POWER BI

POWER BI BOOTCAMP

What is Business Intelligence?

- Everything concerning a business's past and current status is recorded as bits of data. Marketing numbers, human resources, company budgeting, sales volumes — you name it. As a business, you might be interested in extracting value from this information instead of just collecting it.
- Business intelligence (BI) is a set of technologies and practices for transforming business information into actionable reports and visualizations.
- Basically, business intelligence is a data analytical approach, answering the questions what was happening? and what is happening?. This type of data processing is also called descriptive analytics.
- Predictive analytics is concerned with forecasting based on data processing of past events. Instead of producing overviews of historical events, predictive analysis makes forecasts about future business trends. Those predictions are based on past events analysis.

Youtube video	
Business Intelligence Vs Data Science	
Backword Approach	Forword Approach
Provides reports But that can't predict future	provides patterns and reports that states how the data will look in future
pre-planed and restricted data - (No exploration)	No pre-planing data is required from source as per requirement
Helps to answer the question you know	Helps you to discover new question
Business useres	Data Scientist

Warehoused data	Distributed System (Data)
What happened ?	What will happened?
Input :- past data	Input :- Dynomic Required Data
Output:- Present Solution	Out put:- Future Prediction

WHO IS A BUSINESS INTELLIGENCE DEVELOPER?

A business intelligence developer is an engineer that's in charge of developing, deploying, and maintaining BI interfaces. Those include query tools, data visualization and interactive dashboards, ad hoc reporting, and data modeling tools.

WHAT ARE THE DIFFERENT BI TOOLS AVAILABLE ?

MICROSOFT POWER BI COMPONENTS

1. POWER BI DESKTOP

- POWER QUERY
- DATA, MODEL AND REPORTING

2. POWER BI SERVICE

- WORKSPACE
- APP
- DASHBOARD

3. POWER BI MOBILE APP

TYPES DATA	
1. Categorical Data (Qualitative Data)	2. Numerical Data (Quantitative Data)

Ordinal	Nominal	Continious Data	Descrete Data
Date	Store ID	Sales Amount	No. Item Sold
01-Jan-22	A	85	130
02- Jan-22	B	70.5	120
03-Jan-22	C	83.5	156
04-Jan-22	D	298.3	170
05-Jan-22	E	320.5	220

TYPES DATA- **1. Categorical Data (Qualitative Data)**

2. Numerical Data (Quantitative Data)

1. Categorical Data (Qualitative Data)

- **Ordinal**
- **Nominal**

2. Numerical Data (Quantitative Data)

- **Continious Data**
- **Descrete Data**

2.Numerical Data(Quantitative Data):- Numerical data show as a value we can calculate business Groth that called as Numerical data.

Numerical data is a data type expressed in number, rather than natural language description. Numerical data called as a Quantitative data numerical data is always collected in number form

- **Descrete Data:-** The Data is expressing whole Number Which have proper Interval in between them. It called Descrete Data. **Ex:-** Like No of Student, Shoe Size,

Examples of discrete data:

The number of students in a class.
The number of workers in a company.
The number of parts damaged during transportation.
Shoe sizes.
Number of languages an individual speaks.
The number of home runs in a baseball game.
The number of test questions you answered correctly

- **Continuous Data:-** In this Data there are not proper interval between them. that is Continuous Data. **Ex:-** Time, Height, Speed.

A Countinuous Variable can take any value think of it like this : If that number in the variable can keep counting then it's a Continuous Variable

The weight of a truck.

The speed of cars.

Time to wake up.

1. Categorical Data (Qualitative Data) :- Categorical data is represented Dimension Value of Business. It Also called as Qualitative Data.

Categorical Data is a collection of information that is divided into groups. I.e. If an organization or agency is trying to get a biodata of its employee the resulting data is referred to as categorical.

- **Ordinal Data:-** In this Segment we get data in proper Sequence

  Ordinal categorical attributes have some sense or notion of order amongst its values., education level and employment roles are some other examples of ordinal categorical attributes.

Ex:- Like Dates , Season, etc.

Examples of ordinal variables include: socio economic status ("low income","middle income","high income"), education level ("high school","BS","MS","PhD"), income level ("less than 50K", "50K-100K", "over 100K"), satisfaction rating ("extremely dislike", "dislike", "neutral", "like", "extremely like")

- **Nominal Data:-** In this Segment Data will not be in proper order.
  Ex:- like Name, Store ID, Product Id , Product Name.

