appear only once in the database.

Step 1: Ensure **Sales Transactions-Database** is the active database and that the **Data** property is selected in the **Properties** window.

Step 2: On the **Analysis** tab, in the **Explore** group, click **Duplicate Key** and then click **Detection**. Alternatively, from the Quick Access Toolbar, click the **Detection** button.

Step 3: Select the **Output records without duplicates** option.

Step 4: In the **File name** field, enter **Non-Duplicate Invoices**
Fig 4.1.5.4 Select Non-Duplicate records

Step 5: Click Key and ensure INV_NO is selected as your first field in Ascending order.

Step 6: Click OK.

Step 7: In the Duplicate Key Detection dialog box, click OK.

The new Non-Duplicate Invoices database will appear.

4.1.6 Gap Detection

The Gap Detection task identifies “missing” information in a database. The results of the test are displayed in the Results area of the Properties window. The Gap Detection task can be used on a Numeric field to identify gaps in a numerical sequence, such as missing check numbers in a payment database.

The Gap Detection task can be used on a Character field to identify gaps in a numerical sequence that is a sub-set of a Character field. For example, it may be used to identify gaps in an invoice number sequence for each store where the first character of the invoice number identifies the store (for example, A123).

The Gap Detection task can be used on a Date field to identify gaps in a range of dates. It provides the option for excluding weekends and/or holidays, if required. For example, you can identify sales transactions that occurred on non-working days.

Step by step procedure for Gap Detection on Numeric field

Step 1: Ensure that Sales Transactions-Database is the active database and that the Data property is selected in the Properties window.

Step 2: On the Analysis tab, in the Explore group, click Gap Detection. Alternatively, on the Quick Access Toolbar, click the Gap Detection button.

Fig 4.1.6.1 Gap Detection dialog box

Step 3: From the Field to use drop-down list, select INV_NO.

Select a range of values to test. By default, all values in the INV_NO field are tested and the minimum and maximum values are listed as starting key value and Ending key value. You can also change
the Gap Increment to set a minimum gap to report.
Step 4: Ensure the **Create result** check box is selected.
Step 5: In the **Result name** field, enter **Missing Invoice Sequences**.

![Fig 4.1.6.2 Result name.](image)

Step 6: Click **OK**.
The result will display the ranges of invoice numbers that are missing in the database. Note that each row shows a gap. Certain rows may contain more than one missing invoice number.

![Fig 4.1.6.3 Result Missing invoice](image)

**Step by step procedure for Gap Detection on Date field**

Step 1: Ensure that **Sales Transactions-Database** is the active database and that the **Data** property is selected in the **Properties** window.

Step 2: On the **Analysis** tab, in the **Explore** group, click **Gap Detection**.
The Gap Detection dialog box appears.

Step 3: In the **Field to use** drop-down list, select **INV_DATE**.
You can select a range of dates to test. By default, all values in the INV_DATE field are tested and the minimum and maximum values are listed as Starting date value and Ending date value. You can also choose to ignore weekends and holidays that are set by the user.

Step 4: Accept the default option to test all values.

Step 5: Select the **Ignore weekends** check box.

Step 6: Select the **Ignore holidays** check box and click **Set Holidays**.

The Set Holidays dialog box appears.

Step 7: From the Set Holidays toolbar, click the New button.

Step 8: Click the **Browse** button to select December 25 and December 26 (Fig 4.1.6.4).

Step 9: Click **OK**.

Step 10: In the Gap Detection dialog box, ensure the **Create result** check box is selected.

Step 11: In the **Result name** field, enter **Missing Dates**.

![Fig 4.1.6.4](image-url)  
*Fig 4.1.6.4*

![Fig 4.1.6.5 Missing Dates](image-url)  
*Fig 4.1.6.5 Missing Dates*
Step 12: Click OK.

The result will display the ranges of dates that are missing in the database. Note that each row shows a gap. Certain rows may contain more than one missing date.

![Resulted Database Image]

**4.2 DEFINE ACTION FIELD**

Defines a custom action to be executed when the field is selected. It allows looking up for the corresponding values in the linked file. For example, you have two files. Account details file has details such as account number, branch code, status of the account and balance amount in the account. Customer details file has details such as account number, customer name, address, etc. This function will allow you to perform a defined task on common fields present in both the files in our case account number and will fetch the data from another file corresponding to the selected field.

**Step by step procedure for using indexed function**

Open databases ‘Account Details’ and ‘customer details’

Define action on filed ACC_NO

Step 1: Data > Define Action Field

Step 2: Click on column ACCNO heading to select entire column
Fig: 4.2.1 – Define Action Field Options

Step 3: Click on … besides file name ‘account details.imd’ and select ‘customer details.imd’ from the list

Fig. 4.2.2 – Create Action link

Step 4: Select look up field option and click select field ACCNO and click on OK
Fig. 4.2.3 – Select Lookup field

Step 5: Click ok on the Define action field dialogue box

Step 6: Observe ACCNO field. Filed converted into a hyperlink type filed.

Step 7: Click on any record in accno field to see the corresponding information available for the selected ACCNO in the customer detail file. Refer Fig. 4.2.4 below for results

Fig. 4.2.4 – Result of define action field

Step 8: To remove the link created by using define action field, go the option and select remove action link option
4.3 SEARCH

IDEA offers a variety of search options such as character, date, numeric field search. In character based search feature a particular word is searched in multiple databases and within databases in a particular field. This option is mostly used for character intensive databases having multiple character fields. IDEA offers an option of saving the search result in a separate database for future reference.

IDEA offers advanced search features as under:

Boolean Search: *Boolean Search* is a type of search allowing users to combine keywords with operators such as AND, NOT and OR to produce more relevant results.

Wildcard Search: *Wildcard searches* are not simply exact string matches, but are based on character pattern matching between the characters specified in a query and words in documents that contain those character patterns.

Proximity Search: *Proximity search* looks for documents where two or more separately matching term occurrences are within a specified distance, where distance is the number of intermediate words or characters.

**Step by step procedure for using search function**

Step 1: Data > Search

Step 2: In ‘text to find’ - type the word to search

Step 3: Tick case sensitive for more accurate search result

Step 4: Tick ‘Create an extraction database’ and provide file name to save search result

Step 5: Under scope option select field in which search word will be searched

Step 6: Click on … besides ‘look for other databases’ to look for search word in other databases
Fig: 4.3.2 Search options

Fig: 4.3.3 Search result
By clicking on the respective record in search window Fig 4.3.4. IDEA opens the database and points the cursor to that record in the database

4.4. BENFORD’S LAW

Benford’s Law, also called the first-digit law, was made famous in 1938 by Physicist Frank Benford, who after observing sets of naturally occurring numbers, discovered a surprising pattern in the occurrence frequency of the digits one through nine as the first number in a list. In essence, the law states that in numbered lists providing real-life data (e.g., a journal of cash disbursements and receipts, contract payments, or credit card charges), the leading digit is one almost 33 percent (i.e., one third) of the time. On the other hand, larger numbers occur as the leading digit with less frequency as they grow in magnitude to the point that nine is the first digit less than 5 percent of the time.

In the 1970s, Hal Varian, a professor at the University of California’s Berkeley School of Information, suggested that the law could be used to detect possible fraud in lists providing socioeconomic information. Since then, Benford’s law has been applied to large numbers of data to detect unusual patterns that are often the result of errors or, worse, fraud. As part of their work, internal auditors often employ tools and scientific methods that enable them to detect instances of fraud.

Benford’s law states that if there is a set of non-manipulated, naturally occurring numbers, the occurrence frequency of digits one through nine as the first digit should be expected. As we can see from the numbers in table 1, naturally 30 percent of numbers have one as a leading digit, and nine occurs as a leading digit only one time in twenty. Because most financial and accounting data conform to naturally occurring numbers, by comparing the occurrence frequency of these first digits to Benford’s pattern, auditors should be able to determine irregularities and possible manipulations.

In 1938, the research and calculations were performed manually, which was painstaking. Today, with computing power and the ease of accessing big data sets, one can see that Benford’s Law of expected numbers is valid. It tests data such as Twitter users by followers’ count, most common
iPhone passcodes, population of Indian cities, government spending, and even includes the first 652,066 Fibonacci numbers.

The expected values for any data set of the first leading digit and also for the first two leading digits are outlined in Table 6.1 below. For the first digit test, the first leading digit output is depicted in the graph in Fig 4.4.1. For example, the leading digit 1 appears 30 percent of the time, whereas the leading digit 9 appears 4.6 percent of the time. The bars are the actual data counts and the lines are the lower and upper boundaries along with the expected count. This data set conforms to Benford’s Law.

<table>
<thead>
<tr>
<th>First Digit Frequency</th>
<th>Second Digit Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>0.30103</td>
</tr>
<tr>
<td>2</td>
<td>0.17609</td>
</tr>
<tr>
<td>3</td>
<td>0.12494</td>
</tr>
<tr>
<td>4</td>
<td>0.09691</td>
</tr>
<tr>
<td>5</td>
<td>0.07918</td>
</tr>
<tr>
<td>6</td>
<td>0.06695</td>
</tr>
<tr>
<td>7</td>
<td>0.05799</td>
</tr>
<tr>
<td>8</td>
<td>0.05115</td>
</tr>
<tr>
<td>9</td>
<td>0.04576</td>
</tr>
</tbody>
</table>

Table 4.4.1: Benford’s Law First Digit Frequency and First Two Digits Frequency

Following are the assumptions of Benford’s Law: The numbers in the data set should describe the same object

There should be no built-in maximum or minimum to the numbers

The numbers should not be assigned, such as telephone numbers, bank account numbers, social insurance, or social security numbers

Does not apply to uniform distributions such as lottery balls where the uniform balls are selected and not the actual numbers

Primary Benford’s Law tests are the first digit, first two digits, first three digits, and second digit tests. Advanced Benford’s Law tests are summation and second order. Associated tests are last two digits, number duplication, and distortion factor model. All but the last two tests can be automatically executed from within the IDEA software.

The number duplication test identifies specific numbers causing spikes or anomalies in primary and summation tests. Spikes in the primary tests are caused by some specific numbers occurring abnormally too often. Abnormally large numbers in value cause spikes in the summation test.

The distortion factor model shows whether the data has an excess of lower digits or higher digits. It assumes that the true number is changed to a false number in the same range or percentage as the true number.

Most presentations and articles discuss using Benford’s Law to detect numbers near their authorization limits. For example, if someone’s authorization limit is Rs. 10,000, then many first two
digits in the 99, 98, and 97 areas will be detected using Benford’s Law if they are trying to maximize
authorizing expenditures.