These categories cannot be ordered in a meaningful way. Nominal data are used to label variables without any quantitative value. Common examples include **male/female (albeit somewhat outdated), hair color, nationalities, names of people**, and so on. In plain English: basically, they're labels (and nominal comes from "name" to help you remember)

Data Analytics

Youtube video	
Data Analytics Life cycle	
1. Discovery	**4. model Building**
Domain knowledge	**Bulding of model**
Resources	**Execution of model/ method**
Data	**Fast hardware & poralul processing**
People (stakeholder)	**5. Communication Results**
Time	**Success or Failure**

Froming of Problem	**Key Finding Business Value & Summarised Narrative to stakehoulders**
2. Data Preporation	**6. Operationalise**
ETL (Extract, Transform and Loadig)	**Final Report , briesing**
Data Conditioning	**Code & Technical Documents are Submitted/delivered (A.G)**
data Clening	
Data Selection	
3. Model Planning	
Clustering, Association Rules,Regression & Classification	
Selection of variables and Method and technics	

- It is a process of inspecting, cleansing, transforming, and modelling data so that we can derive some useful information from the data and use it for future predictions.
- Data analysis tools make it easier for users to process and manipulate data, analyse the relationships and correlations between data sets, and it also helps to identify patterns and trends for interpretation.

There are mainly three stages of Data Analytics

- Get the Data
- Analyse the data
- Communicate the result

Get The Data

- A Data Analyst's first point of all is having access to the data source.

- This is the stage where the individual works with multiple data sources like databases, spreadsheets, to get the necessary data before doing analysis on the data.
- There is need for the data analysts to be strong at query languages, working with spreadsheet.
- **Skills:** SQL, spreadsheet like MS Excel, Google sheets

Analyse The Data

Youtube video	
Data Visualisation	
Information	Pictorial/ Graphical, Representation
Decision Making	
Finding Solution to Problems	
for understanding the data clearly	
To find Relationship among the Data	
Comparative Analysis (A.G)	

We have make that All thing visualisation from.

Ex:- Python/R, problem-solving skills, Power BI/Tableau/Google Data Studio, attention to detail, curious mindset, business domain.

Whenever you get a data you will like this way

Ex:- Sales table, Product table, Product sub category, monthly Revenue, Monthly order quantity

• At this point, a lot is being demanded from the Data Analyst as this is where the core of the work gets done.

• The individual is involved in going through the data, applying a methodical and logical approach with the necessary questions in mind. During this stage, insights are gotten from the data, patterns and trends are also observed.

• The critical questions in the mind of a Data Analyst is Why and What. The individual is interested in answering why the following trends are found in the data and what could possibly be the causes of the observed trends and what can be done to improve on it.

• This is where the problem-solving skills of a Data Analyst is much appreciated as it is expected that the individual should be able to approach the data from different critical perspectives and then come-up with possible resolutions that will help improve the situation.

• Also, in order to effectively answer these questions, a good knowledge of the business domain is very necessary.

An enterprise data warehouse (EDW) is **a database, or collection of databases, that centralizes a business's information from multiple sources and applications, and makes it available for analytics and use across the organization**.

Communicate The Results (Data)

You have to Communicate what you have to done. You have to express that thing that is thing over here. You have to express what you done in this report on front of client.

• Communication is very important in the Data Analysis field.

• This is because all insights gotten from the data needs to be communicated appropriately in order to be appreciated and implemented by the involved business stakeholders.

• This stage is where the analyst presents the answers to the Why and What questions that is asked in the second stage.

• The results must be presented in a way that it is not overwhelming and without brain clutter.

• Most of the skills needed at this stage is more of soft skills than technical as most of the technical work is already covered in the previous stages.

Skills: Organization skills, communication and presentation skills.

Data Warehousing

A data warehouse is a type of data management system that is designed to enable and support business intelligence (BI) activities, especially analytics. Data warehouses are solely intended to perform queries and analysis and often contain large amounts of historical data.

The place where all the business data is stored is called data warehouse.

data warehouse, also known as an enterprise data warehouse, is a system used for reporting and data analysis and is considered a core component of business intelligence.

What is Data warehouse:- Data warehouse is a collection of past data used for futures Decision.

Old data Analytics used for Futures Business.

Future business growth increase

Collecting data from various source

Listing and reporting of collecting data.