Another example Bank has the limit upto Rs 50,000/- for cash deposit without PAN. For money
laundering transactions one will deposit just below the limit i.e Rs 49,999/- or Rs 49,500/-. Then last
two digits in the 99 and 00 will be detected using Benford’s law last two digits in IDEA.

Some other practical applications include:

- Accounts payable (expenses) data
- Estimations (accruals) in the general ledger
- Sales
- Purchases
- Non-arm’s-length transactions
- Customer refunds
- Bad debts
- Anti-money laundering

**BENFORDS’S LAW IN IDEA**

The Benford’s Law feature in IDEA can provide a valuable reasonableness test for large data sets.
IDEA only tests positive numbers 10 and over in the data file. For negative numbers, values greater
than minus 10 are excluded (exclude −9, −8, . . . −1). These steps eliminate immaterial items from the
analysis. Positive and negative numbers are analyzed separately.

The positive and negative numbers are evaluated on their own due to the fact that positive
numbers behave very differently from negative numbers. For example, where positive earnings are
manipulated for management bonuses, there is motivation to increase the earnings, moving away
from zero toward larger numbers. Where there are losses and management wishes to improve stock
prices, there is incentive to move the larger negative number to a smaller one toward zero.

IDEA can apply most of the Benford’s Law tests and can also display suspicious results in graphical
format. Tests provided in IDEA are the first digit, first two digit, first three digits, second digit, last
two digits, second order, and summation tests as shown in Fig 4.4.2

Analysis > Benford’s Law
Fig: 4.4.2 Applying Benford’s Law feature in IDEA

Fig: 4.4.3 Benford’s first digit test
This first two-digit primary test output from IDEA indicates that it does not conform in Fig 4.4.4. The graph highlights the three most highly suspicious numbers and the three most suspicious items. By placing the cursor over highly suspicious bar numbers 21, 70, 99, options for extracting or displaying the records are offered. Field statistics may also be displayed.

**Conclusion**

Benford’s analysis, when used correctly, is a powerful tool for identifying suspect accounts or amounts for further analysis. Benford’s analysis is a tool to complement additional tests/tools. Benford’s Law is a wonderful tool for initial risk assessment of the contents of a data set. It provides the auditor or investigator with a good starting point. The user must understand the business and the industry to effectively use this tool. Knowledge of the business can quickly eliminate false positives.

**4.5 ADVANCE @ FUNCTIONS**

Step by Step procedure for accessing @Functions:

- Open any database
- In the Properties window, click Criteria.
- The Equation Editor is displayed. The right-side of the window displays the list of available @Functions and displays the help page for selected @Functions.
- Expand a category node to display a list of associated @Functions.
- Select a @Function from the list to display its information.

IDEA provides more various @Functions for performing operations such as date arithmetic, financial and statistical calculations as well as text searches.

The @Functions are accessed through the Equation Editor. Quick help including the syntax, a description, and an example of use for each of the @Functions is available when the @Function is highlighted in the list in the Equation Editor window.

Read the definitions below to become familiar with each @Function:

- Numeric expression: Refers to a number or an equation that evaluates to a number.
- String: A series of characters, such as a name or address.

<table>
<thead>
<tr>
<th>@Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Abs</td>
<td>Returns the absolute value of a numeric expression.</td>
</tr>
<tr>
<td>@Afternoon</td>
<td>Returns 0 if time is in the AM and 1 if time falls in the PM and -1 for an invalid time.</td>
</tr>
<tr>
<td>@Age</td>
<td>Calculates the number of days between two dates.</td>
</tr>
<tr>
<td>@AgeDateTime</td>
<td>Returns the number of seconds between two dates and times.</td>
</tr>
<tr>
<td>@AgeTime</td>
<td>Returns the number of seconds between two times.</td>
</tr>
<tr>
<td>@AllTrim</td>
<td>Removes all leading and trailing spaces.</td>
</tr>
<tr>
<td>@Ascii (not in IDEA Unicode)</td>
<td>Provides the ASCII value of a character.</td>
</tr>
<tr>
<td>@Between</td>
<td>Determines if a numeric expression falls within a specific range.</td>
</tr>
<tr>
<td>@BetweenDate</td>
<td>Returns a number indicating whether a date value falls within a specified range (1) or not (0).</td>
</tr>
<tr>
<td>Function Description</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>@BetweenTime</td>
<td>Returns a number indicating whether a time value falls within a specified range (1) or not (0).</td>
</tr>
<tr>
<td>@Bit</td>
<td>Identifies a bit value.</td>
</tr>
<tr>
<td>@BitAnd</td>
<td>To mask out unwanted bits.</td>
</tr>
<tr>
<td>@BitOr</td>
<td>To set required bit.</td>
</tr>
<tr>
<td>@Chr (not in IDEA Unicode)</td>
<td>Provides the character equivalent of a specified ASCII code.</td>
</tr>
<tr>
<td>@CompareNoCase</td>
<td>Ignores uppercase letters when comparing expressions.</td>
</tr>
<tr>
<td>@CompIf</td>
<td>Determines if a record satisfies multiple criteria.</td>
</tr>
<tr>
<td>@Ctod</td>
<td>Converts character dates to IDEA Date Format.</td>
</tr>
<tr>
<td>@Ctot</td>
<td>Converts character time values to IDEA Time Format.</td>
</tr>
<tr>
<td>@CurForm</td>
<td>Converts numeric value into a formatted text.</td>
</tr>
<tr>
<td>@CurVal</td>
<td>Converts formatted character fields to numeric fields.</td>
</tr>
<tr>
<td>@Date</td>
<td>Returns the present date.</td>
</tr>
<tr>
<td>@Day</td>
<td>Returns the day in a date expression.</td>
</tr>
<tr>
<td>@DaysToD</td>
<td>Converts a number of days since Jan. 1, 1900 to date format.</td>
</tr>
<tr>
<td>@Db</td>
<td>Calculates the fixed declining-balance depreciation for a specified period.</td>
</tr>
<tr>
<td>@Ddb</td>
<td>Calculates double declining-balance depreciation.</td>
</tr>
<tr>
<td>@Delete</td>
<td>Deletes a specified number of characters from a string.</td>
</tr>
<tr>
<td>@Dow</td>
<td>Returns the day of the week.</td>
</tr>
<tr>
<td>@Dtoc</td>
<td>Converts date expressions to character.</td>
</tr>
<tr>
<td>@DToDays</td>
<td>Reveals the number of days between Jan. 1, 1900 and a specified date.</td>
</tr>
<tr>
<td>@Dtoj</td>
<td>Converts dates to Julian format.</td>
</tr>
<tr>
<td>@Exp</td>
<td>Calculates the exponent of a numeric expression.</td>
</tr>
<tr>
<td>@FieldStatistics</td>
<td>Returns the numeric value for a specified field statistic.</td>
</tr>
<tr>
<td>@FindOneOf</td>
<td>Finds the position of the first matching character in 2 strings.</td>
</tr>
<tr>
<td>@FinYear</td>
<td>Returns the financial year for a given date based on the year end.</td>
</tr>
<tr>
<td>@Format12hourClock</td>
<td>Returns a string representing time formatted as HH:MM:SS TT.</td>
</tr>
<tr>
<td>@Fv</td>
<td>Calculates the future value of an investment.</td>
</tr>
<tr>
<td>@GetAt</td>
<td>Returns the character that appears in a specified numeric position.</td>
</tr>
<tr>
<td>@GetNextValue</td>
<td>Returns the next value in the selected field.</td>
</tr>
<tr>
<td>@GetPreviousValue</td>
<td>Returns the previous value in the selected field.</td>
</tr>
<tr>
<td>@Hours</td>
<td>Returns the hours portion of a given time.</td>
</tr>
<tr>
<td>@If</td>
<td>Allows a choice of two results based on the evaluation of a condition.</td>
</tr>
<tr>
<td>@Insert</td>
<td>Inserts a string into an existing string.</td>
</tr>
<tr>
<td>@Int</td>
<td>Returns the integer portion of a numeric value.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>@Ipmt</td>
<td>Calculates the interest payment for a given period.</td>
</tr>
<tr>
<td>@Irr</td>
<td>Calculates internal rate of return.</td>
</tr>
<tr>
<td>@IsBlank</td>
<td>Tests if a character field is blank.</td>
</tr>
<tr>
<td>@IsFieldDataValid</td>
<td>Returns a 1 if the data in the field is valid or a 0 if the data is invalid.</td>
</tr>
<tr>
<td>@Isin</td>
<td>Returns the starting position of a string within another string (case-sensitive).</td>
</tr>
<tr>
<td>@Isini</td>
<td>Returns the starting position of a string within another string (NOT case-sensitive).</td>
</tr>
<tr>
<td>@Jtod</td>
<td>Converts Julian dates to IDEA Date Format.</td>
</tr>
<tr>
<td>@JustLetters</td>
<td>Returns a string with all the numeric characters removed.</td>
</tr>
<tr>
<td>@JustNumbers</td>
<td>Returns all the numbers (leading and trailing).</td>
</tr>
<tr>
<td>@JustNumbersLeading</td>
<td>Returns the leading numbers.</td>
</tr>
<tr>
<td>@JustNumbersTrailing</td>
<td>Returns the trailing numbers.</td>
</tr>
<tr>
<td>@LastDayofMonth</td>
<td>Returns the last day for any given month and year combination.</td>
</tr>
<tr>
<td>@Left</td>
<td>Returns the specified leftmost characters in a string.</td>
</tr>
<tr>
<td>@Len</td>
<td>Returns the length of a string, including any trailing spaces.</td>
</tr>
<tr>
<td>@List</td>
<td>Determines which criteria in a list of values is met by an expression.</td>
</tr>
<tr>
<td>@Log</td>
<td>Calculates natural logarithms.</td>
</tr>
<tr>
<td>@Log10</td>
<td>Calculates logarithm 10x.</td>
</tr>
<tr>
<td>@Lower</td>
<td>Converts all characters in a string to lower-case.</td>
</tr>
<tr>
<td>@Ltrim</td>
<td>Removes leading spaces from a string.</td>
</tr>
<tr>
<td>@Match</td>
<td>Determines which criteria in a list of values are met by an expression.</td>
</tr>
<tr>
<td>@Max</td>
<td>Returns the greater value of two numeric expressions.</td>
</tr>
<tr>
<td>@Mid</td>
<td>Extracts a portion of text from within a string.</td>
</tr>
<tr>
<td>@Min</td>
<td>Returns the smallest value of two numeric expressions.</td>
</tr>
<tr>
<td>@Minutes</td>
<td>Returns the minutes portion of a given time.</td>
</tr>
<tr>
<td>@MIRR</td>
<td>Calculates modified internal rate of return.</td>
</tr>
<tr>
<td>@Month</td>
<td>Returns the month in a date expression.</td>
</tr>
<tr>
<td>@NoMatch</td>
<td>Determines if an expression meets none of the criteria in a list of values.</td>
</tr>
<tr>
<td>@Npv</td>
<td>Calculates the net present value of an investment.</td>
</tr>
<tr>
<td>@Ntod</td>
<td>Converts a numeric expression into an IDEA date format.</td>
</tr>
<tr>
<td>@Ntot</td>
<td>Converts a numeric expression into an IDEA time format.</td>
</tr>
<tr>
<td>@Pmt</td>
<td>Calculates a loan payment.</td>
</tr>
<tr>
<td>@Ppmt</td>
<td>Returns the Principal amount of a loan payment.</td>
</tr>
<tr>
<td>@Preco</td>
<td>Returns the physical record number.</td>
</tr>
<tr>
<td>@Proper</td>
<td>Capitalizes the first letter of each word in a string.</td>
</tr>
<tr>
<td>@Pv</td>
<td>Returns the present value of an investment.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>@Qtr</td>
<td>Returns 1-4 representing the quarter a given date falls in based on the specified year end.</td>
</tr>
<tr>
<td>@Random</td>
<td>Generates a random number.</td>
</tr>
<tr>
<td>@Rate</td>
<td>Calculates the interest rate of an investment/loan.</td>
</tr>
<tr>
<td>@Recno</td>
<td>Returns the logical record number (index-sensitive).</td>
</tr>
<tr>
<td>@RegExp</td>
<td>Matches character expressions using a complex set of rules.</td>
</tr>
<tr>
<td>@Remove</td>
<td>Eliminates all instances of a specified character.</td>
</tr>
<tr>
<td>@Repeat</td>
<td>Repeats the first character of a string a specified number of times.</td>
</tr>
<tr>
<td>@Replace</td>
<td>Replaces a string or substring with another.</td>
</tr>
<tr>
<td>@Reverse</td>
<td>Reverses the order of characters in a string.</td>
</tr>
<tr>
<td>@Right</td>
<td>Isolates the specified rightmost characters in a string.</td>
</tr>
<tr>
<td>@Round</td>
<td>Rounds to the nearest integer.</td>
</tr>
<tr>
<td>@Seconds</td>
<td>Returns the seconds portion of a given time.</td>
</tr>
<tr>
<td>@Seed</td>
<td>Sets the random-number seed.</td>
</tr>
<tr>
<td>@SimilarPhrase</td>
<td>Measures the similarity between two specified phrases or Character fields.</td>
</tr>
<tr>
<td>@SimilarWord</td>
<td>Measures the similarity between two strings (either single words or character expressions) or Character fields.</td>
</tr>
<tr>
<td>@SimpleSplit</td>
<td>Extracts a specified occurrence of a segment of a character string that resides between certain characters like hyphens or backslashes.</td>
</tr>
<tr>
<td>@Sln</td>
<td>Returns the straight-line depreciation of an asset.</td>
</tr>
<tr>
<td>@SpacesToOne</td>
<td>Strips spaces leaving only one space between words in a string.</td>
</tr>
<tr>
<td>@SpanExcluding</td>
<td>Returns the characters in a string that appear before any characters in a specified string.</td>
</tr>
<tr>
<td>@SpanIncluding</td>
<td>Returns the characters at the beginning of a string that match any character of a specified string.</td>
</tr>
<tr>
<td>@Split</td>
<td>Breaks a character string into segments separated by characters, such as spaces or commas, and returns a specified segment.</td>
</tr>
<tr>
<td>@Squrt</td>
<td>Calculates a square root.</td>
</tr>
<tr>
<td>@Str</td>
<td>Converts numeric expressions to strings.</td>
</tr>
<tr>
<td>@Stratum</td>
<td>Groups records by interval.</td>
</tr>
<tr>
<td>@Strip</td>
<td>Removes all spaces, punctuation and control characters.</td>
</tr>
<tr>
<td>@StripAccent</td>
<td>Removes an accent from an accented character.</td>
</tr>
<tr>
<td>@Syd</td>
<td>Returns the sum-of-years digit depreciation for an asset.</td>
</tr>
<tr>
<td>@Time</td>
<td>Returns the present time.</td>
</tr>
<tr>
<td>@Trim</td>
<td>Removes trailing spaces.</td>
</tr>
<tr>
<td>@Ttoc</td>
<td>Converts a time or number into a string with the HH:MM:SS format.</td>
</tr>
<tr>
<td>@Upper</td>
<td>Converts all characters in a string to upper-case.</td>
</tr>
<tr>
<td>@Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>@Val</td>
<td>Converts a character expression to numeric.</td>
</tr>
<tr>
<td>@Workday</td>
<td>Returns 1 if a given date falls between Monday-Friday and 0 if the date falls on a Saturday or Sunday.</td>
</tr>
<tr>
<td>@Year</td>
<td>Returns the year in a date expression.</td>
</tr>
</tbody>
</table>