Analytic and processing of data

A Data warehouse exists on top of several data bases and used for business intelligence.

Data warehouse consumes data from all these databases and Creates a layer optimized to perform data analytics.

Schema is done on import.

Detabace

Large amount of data that is stored in a computer and can easily be used, added to, etc.

Database, also called electronic database, **any collection of data, or information, that is specially organized for rapid search and retrieval by a computer**. Databases are structured to facilitate the storage, retrieval, modification, and deletion of data in conjunction with various data-processing operations.

What is Database :- database is manage to data

Data base are typically Structured with a defined Schema.

Items are organized as a set of tables with columns and Rows.

Columns include attributes and rows indicate an object or entity.

Database is Designed to be transactional and they are not designed to performed data analytics.

Most relational databases contain many tables. For example, a database might have a Customers table, a Products table, an Orders table, and many more. In fact, it's not uncommon for a corporate database to contain hundreds of tables.

Other Notes only Information purpose

DBMS stands for Database Management System. DBMS is a system software responsible for the creation, retrieval, updation, and management of the database. It ensures that our data is consistent, organized, and is easily accessible by serving as an interface between the database and its end-users or application software.

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database.

RDBMS stands for Relational Database Management System. The key difference IS , compared to DBMS, is that RDBMS stores data in the form of a collection of tables, and relations can be defined between the common fields of these tables. Most modern
database management systems like MySQL, Microsoft SQL Server, Oracle, IBM DB2, and Amazon Redshift are based on RDBMS.

The RDBMS provides an interface between users and applications and the database, as well as administrative functions for managing data storage, access, and performance.

OLTP and OLAP

OLTP stands for Online transaction processing and is primarily used for data processing and not for data analysis.	OLAP stands for Online Analytical Processing and is used for data analysis and not data processing.
OLTP is for running the business	OLAP is for analyzing the business.

Why can't we use OLTP for analysis?

- In the world of sports, let's say cricket; Who is the best cricket player? Or in terms of demographic data collection, In which state, the population is increasing? In which state, the male population is increasing? In which state, the female population is increasing?
- For analysis, we need to study the data for a certain period. We will need to analyze the past data to answer the above question, data for the last 5–10 years, some historical observation is mandatory.
- OLTP stores the real-time data, but it does not store the historical data. The historical loading takes time and needs study of data for analysis. If we use OLTP for both the purpose (also the work of OLAP), the OLTP will become very slow. The real-time data contains transactional data, and the transactions are required to be extremely fast for great user experience within the application.
- So, the industry introduces another database OLAP for analysis.
- If someone asks like who scored maximum runs in today's match? Then, OLTP is best suited. But for the question like who scored maximum runs in ODI till date? Then, OLAP will be needed.

Source of Data

- The data is stored in OLTP via interactions in applications and real-time transactions in the application.
- For OLAP, the source of data is OLTP or file system
- The data from OLTP or filesystem is moved to OLAP every day, every week or every month depending on the requirement.
- The process of moving data from OLTP to OLAP is the ETL(**extract, transform, and load)** process. Extract from the source, the transformation of data and loading of data to a database.
- So, the transformation is critical as it happens to depend on the requirement of analysis. First, convert the data into some particular form like a date should be in some specific format etc. Then load the data into the warehouse.
- This warehouse is the OLAP. Any Datawarehouse system is an OLAP system.

Example

OLTP	OLAP
• Online banking like payment transactions • Booking transactions like online train/airline ticket booking • Sending a text message with meta details stored	• A company might compare its product sales in September with sales in October in different product categories like laptops, mobiles, TVs, then compare those results with another location which may be stored in a separate database.
• Order entry in e-commerce websites. • Add a product to shopping cart	• Amazon analyzes purchases by its customers to come up with a personalized homepage for a user with products suggestion powered by a better recommendation system.

Feactures	OLTP	OLAP
Characteristic	Operational. P	Information P
Orientation	Transaction	Analysis
User	Normal Emp	Knowledgeble Person
Function	Day to Day Operation	Decision Support
Data	Current	Historical / Summarised
Access	Read & Write	Only Read
Focus	Data in	Information Out
Number of Record Accessed	100s	Millions
No of users	1000s	100s
DB Size	Max GB	>TB

OLTP or ***Online Transaction Processing*** is a type of data processing that consists of executing a number of transactions occurring concurrently—online banking,

OLAP (for **online analytical processing**) is software for performing multidimensional analysis at high speeds on large volumes of data from a data warehouse, data mart, or some other unified, centralized data store.