Examples of @functions:

<table>
<thead>
<tr>
<th>@Function</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>@age</td>
<td>For creating a new virtual numeric field with difference in days.</td>
<td>To arrive at the difference in days between ‘Token Date’ and ‘Cheque Date’ in ANY BILL PAY FILE use @age(Cheque Date, Token Date) through Field Manipulation.</td>
</tr>
<tr>
<td>@dow</td>
<td>Identify transactions made on any day of the week.</td>
<td>To capture cheques issued on Sunday use @dow(Cheque Date)=1 through a Direct Extraction. (1 is Sunday in IDEA)</td>
</tr>
<tr>
<td>@isblank</td>
<td>Filter out blank character fields</td>
<td>To present blank scheme agency name in GRANTS-IN-AID data use @isblank(Scheme Name) through Criteria.</td>
</tr>
<tr>
<td>@isini</td>
<td>Capture specific character strings from a character or alphanumeric field where the strings are not case sensitive.</td>
<td>To display cheque in favour of field containing ‘Shri’ for cheques issued inter-se Government use @isini(&quot;shri&quot;, cheque in favour) through Direct Extraction.</td>
</tr>
<tr>
<td>@compif</td>
<td>Conditional function which allows for preparing an expression with multiple conditions and consequent results. Useful for ‘If then analysis’</td>
<td>In the Disaster Claim file, to create a new virtual numeric field titled Sanctioned Amount using @compif(Category=&quot;0400&quot;, 1000000, Category = “0104”, 100000, “0103”, 500000, 1, 0) through field manipulation. This means if the category is 0400 the result in the new virtual numeric field will be Rs. 1000000 and so on so forth.</td>
</tr>
<tr>
<td>@strip</td>
<td>Removes all special characters, spaces and punctuation marks from a character or alphanumeric field.</td>
<td>To check if the same sanction number has been used more than once in the same controller, DDO, PAO, Function Head and Scheme by inserting special characters or spaces to bypass inbuilt system checks. Append a virtual character field with criteria @strip(Sanction Number). Then run a duplicate key detection test on the stripped sanction number field.</td>
</tr>
<tr>
<td>@Function</td>
<td>Purpose</td>
<td>Example</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>@left</td>
<td>Removes the left most occurrences of a character field or alpha-numeric field.</td>
<td>In the Disaster Claim file, every case claim is administered a Unique Case ID of 12 digits. The first 4 digits represent the category of claim i.e. death (0400) etc. can be captured separately by creating a virtual character field using @left(Unique Case ID, 4) through Field Manipulation.</td>
</tr>
<tr>
<td>@right</td>
<td>Removes the right most occurrences of a character field or alpha-numeric field.</td>
<td>In the Disaster Claim file, every case claim is administered a Unique Case ID of 12 digits. The last 6 digits represent the claim serial number which can be captured separately by creating a virtual character field using @right(Unique Case ID, 6) through Field Manipulation.</td>
</tr>
<tr>
<td>@mid</td>
<td>Removes the central most occurrences of a character field or alpha-numeric field.</td>
<td>In the Disaster Claim, every case claim is administered a Unique Case ID of 12 digits. The 5th and 6th digit together represent the case tribunal number which can be captured separately by creating a virtual character field using @mid(Unique Case ID, 5,2) through Field Manipulation.</td>
</tr>
<tr>
<td>@year</td>
<td>Derives the year from any date field in IDEA</td>
<td>In Tax collection data via APPLICATION SYSTEM containing nodal scroll date, a new virtual numeric field can be created using @year(nodal scroll date) to derive the year of collection through Field Manipulation.</td>
</tr>
<tr>
<td>@month</td>
<td>Derives the month from any date field in IDEA</td>
<td>In Tax collection data via APPLICATION SYSTEM containing nodal scroll date, a new virtual numeric field can be created using @month(nodal scroll date) to derive the month of collection through Field Manipulation.</td>
</tr>
<tr>
<td>@day</td>
<td>Derives the day from any date field in IDEA</td>
<td>In Tax collection data via APPLICATION SYSTEM containing nodal scroll date, a new virtual numeric field can be created using @day(nodal scroll date) to derive the day of collection through Field Manipulation.</td>
</tr>
<tr>
<td>@DtoDays</td>
<td>Computes the days corresponding to any date field in IDEA</td>
<td>In Gradation List to compute the days for the retirement date add a virtual numeric field with criteria @DtoDays(Retirement Date) through Field Manipulation.</td>
</tr>
<tr>
<td>@Function</td>
<td>Purpose</td>
<td>Example</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>@DaystoD</td>
<td>Reverse compute Date from days.</td>
<td>In Gradation List to compute the date from the days add a <strong>virtual date field</strong> with criteria @DaystoD(Retirement Date Days+30) through Field Manipulation. The +30 is to suggest adding 30 grace-days as an example.</td>
</tr>
<tr>
<td>@match</td>
<td>Looks for multiple values in a field of choice through one single equation without the need of having to write multiple equations.</td>
<td>In Pension payments file to filter out basic pension equal to Rs. 3500 or Rs. 40000 (i.e. outliers) @match(Basic, 3500, 40000) can be run through a Direct Extraction. This avoids the need to run two equations like (Basic=3500) OR (Basic=40000)</td>
</tr>
</tbody>
</table>

### 4.6 CUSTOM FUNCTIONS

Custom functions are written using IDEAScript. Custom functions are used just like the included @Functions within IDEA. The custom functions are accessible in the equation editor in the same area as the @ functions. Custom functions are prefixed with the # symbol instead of the @ symbol as in regular functions and have the file extension .ideafunc. Custom functions should be located under the active IDEA project in the Custom Functions.ILB folder.

**Step by Step guide to write custom functions:**

1. Click on Criteria under properties section
2. Click on # sign in equation editor
3. Select new to create new custom function and define a short but very effective custom function as shown in Fig. 4.6.2 below.
The custom function converts a character field containing date information into a date field in IDEA format.

CDate is a Visual Basic function that converts a character date field into an actual date field that is usable by IDEA. For example, the function can convert April 26, 2017, Apr 26/17, or 4/26/17 characters to an IDEA date field in the default format of DD/MM/YYYY.

Step 4: Click on OK to save the custom function

The saved custom function is now available for use within equation editor dialogue box under custom function section as shown in Fig. 4.6.3
4.7 ADVANCE SAMPLING

Statistics involves the study of research designs, the collection of the data, describing the data, analyzing the data, and then forming a conclusion. We are interested mainly in the analysis of data that has already been collected by us from various business systems. We hope to be able to arrive at various conclusions after analyzing the data. Understanding some basic statistics allows you to understand the makeup or distribution of your data files. This is especially useful when your data file is large and contains millions of records. There are various types of statistical analysis, but the two major categories are descriptive statistics and inferential statistics.

SAMPLING

The act, process, or technique of selecting a representative part of a population for the purpose of determining parameters or characteristics of the whole population. Simply put, sampling is a process of selecting a subset of the population or a number of records from the data set for the purpose of making inferences or conclusions to the entire population or data set.

Audit sampling is the audit procedure of examining a portion of items within a class of transactions in order to evaluate one or more characteristics of that entire class. Either statistical or non-statistical sampling methods may be used against a part of the entire data set to make a conclusion regarding the entire data set.

Sampling is effective when the audit procedure or step does not require a 100 percent review of the population of the class, but a decision or conclusion is required and it is not cost effective to audit 100 percent of the transactions.

STATISTICAL SAMPLING

Statistical sampling uses statistical mathematical calculations for selecting and then evaluating a sample from the data set. Statistical sampling outlines in numeric terms the parameters and precision levels associated with the sample conclusion.

One such use of statistical sampling that we are most familiar with are polls used to determine candidate’s current standings in upcoming elections or in popularity surveys.

An auditor may be verifying an account balance through statistical sampling and conclude that it is Rs. 100,000 plus or minus 5 percent or Rs. 5,000 each way (Rs. 95,000 to Rs. 105,000), 19 times out of 20. The conclusion would be that given the precision of the sample at 5 percent (plus or minus Rs. 5,000), there is assurance that the balance is correct with a confidence level of 95 percent. In addition, if the materiality was predetermined to be at 7 percent, then it can be concluded that there would no material error based on the precision level.

Confidence level is the remaining factor when the acceptable sampling risk is eliminated. In the example of selecting a 95 percent confidence level, you allow only a 5 percent chance of getting the wrong sample that does not adequately represent the entire population.

Using statistical samples to obtain familiarity with the data set might be useful but if it is not used to reach a conclusion, it cannot be considered as part of the audit procedure. In addition to formulating a conclusion, statistical sampling must use statistical calculations, and the sample must be random.

Proper use of statistical sampling is beneficial because it:

- Requires a scientifically accepted and defined approach.
● Allows the auditor to maintain professional judgment in regard to audit risks and materiality.
● Displays the sample results and conclusion in relation to the selected data set population along with defined judgment selections.

NON-STATISTICAL SAMPLING

Non-statistical sampling does not involve the use of statistical calculations. It relies on the subjective sampling selections by the auditor and has less of a standardized approach. Non-statistical sampling is beneficial where:

● The auditor needs to employ professional subjective judgment.
● There is a unique issue where there is a need for a less rigid standardized approach.
● The auditor should not be restricted to explicit numbers as to materiality or risk.

In order to effectively perform non-statistical sampling, the auditor must have a good knowledge of the data set. Knowing the contents of the data or population allows for a supportable sample selection choice and also supports the conclusion of the results. Sample selection may be based on random sampling or other nonmathematical techniques such as judgmental, haphazard, or block selection.

Judgmental selection is frequently used when the auditor is very experienced and selects samples based on sound judgment. Typically, the auditor will make the selections based on a combination of representativeness of the population, value of the items, and relative risk of the items.

Haphazard selection is where the auditor picks items without basis of any mathematical formula. The auditor believes that the items selected are representative of the population and no intentional bias was applied to any of the included or excluded items.

Block selection is where a contiguous sequence of items is selected as samples. These blocks may be invoice numbers from 1000 to 1100 or a specific type of transactions for the month of March. Block selection effectiveness can be much improved by sampling several blocks.

SAMPLING RISK

The sample selected either through statistical sampling or non-statistical sampling methods might not truly reflect the population even if done with the utmost care. This is the cause for sampling risk, where the auditor’s conclusion based on the selected sample may differ from the reality of the conditions of the entire population of the data set. Sampling risk occurs due to limited time and resources that prevent an audit of the entire population.

Alpha or Type I risk is the risk of incorrect rejection. That is, the auditor incorrectly concludes from the sample that the population errors are worse than they actually are.

Beta or Type II risk is the risk of incorrect acceptance. That is, the auditor incorrectly concludes from the sample that the errors in the population are better than they really are.

Auditors are usually more interested in beta (Type II) risks, as they are concerned about the failure to detect material misstatements.

NON-SAMPLING RISK

Auditors may draw incorrect conclusions from using sampling techniques where the wrong inferences are not because of the selected samples themselves, but for other reasons not directly connected with the sample contents.
Non-sampling risk is where the auditor may have selected an appropriate sample but arrived at a wrong conclusion. Examples include employing inappropriate audit procedures, failure to recognize errors present, or misinterpreting the results or evidence.

**NON-STATISTICAL SAMPLING METHODS IN IDEA**

The non-statistical sampling features built into IDEA are shown in Fig 4.7.1. IDEA has easy-to-use random, systematic, and stratified random sampling built in.

![Fig: 4.7.1 Non-statistical Sampling Feature in IDEA](image)

**STATISTICAL SAMPLING METHODS**

There are three basic types of sampling methods that auditors may use. The choice of methods depends on the main purpose of the sample and substantive test.

**MONETARY UNIT SAMPLING (MUS)**

Monetary Unit Sampling method is used to estimate the total monetary amount of potential misstatement in a population. While other methods are based on occurrences or number of records, this method is based on rupee values where the higher monetary value transactions have a higher likelihood of being chosen in a sample. MUS is similar to systematic sampling, but where systematic sampling may sample every thousandth record, MUS will sample every thousandth rupees. It is typically used to determine the accuracy of financial accounts, where size is the most important factor, and where errors are expected to be few and far between. MUS provides a substantive assessment of error or misstatement in rupee figures and is specifically designed to predict overall error.

MUS should be used when:

- The process audited is well established and known to be reliable.
- The likelihood of errors (misstatements) is low.
- Obtaining a small sample size is important.
- You want to target larger rupee transactions, and expect to see some spikes in the data.
Step by step procedure to perform MUS sampling:

1. Planning:
   - Determine the objectives of the exercise.
   - Define the population.
   - Define what a misstatement means.
   - Determine sample size, using the following:
     - **Confidence level** - A percentage value comfort level that the sample will be representative and that you have the capabilities to interpret the results correctly.
     - **Tolerable error** - The point of no return past which you would no longer have faith in the process audited, nor the validity of the sample.
     - **Expected error** - The amount of errors or misstatements that are reasonably expected in a population.

2. Performing MUS Sampling Procedures
   - Select the samples.
   - Perform the audit procedures.
   - Record and analyze any errors observed.

3. Evaluation
   - Create a projected misstatement by summarizing errors and extrapolating these across population.
   - Compare ranges of the projected misstatement against the tolerable error limit.
   - Draw final conclusions.

As seen in Fig 4.7.2, IDEA will simplify all these steps for you.

![Fig: 4.7.2 Monetary Unit Sampling Feature](image-url)
In the example of testing sales, seen in Fig 4.7.3, we select absolute values as there are a few credit or refund values that are negative amounts in the Total field. We select a confidence level of 95 percent. Confidence levels below 90 percent are not generally recommended for MUS sampling. It should be recognized that the higher the confidence level, the larger the sample size needed.

The tolerable error amount or percentage must be entered. This is the absolute error limit that can be tolerated. The higher the tolerable error, the more errors you can accept and the lower the sample size needs to be. In this example, with sales in excess of Rs.1.7 million, 1 percent or Rs. 17,000 was decided as the maximum tolerable error. A 1 percent loss would be material enough to take actions such as to initiate an investigation or redesign the sales system. Since MUS is only to be used for processes where there is a high degree of confidence based on actual experience, significant errors may indicate fraud, embezzlement, or untrained or incompetent staff.

The expected error is the anticipated misstatement that is a realistic estimate of likely errors expected to found in the process. It is historical experience that dictates the estimate amount. Both the expected error and tolerable error can be entered either as an amount or percentage. In this example, management estimates that approximately 10 percent or Rs. 1,700 is the expected error.

When the Estimate button is selected, IDEA’s MUS calculation determines that in order to be 95 percent confident, it is necessary to set the sample size at 363 records and that there should be no more than 36.25 percent tainting or total percentage of errors found.

Once the Accept button is selected, the Monetary Unit Sampling—Extract screen appears as displayed in Fig 4.7.4

![Fig: 4.7.3 Monetary Unit Sampling planning](image-url)
Three hundred sixty-three records are then extracted and an AUDIT_AMT field is created as in Fig 4.7.5. The value of this field is equal to the field being audited but can be changed to reflect the proper amount determined during the execution of the audit procedures.

Suppose three discrepancies were found, including the one in record number 250 where the actual amount was Rs. 60.00 instead of the Rs. 55.32 recorded in the sales system as shown in Fig 4.7.6. To evaluate the MUS sample, select the Single Sample option.
The screen in Fig 4.7.7 appears

Once OK is selected, the summary shown in Fig 4.7.8 appears displaying:

- Zero overstatements were discovered.
- Three understatements were discovered.
The net most likely error is Rs. 1,512.47.

Both the gross and net upper error limits are less than materiality (tolerable error) of Rs. 17,000.

The conclusion is that with 95 percent certainty the projected total errors based on errors found in the sample is within the accepted range and will not exceed the tolerable error of Rs. 17,000. That is, you are 95 percent assured that the sample is representative of the population and that no actions to investigate or redesign the sales system are needed.

**ATTRIBUTE SAMPLING**

Attribute sampling is a statistical sampling technique often used to test internal controls; it evaluates the individual attributes of a record to be either true or false. Examples include:

- Having two required signatures for check authorizations over certain amounts.
- Whether account receivables are overdue.
- If travel expense claims are valid or not.