Types of tables

Raw Data & OLTP

Sales			
Date	Storename	Address	Sales
1/12/2020	A	Mumbai	100
2/12/2020	B	Pune	200
3/12/2020	C	Delhi	300
4/12/2020	A	Mumbai	250
5/12/2020	B	Pune	100
6/12/2020	D	Banglore	150
7/12/2020	E	Nagpur	400

TARGET			
Date	Storename	Address	TARGET
1/12/2020	A	Mumbai	100
2/12/2020	B	Pune	200
3/12/2020	C	Delhi	300
4/12/2020	A	Mumbai	250
5/12/2020	B	Pune	100
6/12/2020	D	Banglore	150
7/12/2020	E	Nagpur	400

ORDER			
Date	Storename	Address	ORDER
1/12/2020	A	Mumbai	100
2/12/2020	B	Pune	200
3/12/2020	C	Delhi	300
4/12/2020	A	Mumbai	250
5/12/2020	B	Pune	100
6/12/2020	D	Banglore	150
7/12/2020	E	Nagpur	400

Dimension Table			
Store ID	Storename	Address	Manager
101	A	Mumbai	Sehul
102	B	Pune	Rahul
103	C	Delhi	GHOLAP
104	D	Banglore	Mohit
105	E	Nagpur	AKRUJ

FACT TABLE

	Sales	
Date	Store ID	Sales
1/12/2020	101	100
2/12/2020	102	200
3/12/2020	103	300
4/12/2020	101	250
5/12/2020	102	100
6/12/2020	104	150
7/12/2020	105	400

TARGET		
Date	Store ID	TARGET
1/12/2020	101	100
2/12/2020	102	200
3/12/2020	103	300
4/12/2020	101	250
5/12/2020	102	100
6/12/2020	104	150
7/12/2020	105	400

ORDER		
Date	Store ID	ORDER
1/12/2020	101	100
2/12/2020	102	200
3/12/2020	103	300
4/12/2020	101	250
5/12/2020	102	100
6/12/2020	104	150
7/12/2020	105	400

REPORTING TABLE

MONTH NAME	STORE NAME	STORE ID	TOTAL SALES	TOTAL TARGET	TOTAL ORDER
20-Dec	A	101	350	350	350
20-Dec	B	102	300	300	300
20-Dec	C	103	300	300	300
20-Dec	D	104	150	150	150
20-Dec	E	105	400	400	400

Types of tables

There are three types of table 1. Dimension 2.Fact 3. Reporting

1. Dimension

• Dimension Tables are where all descriptive entities stores that describe the numbers in the Fact Table. In other words, the numbers in the Fact Table are described by the entities of the Dimension Tables.

• We can do filtering, grouping and labeling functions on these Dimension Tables.

A dimension is a structure that categorizes facts and measures in order to enable users to answer business questions. Commonly used dimensions are people, products, place and time. In a data warehouse, dimensions provide structured labeling information to otherwise unordered numeric measures.

It is a table in which we manage category, do Filtering, there is No numerical Value except Date, Call as Dimension Table.

2. Fact

• Fact Table is the table that contains the business numbers in a data warehouse. There are some cases in which the numbers are in the Dimension Tables too. This table contains all the primary keys in the Dimension Tables as foreign keys.

• We can do aggregation on these Fact Tables.

In data warehousing, a fact table consists of the measurements, metrics or facts of a business process. It is located at the center of a star schema or a snowflake schema surrounded by dimension tables. Where multiple fact tables are used, these are arranged as a fact constellation schema.

Thus, the fact table consists of two types of columns. The foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed. In this example, **the customer ID column in the fact table is the foreign key that joins with the dimension table**.

It is a table having some business value's / Fact value's & It is depends on dimension table.

3. Reporting table

• Reporting tables are derived by joining the fact tables and dimension tables. Such tables contain the actual values of the dimensional attributes instead of having the keys of the dimension tables.

• Such tables are useful when we need to present the data in a tabular report. However, it is not used much in BI because the same objective can be achieved by facts and dimensions if we have proper data modelling in place.

Reporting table :- The table which is derived from Dimension table & fact table & Such table used to present in Tabular form.

What is Tabular Report ?

A tabular report **displays descriptive statistics in a table format**. A tabular report lets you place data into classes or categories for analysis and calculate statistics, such as counts, sums, means, and percentages on these classes.