Attribute sampling should be used when:

- There is a need for a statistical sampling solution and judgmental sampling will not suffice.
- The objective of the review is to test compliance to internal controls.
- The compliance testing should evaluate to a true or false result.
- A random selection process will meet the objectives of your review.
Step by step procedure to perform attribute sampling:

1. Planning
   - Determine the objectives of the exercise.
   - Define the population.
   - Define what a misstatement means.
   - Determine sample size, using the following:
     - **Confidence level:** A percentage-value comfort level that the sample will be representative and that you have the capabilities to interpret the results correctly.
     - **Tolerable error:** The point of no return past which you would no longer have faith in the process audited, nor the validity of the sample.
     - **Expected error:** The amount of errors or misstatements that are reasonably expected in a population.

2. Performing Attribute Sampling Procedures
   - Select the sample.
   - Perform the audit procedures.
   - Record and analyze any errors observed.

3. Evaluation
   - Create a projected misstatement by summarizing errors and extrapolating these across the population.
   - Compare ranges of the projected misstatements against the tolerable error limit.
   - Draw final conclusions.

Similar to MUS sampling, attribute sampling requires a user to set certain boundaries and tolerances for the calculations to be performed.

- **Tolerable deviation rate as a percentage:** Also known as tolerable error rate, this is the absolute maximum percentage of transactions in error (i.e., not in compliance) that is acceptable as a cost of doing business. If you have more errors than the tolerable error rate, this internal control is not working and must be redesigned. The higher the tolerable error, the more errors you can tolerate, and the lower the sample size needs to be.

- **Expected deviation rate as a percentage:** Also known as expected error rate, this is the percentage of errors (i.e., noncompliance) you would reasonably expect to see, based on experience. As a rule, the lower the expected error, the lower the sample size.

- **Confidence level as a percentage:** This is the likelihood that the sample records chosen are indeed representative of the population at large, and that you will correctly interpret the results. The more confident you need to be; the more samples you require.

In the example in Fig 4.7.9, management decided that the maximum percentage tolerable deviation rate is 10 percent. Anything above 10 percent would suggest that the control is not working and may need to be redesigned. Based on previous history, management expects a deviation rate of 3 percent.
Management is happy to accept a confidence level of 90 percent that the sample is representative of the population or data set. The population size or number of records is 89,979.

The population size, percentage tolerable deviation rate, percentage expected deviation rate, and the confidence level must be entered into the Planning tab of the Attribute Sampling box.

Once the compute button is clicked, based on the information entered, IDEA informs you that you would need a sample size of 52 and that there must not be more than two deviations or errors in the sample to achieve a 90 percent confidence level that deviation level of the population is not more than 10 percent. Refer to Fig 4.7.10
The random record sampling feature in Fig 4.7.11 can be used to randomly select the 52 required records. A detailed audit of the 52 selected records determines whether the control was met. The number of deviations is noted.

![Random Record Sampling](image)

**Fig: 4.7.11 Obtaining sample for attribute sampling**

In the Sample Evaluation tab of the Attribute Sampling box, entries for the population size, sample size, and percentage of desired confidence level are made again. The number of deviations (five) discovered during the audit of the 52 samples must also be included, as noted in Fig 4.7.12.

Since five deviations are more than the critical number of deviations (two) in the sample calculated in the planning stage, the conclusion is that there is 90 percent certainty that the number of deviations could be significantly higher in the overall population than the 10 percent tolerable percentage. It is 90 percent certain that there could be as many as 17.11 percent of errors if the entire population was audited. As such, this sample cannot be considered as representative and it is likely the control must be redesigned.

![Sample Evaluation](image)

**Fig: 4.7.12 Conclusion from the audit results of attribute sampling**
4.8 USING ADVANCED STATISTICAL METHODS

4.8.1 – CORRELATION

A correlation is a relationship between two things or mathematical variables that tend to vary or move together. The data is represented by the letters $x$ and $y$ where $x$ is the independent variable and $y$ is the dependent variable. The independent variable $x$ is usually described first. There is usually some logical connection between the two variables.

Many studies have been done on income levels of high school graduates, college, and university graduates and those with post-graduate degrees. The question in those studies is whether there is a connection or correlation between educational levels and income levels. Educational level would be $x$, the independent variable that would influence the income level variable of $y$, the dependent variable.

How much does the independent variable have influence over the dependent variable can be determined by calculating the correlation coefficient and is designated as $r$. A perfect linear correlation would have $r$ equaling to 1 and a perfect negative correlation would have $r$ equaling –1. The closer $r$ is to 1 or –1, the stronger the correlation. Where there is no correlation, $r$ would equal to 0.

There are three basic formulas for calculating $r$. The correlation coefficient can be calculated for the population, the sample, or the product moment. The most common one is the product-moment correlation coefficient as shown below.

$$r = \frac{\Sigma(xy)}{\sqrt{\left(\Sigma x^2\right) \left(\Sigma y^2\right)}}$$

$\Sigma$ denotes the summation symbol, so the formula for the top part or numerator is to multiply the $x$ variables by the $y$ variables and then take the sum of those numbers. For the denominator, you square the $x$ variables and take the sum of them. Do the same for the $y$ variables and then multiple the resulting two sums. Apply the square root to the results and then divide the final results of the numerator by the final results of the denominator to obtain the correlation coefficient.

Since we already know how to calculate the Z-scores (or have IDEA calculate it for us), a simpler method of obtaining $r$ would be to multiply the Z-scores of the $x$ variables by the Z-scores of the $y$ variables and then total up all the results. Take the total of the results and divide by the number of records less one.

The formula would be

$$r = \frac{\Sigma(Z_x * Z_y)}{(n - 1)}$$

where $Z_x$ is the Z-score for the $x$ variable and $Z_y$ is the Z-score for the $y$ variable. The letter $n$ in the formula represents the number of records.

In general, the correlation coefficient, whether negative or positive, can be interpreted as:

<table>
<thead>
<tr>
<th>Interval</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>.0 to .2</td>
<td>No correlation</td>
</tr>
<tr>
<td>.2 to .4</td>
<td>Weak correlation</td>
</tr>
<tr>
<td>.4 to .6</td>
<td>Moderate correlation</td>
</tr>
<tr>
<td>.6 to .8</td>
<td>Strong correlation</td>
</tr>
<tr>
<td>.8 to 1.0</td>
<td>Very strong correlation</td>
</tr>
</tbody>
</table>

The calculations of the correlation coefficients were shown for a better understanding of correlation. The auditor does not need to perform the calculations using the formulas, as IDEA has a built-in correlation feature that provides the correlation coefficient and you merely have to interpret the
Using a summarized monthly sales file from a POS system, we select correlation from the statistics area of IDEA. For demonstration purposes, the fields we correlate will be the HD_NETAMOUNT_SUM field, which is the sales amount before taxes, and the PAYAMOUNT_SUM_SUM_SUM field, which are amounts paid that include taxes. We will also include payments of cash only and payments of debit cards only for the correlation calculation as shown in Fig 4.8.1.1. In addition to the results being displayed, which can be exported to various file formats, you may optionally create an IDEA database of these results.

As expected, there is a perfect positive correlation of 1.000 between sales before taxes and payments. As sales go up or down, the sales tax moves accordingly, so the total payment by customers’ correlates to sales net of taxes.

There is a strong correlation between both the cash tender and the debit cards tender to the payment amounts of 0.710 and 0.792, respectively. There is no correlation between cash and debit cards payments as the correlation coefficient is 0.189. See Fig 4.8.1.2.

![Fig: 4.8.1.1 Applying the Advanced Statistical Analysis Correlation Feature of IDEA](image1)

![Fig: 4.8.1.2 Correlation coefficient results](image2)
While IDEA does not calculate the coefficient of determination, it can be done simply by squaring the correlation coefficient that is already calculated for you. The coefficient of determination, or \( r^2 \), tells us how much of the variation in one variable can be attributed to the variation of the other variable. This calculation when multiplied by 100 will be expressed in a percentage.

In our example of the correlation between cash tender and the payment amount of 0.710, 50.41 percent \((0.710 \times 0.710 \times 100)\) of the variation can be attributed to the other variable.

One has to be mindful that even if the variables are calculated to have a strong correlation, there may not be a cause-and-effect relationship.

Are there direct cause-and-effect relations? That is, does \( x \) cause \( y \) or, in our previous example, does the level of education cause the income level? Maybe it is a reverse cause and effect where \( y \) causes \( x \). That is, income levels determine your education level.

Possibly there is a third variable or a combination of several other variables that caused the relationship, such as networking relationships while in school that resulted in higher paying jobs. Maybe the whole relationship between the two variables was just a coincidence?

### 4.8.2 TREND ANALYSIS

Trend analysis is based on regression analysis. Regression analysis produces a line of best fit and predictions can be made based on the line. It is also known as the least square line, because the line passes through the distribution where the distance squared from the line is minimized.

Similar to that of correlation, the \( x \) variable can estimate or predict the \( y \) variable. That is, if \( x \) changes, then how much \( y \) changes can be estimated. For two numeric variables, you can predict \( y \) from the \( x \) variable if the correlation coefficient is strong and there is a linear pattern for the variables. Normally you would want the \( r \) correlation coefficient to be better than plus or minus 0.60.

Unless there is perfect correlation, the prediction for the value of \( y \), given \( x \), is merely a prediction. It is a guess but an educated guess with some sound scientific basis. The prediction will be subject to some amount of error. The standard error of the estimate measures how much the predicted values deviate from the actual \( y \) values. IDEA uses the mean absolute percent error (MAPE) to calculate the accuracy of predictions. MAPE is the average of the percentage errors, is expressed in percentage terms, and works best with positive amounts. The MAPE number is the predicted line on the average that is away from the actual line in percentages. Low MAPE values mean that the past data has a good fit to the regression line and you can have more confidence in the data and prediction.

To use trend analysis in IDEA, the database needs at least one numeric field, and the field where trend analysis is to be applied cannot contain any bad data. The audit unit field cannot be the same field as the trend analysis field. In addition, the database should not contain seasonal data. If it contains seasonal data, then time series analysis should be selected over trend analysis. Time series works similarly to trend analysis.

In our data file, we will perform trend analysis on the 12-month data from the debit card payments field of DEBIT_2010_2011 and generate 3 months of forecasts. It is not necessary to provide a reference field or audit units. Refer to Fig 4.8.2.1.