Data which is in the form of table

NORMALIZATION

• Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updating anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

1. It was first proposed by Edgar F. Codd as part of his relational model.

Normalization or normalisation refers to **a process that makes something more normal or regular**.

It is process which use to minimizing redundancy you have whatever a data, That data we can make easy to understand so such kind of thing we can say normalization. We can display that data in the table.

• There are different types of normalization:

• **1st Normal Form** • 2nd Normal Form • 3rd Normal Form • BCNF

• 1st Normal Form

• If a relation contains composite or multi -valued attribute, it violates 1st NF, or a relation is in 1st NF if it does not contain any composite or multi -valued attribute. A relation is in 1st NF if every attribute in that relation is singled valued attribute .

• **Example** – Relation STUDENT in Table 1 is not in 1NF because of multi - valued attribute STUD_PHONE. Its decomposition into 1NF has been shown in Table 2.

1st Normal Form

Tabal 1

Student No	Studend Name	Student Phone	student State	Student Country
1	Ram	9716271721, 9871717178	HARYANA	INDIA
2	Ram	9898297281	PUNJAB	INDIA
3	Suresh	99222121	PUNJAB	INDIA

Conversion to First normal form

Tabal 2

Student No	Studend Name	Student Phone	student State	Student Country
1	Ram	9716271721	HARYANA	INDIA
1	Ram	9871717178	HARYANA	INDIA
2	Ram	9898297281	PUNJAB	INDIA
3	Suresh	99222121	PUNJAB	INDIA

First normal form is a property of a relation in a relational database. A relation is in first normal form if and only if no attribute domain has relations as elements. Or more informally, that no table column can have tables as values.

- In a one table have to a one value
- If there are two values convert the table to another form and insert a row.

• 2nd Normal Form

• To be in 2NF, a relation must be in 1st NF and relation must not contain any partial dependency.

• **Partial Dependency** – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

Here

• COURSE_FEE cannot alone decide the value of COURSE_NO or STUD_NO

• COURSE_FEE together with STUD_NO cannot decide the value of COURSE_NO

• COURSE_FEE together with COURSE_NO cannot decide the value of STUD_NO;

Student No	Course No	Course Fee
1	C1	1000
2	C2	1500
1	C4	2000
4	C3	1000
4	C1	1000
2	C5	2000

-

Tabal 1

Student No	Course No
1	C1
2	C2
1	C4
4	C3
4	C1
2	C5

Tabal 2

Course No	Course Fee
C1	1000
C2	1500
C4	2000
C3	1000
C1	1000
C5	2000

-

Hence,

• COURSE_FEE would be a non-prime attribute, as it does not belong to the one only candidate key {STUD_NO, COURSE_NO}

• But COURSE_FEE is dependent on COURSE_NO, which is a proper subset of the candidate key.

• Non-prime attribute COURSE_FEE is dependent on a proper subset of the candidate key, which is a partial dependency and so this relation is not in 2NF.

- **2nd Normal Form :-** In Second Normalization we don't have partial Dependency in our table if have such a partial dependency the column which are dependent any other column we are making a separate that thing we are Creative another table such kind of Normalization we are say second Normalization form

---

3RD NF

• A relation is in 3NF, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

• Transitive dependency – If A->B and B->C are two FDs then A->C is called transitive dependency.

• FD set: {STUD_NO -> STUD_NAME, STUD_NO -> STUD_STATE, STUD_STATE -> STUD_COUNTRY, STUD_NO -> STUD_AGE}

Candidate Key: {STUD_NO}

• For this relation in table 4, STUD_NO -> STUD_STATE and STUD_STATE -> STUD_COUNTRY are true.
• So, STUD_COUNTRY is transitively dependent on STUD_NO. It violates the third normal form.

• To convert it in third normal form, we will decompose the relation STUDENT (STUD_NO, STUD_NAME, STUD_PHONE,

STUD_STATE, STUD_COUNTRY_STUD_AGE) as:

STUDENT (STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_AGE) STATE_COUNTRY (STATE, COUNTRY)

Student No	Student Name	Student State	Student Country	Student Age
1	RAM	HARYANA	INDIA	20
2	RAM	PUNJAB	INDIA	19
3	SURESH	PUNJAB	INDIA	21

Third normal form (3NF) is a database schema design approach for relational databases which uses normalizing principles to reduce the duplication of data, avoid data anomalies, ensure referential integrity, and simplify data management. It was defined in 1971 by Edgar F.