For debit card payments, it is trending slightly downward, and it is predicted that at the end of the three months, debit payments would drop to approximately Rs. 18,000 for that month as seen in Fig 4.8.2.2.
The MAPE is 5.36 percent, which provides high confidence as the reliability of the predictions. In contrast to the debit card payments, cash payments are trending upward when we select CASH_2010_2011 as the field to trend.

There is less reliability in the prediction as the MAPE percentage is 23.17. It is predicted that at the end of the three months, cash payments would increase to approximately Rs. 14,500, as displayed in Fig 4.8.2.3.

We will perform a trend analysis showing a reference field and audit units. There are five branches and we will include all of them as audit units. The reference field of GLOBAL_AVERAGE_SALES is the average sales for the five stores broken down by months. We have eight years of data so each store would have 96 records (8 years × 12 months). We will generate forecasts for three months as shown in Fig 4.8.2.4.

![Fig. 4.8.2.1 Trend analysis in IDEA](image1)

![Fig. 4.8.2.2 Trend analysis results of debit card payments with forecast of three months](image2)
Fig. 4.8.2.3 Trend analysis results of cash payments with forecast of three months

Fig. 4.8.2.4 Applying trend analysis of sales referencing global average sales for each branch

Branch A outperformed the average of the five branches. The actual data is above the reference data line and shows good promise of trending upward in Fig. 4.8.2.5. The prediction is sound as the MAPE is 5.26 percent.
By selecting all five branches as the audit unit, we can display other branches by choosing from the Audit unit pull-down menu. We will look at one more by selecting

*Fig: 4.8.2.5. Result of branch A’s trend analysis with three month’s forecast*

Branch B see Fig 4.8.2.6 fits right in around the global average sales and is also trending up.

*Fig: 4.8.2.6. Result of branch B’s trend analysis with three month’s forecast*
4.8.3 TIME SERIES

For data with seasonal values where there are higher values in certain months (or any other time units) and lower values in other months, the time series analysis option is more appropriate. In our example in Fig 4.8.3.1, we will use gas-heating costs as the field for the time series and all five branches as the audit units. We ensure that we input the correct season length of 12 as our records are broken down by months. We will generate 12 months of forecasts.

**Fig: 4.8.3.1. Applying Time Series Trend Analysis of Heating Costs for Each Branch with 12 Month Forecast**

From the Fig 4.8.3.2, you can see that heating costs are higher in the winter months and lower in the summer months for branch A. The mean absolute percentage error of 11.40% is fairly reliable in terms of the forecast. You can also view the other branches by using the pull-down menus from
Trend analysis using IDEA is simple and the auditor need not be concerned with the complex formulas to calculate the regression line and the mean average percentage error.

### 4.9 IDEA SCRIPT

IDEA Script is a programming tool in IDEA that allows for combining numerous steps into a single procedure. The programming language is similar to that of Microsoft’s Visual Basic for Applications. IDEA Script files are also known as macro files. It is a file that performs a series of actions when executed. IDEA Script can record actions for editing and modification to make the macro good for general use. Any user can apply the script to any suitable data files. IDEA Script is a powerful tool that speeds up repetitive procedures.

The tools built into IDEA eliminated the need to enter your computer codes from scratch. The record macro task can be used to record a sequence of steps that will be the framework for your application or macro. The Dialog Editor will help you design customized dialog boxes that you can use to control the flow of your application and to prompt the users for required input. However, IDEA Script cannot always anticipate all of the details you may want to incorporate in your macro so you will still have to write or edit some of the code.

**IDEA Script can be used for:**

- **Automating repetitive tasks:** Regular tasks, such as monthly data that needs to be analyzed, can use IDEA Script to run and repeat the required procedures automatically. IDEA Script can be efficiently used when certain procedures are required to be applied in multiple locations such as departments, divisions, and branches. Another frequent repetitive task is to import files. This can be automated to select the input files, specify the output names, perform the imports, and perform required data cleanup or scrubbing.

- **Creating an automated analysis system:** A set of tests or procedures can be integrated into an IDEA Script where the user may select particular tests to apply.

- **Controlling other software packages:** Using Microsoft’s object linking and embedding (OLE) technology, other OLE-enabled software can be controlled from within IDEA. One example is that of sending IDEA data into an Excel spreadsheet. IDEA Script performs analysis tasks such as summarization, sends the summarized data into an Excel spreadsheet, and then the IDEA Script instructs Excel to chart the summarized data into bar graphs.

- **Creating custom tests:** IDEA Script can be used to create specific tests based on the user’s or the organization’s needs. Typically, these would be performing calculations that involve comparisons and equations.

In using IDEA Script, work will be performed faster and the results will contain fewer errors. The analyses are done more consistently and can be undertaken by other staff members. Standards are adhered to by all users.

**Consideration for Automation**

The following steps should be considered when planning to create an IDEA Script.

- Identify those tasks that are frequently repeated.
- Select those procedures that are well defined and well understood.
Computer Assisted Audit Techniques

- Plan the macros carefully to ensure the results are those expected.
- Include user-input dialog boxes so that others can use the IDEA Script.
- Make the script as flexible as possible so it is useful to other users.
- Test IDEA Script on other computers with different versions of IDEA for compatibility and to ensure that the script works as expected.

**Visual Script versus IDEA Script**

Visual Script can create and edit macros in IDEA. Visual Script is a visual representation of batch processing that allows for simple dragging (or double clicks) of desired action choices from the left side of the window to the right side for project steps of the Visual Script screen, as shown in Fig 4.9.1.

Visual Script, like IDEA Script, allows for automation of repetitive tasks. However, the automation can be performed without the complexity of knowing any programming and writing code. There are limitations, such as not being able to create message boxes, nor will Visual Script run in the IDEA Server environment. Visual Scripts have the .vscript filename extension and are saved to the default Macros.ILB folder under the active project folder.

Visual Scripts can be later converted to IDEA Scripts, through an option in the Visual Script Editor. Because of the easy conversion to IDEA Script, you can start the steps for your macro in Visual Script, convert it to IDEA Script, and then code additional functionalities within the IDEA Script editor. Visual Script has the ability to include IDEA Scripts into it. This makes it simple to group together a number of IDEA Scripts and have Visual Script run them all. Visual Script is an excellent personal tool.

IDEA Script can use dialog boxes, control other software applications, and interact with the operating system. An example of the IDEA Script editor window with a sample script is shown in Fig 4.9.2.

*Fig 4.9.1: Visual script Editor Window*
Creating IDEA Scripts:
The script in Fig 4.9.3 is an example of interacting with the Windows operating system function of renaming a file in a folder. This script does not include dialog windows to allow a user to select the folder and file to be renamed. It is hard-coded into the script that the “text.txt” file must be in the C:\data folder. It can only be renamed to “text_renamed.txt.” The “CreateObject” function allows IDEA Script in IDEA to access Windows file-system functions and other application functions, such as in Microsoft Excel.

An example of a dialog box is shown in Fig 4.9.4. The IDEA Script extracts transactions with even thousand amounts from the active file in IDEA. The script takes all numeric fields within the file and allows the user to select the field to apply the test on. In this case, the AMOUNT field is the appropriate one to select from the pull-down selection box.
In contrast, when using Visual Script, as shown in Fig 4.9.5, the procedure is:

- Under File, select Open Database. All databases in the current project are offered for selection. Choose the file to open.
- Under Extract, select Direct. The direct extraction option opens up. Enter in the File Name area MOD Thousand. Enter the equation of \((\text{AMOUNT} \mod 1000) = 0\) in the Criteria area or Equation editor.
- By double clicking on the “Payments” filename under the Project Steps, all databases in the current project are offered again.
- By double clicking on Direct Extraction under the Project Steps, the Direct Extraction option opens up where you can modify the output filename and change the equation.
CREATING MACRO:

Visual Scripts are simpler to create but the user needs some knowledge to apply, such as being able to understand the equation and modify it. IDEA Scripts are more complex for the programmer but easier for the user. They are also more accurate if error checking is built in. Creating simple IDEA Scripts can be as easy as recording the steps that you are taking to accomplish a task. Fig 4.9.6 shows how you can start and stop recording a macro.

![Fig: 4.9.6 Recording Macro](image)

Start recording the macro. Perform the procedures and then stop recording the macro. A window will then pop up as shown in Fig 4.9.7.

![Fig: 4.9.7 Macro option after ending recording of macro](image)

You can select either Visual Script or IDEA Script. Selecting either one will open the appropriate editor window and automatically paste the actions or codes in. When procedures are performed in IDEA, they are logged in the history found under the Properties window. You can copy all the procedures into IDEA Script or selected tasks. In our example in Fig 4.9.8, we select the “Copy the IDEA Script for the selected task(s)” option since we right-clicked on the record extraction area to
bring up the selection. Once the selection is made, the IDEA Script editor opens and the script is automatically created for you.

![Fig: 4.9.8 Creating Macro from file History](image)

You can make changes to both the source and output filenames along with the equation from within the IDEA Script editor to suit you. Alternatively, you can add dialog boxes and coding to allow for user input. The dialog box shown in Fig 4.9.4 was created in the IDEA Script editor’s dialogs area, displayed in Fig 4.9.9.

![Fig: 4.9.9 Dialogue Editor](image)

4.10 IMPORTING FILES WITH REPORT READER

The information you want to audit is usually available as a printout. In most cases, the printout can be routed or “spooled” to disc, and then transferred to the auditor’s personal computer. Occasionally, it may be necessary to use a print file interceptor, which will redirect the print file being sent to the printer to a file on disc.

The differences between an ASCII file and a Print Report file are mostly in the header and footer, and in the extra lines and totals that you have to include or remove in order to import the file into IDEA.
Advanced Analysis Using IDEA

Report Reader extracts data from plain text reports saved to file or some Adobe Acrobat PDF files and translates it into the specialized file formats of IDEA.

**Step-by-step procedure to import file with Report Reader:**

1. Access the print report or .PDF file.
2. Create the Base Layer.

Report Reader uses a set of layers to define the data to be imported into IDEA. The first layer defined is the Base Layer. The Base Layer tells Report Reader to write a record for each occurrence of the information defined by the layer. The Base Layer must be defined at the detail line, or transaction level. A detail line is the line or lines that contain the information to be identified as a single transaction or record in the resulting IDEA database. The detail line can be a single or multi-line entry in the report.

IDEA will be prompt to:

- **Create a standard layer:** Select this option when the report is in columnar format.
- **Create a floating layer:** Select this option if the data is not displayed in columns.
- **Exclude it from the output:** This option is only available after the Base Layer is created. This option allows you to exclude lines that contain data not required for import.

3. Create an Append Layer.

An Append Layer captures information that is associated with the Base Layer and that may be shared by more than one record. Data defined in the Append Layer will be appended to the records defined in the Base Layer. For example, in a report with transactions categorized by account number, it is possible to have one account number associated with a group of transactions. The individual transaction data is captured in the Base Layer and the Append Layer captures the account number that applies to the group of transactions.

If the information in the Append Layer is located before the transactions in the Base Layer, it is called a Pre-Append Layer. If the information in the Append Layer is located after the transaction in the Base Layer, it is called a Post-Append Layer. For example, in the following report, if the first entry (12-5-2017, Raj Malhotra) was selected as a sample for the Base Layer, the sample for the Append Layer should be Account 1023 and it will be a Pre-Append Layer because it is located before the sample line for the Base Layer.

<table>
<thead>
<tr>
<th>Account 1023</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12-5-2017</td>
<td>Raj</td>
<td>539.00</td>
</tr>
<tr>
<td>12-5-2017</td>
<td>Malho</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Account 1024</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3-5-2017</td>
<td>Pravee</td>
<td>326.75</td>
</tr>
<tr>
<td>4-5-2017</td>
<td>Rahul</td>
<td>434.50</td>
</tr>
</tbody>
</table>

4. Preview the resulting database.
5. Scan the file for errors.
6. Import the file into IDEA.
Exercise: Import Print file into IDEA

Step 1: On the Home tab, in the Import group, click Desktop to begin the import process.

Step 2: In order to import reports into IDEA, select Print Report and Adobe PDF, then select the samplefile.txt file.

Step 3: Click Next. The Report Reader opens and displays the contents of the samplefile.txt file. The report will be displayed as in the following Fig 4.10.1

![Report reader Screen](image)

Fig. 4.10.1 Report reader Screen

Step 4: Scroll through the report and note the formatting within the report, including:

- General information (company name, report name)
- Report headings (range of accounts and period)
- Account information (number description)
- Detail information

Step 5: Creating a Base Layer

- Any detail line of the report can be used as a Base Layer 1.
- Drag your cursor across any detail line in the report to define it as your sample line.
- A message box appears, prompting you to create a standard layer or a floating layer
- Accept the default of Create a standard layer and then click Yes. Refer Fig 4.10.2
- The selected line is now copied to a section at the top of the page that is referred to as the Field Editor. Refer Fig 4.10.3.
Step 6: Define Trap:

- You must at this point define what distinguishes this detail line from the other lines in the report. It can be specific text, a combination of numbers, or a date. In Report Reader, these formats are
called Traps. For this report, you can see that the third field or column in the report is a Date field and it appears to be formatted the same way throughout the entire report (NNNN-NN-NN). Using this information, a trap can be defined in the Anchor Editor section (the line above the Field Editor).

- Place your cursor in the Anchor Editor directly above the first character of the date, just above the number 2. Since you want to import all the transactions (any entry with a date), click four times on the Numeric Trap button followed by a hyphen (-). Refer Fig 4.10.4.

Fig. 4.10.4. Anchor Editor

- Note that all the lines in the report that meet the criteria or follow the pattern are highlighted in blue. If you were to leave out the hyphen (-) you would end up trapping the header line above the field name row (Accrual Basis January through December 2012) because the 2012 lines up with your NNNN trap.

- The other options available to trap information are:
  - Text Trap
  - Space Trap
  - Non-Blank Trap

- You can browse through the report to ensure that all the detail lines and only the detail lines are highlighted. When satisfied that the required lines are selected, you must define the information you want to import into IDEA from the detail lines. This information will be brought into IDEA as separate fields.

Step 7: Define Fields:

- In the Field Editor (the row with the red arrow), highlight the Trans # information.

- The colour of the selected field will be orange and all the corresponding records of the detail lines in the report will be highlighted. Scroll through the report to verify that the field width
is sufficient to capture records with larger amounts of data and adjust the field width in the Field Editor if required. Refer Fig 4.10.5.

**Fig. 4.10.5. Define Fields**

- In the Field Details window, enter the following details:
  - Name: TRANSNO
  - Type: Numeric
  - Decimals: 0
- In the Field Editor, highlight the Type information, leaving enough room for the larger data.
- In the Field Details window, enter the following details:
  - Name: TYPE
  - Type: Character
- In the Field Editor, highlight the Date information.
- In the Field Details window, enter the following details:
  - Name: DATE
  - Type: Date
  - Mask: YYYY-MM-DD
- In the Field Editor, highlight the Num information, leaving enough room for the larger data.
- In the Field Details window, enter the following details:
  - Name: NUM
  - Type: Character (Note mix of numbers and character data, hence the Character type)
In the Field Editor, highlight the Name information, leaving enough room for the larger data.

- In the Field Details window, enter the following details:
  - Name: NAME
  - Type: Character

- In the Field Editor, highlight the Memo information, leaving enough room for the larger data.

- In the Field Details window, enter the following details:
  - Name: MEMO
  - Type: Character

- In the Field Editor, highlight the Debit information, leaving enough room for a bigger number.

- In the Field Details window, enter the following details:
  - Name: DEBIT
  - Type: Numeric
  - Decimals: 2

- In the Field Editor, highlight the Credit information, leaving enough room for a bigger number.

- In the Field Details window, enter the following details:
  - Name: CREDIT
  - Type: Numeric
  - Decimals: 2

When you are done the fields should match the following table:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSNO</td>
<td>Numeric</td>
<td>Decimal :0</td>
</tr>
<tr>
<td>TYPE</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>Date</td>
<td>Mask: YYYY-MM-DD</td>
</tr>
<tr>
<td>NUM</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>MEMO</td>
<td>Character</td>
<td></td>
</tr>
<tr>
<td>DEBIT</td>
<td>Numeric</td>
<td>Decimal :2</td>
</tr>
<tr>
<td>CREDIT</td>
<td>Numeric</td>
<td>Decimal :2</td>
</tr>
</tbody>
</table>

- Click the Save Layer button

Step 7: Creating an append layer

You now have to create an Append Layer to ensure you have an account name attributed to the correct detailed transaction information. In this report, you will be creating a Pre-Append Layer because the account name is located before the transactions of the Base Layer.

- Highlight the line under the report column titles that contains the account name, and then create a new standard layer. In the Anchor Editor, you need to make use of the Space Trap and
Text Trap buttons. The account names are indented a specified amount of spaces in this report so if you trap all the spaces up to when the text begins to identify the account name, you will eliminate any detail information you already pickup and header information in the report.

- Place your cursor in the Anchor Editor at the far left and click the Space Trap button to add spaces up to the start of the account names.
- Click the Text Trap button.

![Fig. 4.10.6]

- In the Field Editor, highlight the account name information.
- In the Field Details window, enter the following details:
  - Name: ACCTNAME
  - Type: Character
- Blank Cells: Use value from previous record.
- The field you just defined is located in the header which means that you would like to repeat the information it contains for each record (detail line) below it. To indicate to Report Reader that you want to repeat that information, the Use value from previous record option is selected in the Blank Cells box in the Attributes section of the Field Details window. Refer Fig 4.10.7
- Click save layer button.
Step 8: Previewing the resulting database

- On the Report Reader toolbar, click the Preview Database button. The entire report will be displayed in a Preview Data Window.

- Click close to exit preview data window

- If you need to change information for a field, click on the field and make any necessary changes through the Field Details window. Changes can also be made by re-opening the layer for editing.
To access the Layer Manager, select Layers > Layer Manager. From the Layer Manager, you can edit, copy, or delete a layer. For more information on the Layer Manager refer to the Report Reader Help system.

Step 9: Importing into IDEA

- You are now ready to import the file into IDEA. Click the import button.
- When prompted to proceed with the import, click Yes.
- Report Reader prompts you to save the template. Click OK.
- The Save As dialog box appears. You are prompted to save your template file .jpm in your current project folder. The template is similar to a record definition. It contains all the information you defined in Report Reader. You can reuse the template to import a similar report or you can open the template and make changes to it.
- In the File name field, enter sample file and click Save.
- The last import screen allows you to enter the name of the generated IDEA database.
- In the Database name field, enter name and click Finish.
- Final output in IDEA format. Refer Fig 4.10.9.

![Fig. 4.10.9 Converted database](image)

Reconciliation of data Imported

Once the data has been defined and imported into IDEA, and before commencing audit testing, it is imperative that the data be reconciled back to the host system. There are several potential errors that may be made during the import process, such as:

- Requesting the wrong data
- Being supplied with the wrong data and/or for the wrong period
- Errors on the extraction of the data from the host system
• Problems with the media while transferring the data
• Errors when importing the data

The data can be reconciled back to the host system by reviewing Numeric field totals, the record count, or a sample of records compared to a print from the host system. It is important that reconciliation reports and control totals are requested along with the data and that they are filed with the documentation from the download/import process.

There are different ways to reconcile the data. Certain IDEA tasks are particularly useful for reconciling data. First, it might be useful to browse the IDEA database. Use the Field Statistics and the Control Total properties, accessible from the Properties window, to reconcile totals and to ensure that the dates match the data you requested and do not include transactions from another period.

4.11 FILED MANIPULATION

The Field Manipulation task is used to modify the layout of the fields in a database. There are two ways to access Field Manipulation:

• On the Data tab, in the Fields group, click the arrow in the bottom right corner
• Double-click any value in the Database window. The Field Manipulation dialog box appears.

MODIFYING FIELDS

To rename a field double-click the field’s name in the Field Name column. Any changes made to the field name will be recorded in the History. The information in the Type column is initially entered by IDEA during import. Four general types of fields: Character, Numeric, Date, Time and three special types of fields: Virtual, Editable and Multistate are available. You cannot change the length of a field or the number of decimals unless you have modified the information in the Type column. You can double-click in the Description column to change field descriptions.
EXERCISE 1
Make the following changes to the Sales Transactions-Database.

- Ensure that Sales Transactions-Database is the active database and that the Data property is selected in the Properties window.
- Double-click any record in the Database window.
- The Field Manipulation dialog box appears.
- Rename the CUSTOMER_NO field to CLIENT_NO.
- In the Description column for the SALES_TAX field, enter 8%.
- In the Description column for the INV_NO field, enter Invoice number.
- Click OK.
- Click Yes to make the changes to the fields in the database.

APPENDING FIELDS
IDEA maintains data integrity upon import. The data cannot be changed. Instead, IDEA allows the creation of Virtual and Editable fields. For example, you can add a Virtual or Editable field that contains a calculation performed on an original data field.

Virtual fields compute results “on the fly”. The results are not stored in the database. However, any database created from a database with a Virtual field contains that field as a calculated value. Virtual fields can be Character, Numeric, Date, or Time fields.

Editable fields and regular fields are stored in the database and require some processing time in order to be physically added to the IDEA database.

EXERCISE 2: APPENDING VIRTUAL FIELDS
Adding new fields might be useful when proving or verifying calculations, when creating a new field with calculated values, or when transforming existing data.

In this exercise, you will create a Virtual field called AMOUNT_CHECK in the Sales Transactions-Database. By creating a calculation, you will determine if the amount in the AMOUNT field is correct.

- Ensure that Sales Transactions-Database is the active database and that the Data property is selected in the Properties window.
- Double-click any record in the Database window.
- The Field Manipulation dialog box appears.
- Click Append.
- A new line is added to the Field Manipulation dialog box.
- Enter the following details for the new field:
  - Field Name: AMOUNT_CHECK
  - Type: Virtual Numeric
  - Length: n/a
Decimals: 2
Parameter: QTY * UNIT_PRICE
Description: Calculated amount
Click OK.
If prompted, click Yes to make the changes to the database.

**Fig 4.11.1: Appending virtual field**

Note the colour of the data in the AMOUNT_CHECK field, which indicates that this is a virtual field not part of original data but field appended by auditor

**EXERCISE 3: APPENDING REGULAR FIELDS**

In IDEA, you can also add regular Numeric, Character, Date, and Time fields through Field Manipulation. These fields will appear as if they were added during import. They will retain the same colour as the other fields and will not have any parameters displayed. As with Virtual fields, any additions or changes made to these fields will be recorded in the History. The parameter of a regular field cannot be changed.

In this exercise, you will create a Numeric field that is identical to the AMOUNT_CHECK Numeric field.

- Ensure that Sales Transactions-Database is the active database and that the Data property is selected in the Properties window.
Double-click any record in the Database window.

The Field Manipulation dialog box appears.

Click Append.

Enter the following details for the new field:

- Field Name: NUM_AMOUNT_CHECK
- Type: Numeric
- Length: n/a (by default it will become 8)
- Decimals: 2

Parameter: AMOUNT_CHECK

Description: Calculated amount

Click OK.

Click Yes to make the changes to the database

Fig 4.11.3: Regular field NUM_AMOUNT_CHECK

Note the colour of the data in the NUM_AMOUNT_CHECK field, which indicates that this is a regular protected field.

- Re-open the Field Manipulation dialog box.

Note that the parameter is not displayed for the NUM_AMOUNT_CHECK field.

- Click OK to return to the database.

**EXERCISE 4: APPENDING EDITABLE FIELDS**

IDEA also allows for the creation of Editable fields. These fields will allow you to create a notation, or manually enter a value to add to a database. The creation and editing of these Editable fields is recorded in the History. Editable fields are also displayed as a different colour to distinguish them.
from Virtual and Regular fields.

- With Sales Transactions-Database as the active database, append an Editable Character field to the database in order to add notes.
- You will need to add an empty value (represented by double quotes) in the Parameter column so IDEA knows you want to enter characters into the field at a later time.
- Enter the following details for the new Editable field:
  - Field Name: NOTES
  - Type: Editable Character
  - Length: 50
  - Decimal: n/a
  - Parameter:
  - Description: Notes on line items

Note that the Editable field created will wrap the text within the field as you type in order to display everything in the current column width.

EXERCISE 5: APPENDING SPECIAL FIELDS

IDEA further allows for the creation of special fields. These fields are termed as Boolean and Multistate fields. They are used generally by auditor to provide remarks in sample file in a symbolic way. Refer Fig 4.11.4 to see types of fields options provided by IDEA.

**Fig. 4.11.4 – Types of fields**

Boolean fields offer two options to user whereas multistate field offers three options. Refer Fig 4.11.5 below.
4.12 JOIN DATABASES AND VISUAL CONNECTOR

IDEA has two main methods of joining databases together based on a key field:

- Join Databases
- Visual Connector

JOIN DATABASES:

Data required for auditing is often spread across several files and perhaps across different computer systems. In order to carry out most audit tests and analyses, the data must be contained within a single file. IDEA provides a Join Databases task, which can be used to:

Combine fields from two databases into a single database for testing. For example, to carry out an inventory valuation, data is required from both the inventory master and pricing information databases.

Test for data, which matches or does not match across systems. For example, match the monthly payroll transactions database with the master payroll database to ensure there are no “ghost” employees, or that all former employees have been removed from the payroll.

Databases can be joined or matched if they contain a common link (referred to as the “key”), such as an employee number field.

Join Databases can perform a host of different join option on two databases, including all records in primary database, matches only, records with no secondary match, records with no primary match, as well as all records in both databases.

Join Databases Precautions

- Only two databases can be joined at one time. If more than two databases must be joined, you must combine two databases, then join a third to the resultant database, and so on. You should also consider the Visual Connector task if you need to join more than two databases.
- The common fields (the “key”) do not need to have the same name, but they must be of identical field type.
The databases must be in the same location. For example, a database in the Samples project cannot be joined to a database in the test project.

When using the Join Databases task, care must be taken to select the primary and secondary databases in the correct order as the secondary database is joined to the primary database. Transaction file should be the Primary file and Master file should be the secondary file. There should be Many to one relationship.

**IDEA provides five join options:**

**Matches only**

In this option, IDEA processes each record in the primary database (in “key” order), cross matching with the secondary database on the key. If there is a match on the key, a record is written to the output database. If there is no match, the next primary database record is processed. Matches only will search for the keys in the secondary database for matches against each record.

**Records with no secondary match**

In this option, IDEA processes each record in the primary database in turn, cross matching with the secondary database on the key. It writes a record to the output database if the key exists in the primary database only.

**Records with no primary match**

IDEA processes each record in the secondary database in turn, cross matching the key with the primary database. IDEA writes a record to the output database if any key exists in the secondary database only.

**All records in primary file**

IDEA processes each record in the primary database (in “key” order) in turn, cross matching with the secondary database on the key.

Each record in the primary database is written to the resultant output database including the selected fields from the secondary database if the key exists.
All records in both files

In this option, IDEA processes the database in key order, writing a record to the output database for every record in each database where there is a key match.

If there is more than one record for a particular key in both the primary and secondary databases, each of the primary database records are matched against the first record for the key in the secondary database as in the options Matches only and All records in primary file. However, all additional records for the key in the secondary database are also written to the output database but all primary fields will be blank for Character fields, 0000/00/00 for Date fields and zero for Numeric fields.

JOIN Vs VISUAL CONNECTOR

Join Databases has two important limitations:

- It can only join two databases at a time.
- The Matches only option is similar to Visual Connector but has the following limitations:
  - When a many-to-many relationship exists, only the first matched record from the secondary database is included in the result.
  - The resulting joined database does not exceed the number of records in the primary database, regardless of the matching ability of the key fields. In other words, there is only one primary match for every secondary match. With the exception of matching types such as all records in secondary database and all records from both files that can and will, respectively, have more records than the primary database.

- On the other hand, Visual Connector is an easy-to-use, visual method of connecting two or more databases together by drawing lines between key fields on a canvas, and does not have any limitations on how many records are joined.
- Visual connector can join more than two files at a time
- Visual connector unlike Join option which has five different options has only two options – matches only and all records in primary file

EXERCISE: Join Customer details.imd with account details.imd

Account details file has fields – ACCNO, STATUS, BR_CODE, BALTODAY, BLYESTDAY
Customer details file has fields – ACCNO, CUST_NAME, CUST_ADD, LOCALITY, STATE

Join both the file using common field ACCNO

Step 1: Open primary file account details and ensure it is active file in IDEA
Step 2: Go to analysis tab and click on join button
Step 3: Primary database automatically selected is Account Details. Select Secondary database customer details.

Step 4: Select option ‘All records in both the files” since we want to join both the files.

Step 5: Provide file name as ‘combined customer details”. Refer Fig 4.12.2.

Step 6: Click on Match option.

Step 7: Provide primary and secondary key which is common in both the files ACCNO. Refer Fig 4.12.3.
Step 8: Click on OK

Step 9: See the output in combined customer details file. Refer Fig 4.12.4. you would observe that joined file contains all the fields from customer details and account details files

**EXERCISE: 2 – Join using visual connector option - Customer details.imd with account details.imd**

Account details file has fields – ACCNO, STATUS, BR_CODE, BALTODAY, BALYESTDAY
Customer details file has fields – ACCNO, CUST_NAME, CUST_ADD, LOCALITY, STATE

Join both the file using common field ACCNO

Step 1: Open primary file account details and ensure it is active file in IDEA

Step 2: Go to analysis tab and click on visual connector button. Refer Fig 4.12.5

Step 3: From the list of databases on the left panel, select file customer details
Step 4: Create a link between both the files by dragging the mouse starting from ACCNO field in account details file and dropping at ACCNO filed at customer details file. This will establish a connection between two files. Refer Fig 4.12.6

Step 5: Click on Ok and select appropriate option. In this case your objective is to join the file, select option “all records in primary file, all matches”. Refer Fig 4.12.7

Step 6: Provide file name – combined details visual connector and click on OK. Refer Fig 4.12.7

Step 7: Observe the output in combined file created using visual connector option. Refer Fig 4.12.8
4.13 COMPARE DATABASES

IDEA can compare the financials from one year to the next, and it’s called Compare Databases. This task requires two databases to be defined, a field in each database to be designated the “total” field, and a field in each database designated as key fields (match fields). The function generates the number of records in the primary database (P_NRECS), the number of records in the secondary database (S_NRECS), the sum of the total field defined in the primary database (P_TOTAL), the sum of the total field defined in the secondary database (S_TOTAL), and a DIFFERENCE column which subtracts the TOTAL fields from each other.

**EXERCISE: Compare two purchase databases**

- Open the Purchases-L4-2010 database.
- You need to add a Virtual Numeric field with two decimal places called TOTAL_GROSS. The field must use a criterion that calculates the sum of all the gross monthly purchases for the year (JAN_GROSS + FEB_GROSS...etc.)
Open the Purchases-L4-2011 database.

You need to add a Virtual Numeric field with two decimal places called TOTAL_GROSS. The field must use a criterion that calculates the sum of all the gross monthly purchases for the year (JAN_GROSS + FEB_GROSS…etc.)

On the Analysis tab, in the Relate group, click Compare.

The Compare Databases dialog box appears.

Click Select and select the Purchases-L4-2010 database as the secondary database.

Click Match to select key fields that are common between both databases.

Select SUPPNO for both and click OK.

From the Total field drop-down list for both databases, select TOTAL_GROSS.

In the File name field, enter Purchases 2011 vs 2010 for L4.

Click OK.

Fig. 4.13.1 – Compare database options