Third normal form:- Means transitive dependency

3rd Normalizatins form

Name	City	State
Rohit	Mumbai	Maharashtra
Rahul	Chennai	Tamil Nadu
Vinay	Mumbai	Maharashtra

Tabal 1

Name	City
Rohit	Mumbai
Rahul	Chennai
Vinay	Mumbai

Tabal 2

City	State
Mumbai	Maharashtra
Chennai	Tamil Nadu
Mumbai	Maharashtra

Types of Facts

• **Facts** are business numbers that are kept in fact tables.

• There are different types of facts based on its aggregation behaviour.

something that you know has happened or is true

• These are:

- Additive facts

- Semi-additive facts
- Non-additive facts

Additive facts:- **Additive facts** are measurements in a fact table that can be summed up across all dimensions.

Store Key	Product Key	Transaction - Date	Revenue
S1	P1	27-Aug-10	100
S2	P1	27-Aug-10	150
S3	P1	27-Aug-10	300
S1	P2	28-Aug-10	600
S2	P2	28-Aug-10	300
S1	P1	29-Aug-10	200
S2	P2	29-Aug-10	100
S3	P3	29-Aug-10	400

Stores wise Sales		
	S1	900
	S2	550
	S3	700

Product wise Sales		
	P1	750
	P2	1000
	P3	400

Daily Sales		
	27-Aug-10	550
	28-Aug-10	900
	29-Aug-10	700

Additive facts are **those facts which give the correct result by an addition operation**. Examples of such facts could be number of items sold, sales amount etc.

Semi-Additive facts:- **Semi-Additive facts** are measurements in a fact table that can be summed up across only a few dimensions keys. Following table is used to record current balance and profit margin for each id at a particular instance of time (Day end)

• In the right table, we cannot sum up current balance across Acct Id

• If we ask balance for Id 21653 we will say that 22000, not 22000+80000

• But the current balance can be summed on transaction date

Semi-Additive facts			
Acc Id	**Trans Date**	**Current Bal**	**Profit Margin**
21653	27-Aug-09	80000	0.06
21654	27-Aug-09	120000	0.08
21653	28-Aug-09	22000	0.08
21654	28-Aug-09	48000	0.12

Semi-additive measures **can be aggregated across some dimensions, but not all dimensions**.

Non-Additive facts:- • **Non-Additive facts** are measurements in a fact table that can not be summed up across any dimension.

Ex: Percentages and ratio columns.

NON ADDITIVE FACTS

Month	Sales	Profit	Profit Margin
Jan	500	250	50%
Feb	800	400	50%
Mar	1000	500	50%
Apr	700	300	42%
May	900	600	66%
Jun	1200	500	41%
Jul	1400	800	57%
Aug	1000	700	70%
Sep	500	200	40%
Oct	600	300	50%
Nov	1500	500	33%
Dec	500	100	20%
			47% **Wrong**
Year	**10600**	5150	**49%** **Write**

Hypothetical Situation

Month	Sales	Profit	Profit Margin
Jan	1000	250	25%
Feb	1000	400	40%

Mar	1000	500	50%
Apr	1000	300	30%
May	1000	600	60%
Jun	1000	500	50%
Jul	1000	800	80%
Aug	1000	700	70%
Sep	1000	200	20%
Oct	1000	300	30%
Nov	1000	500	50%
Dec	1000	100	10%

43%

Year	**12000**	**5150**	**43%**

	S1	S2		
Got	**80**	**45**		**125**
All	**100**	**50**		**150**
	80%	**90.00%**		83%
		85%		

Non-additive measures are **measures that cannot be aggregated across any of the dimensions**. These measures cannot be logically aggregated between records or fact rows. Non-additive measures are usually the result of ratios or other mathematical calculations.

Types Of Fact in Data warehouse

1. **Additive Facts- Facts / Measures on which aggregation can be performed across all Dimension.**
2. **Semi –Additive Facts- Facts/ Measures on which aggregation can be performed across some Dimession.**
3. **Non –Additive Facts- Facts / Measures on which aggregation can be performed across Non of Dimession.**

Slowly changing dimension

What is a Slowly Changing Dimension? A Slowly Changing Dimension (SCD) is **a dimension that stores and manages both current and historical data over time in a data warehouse**. It is

considered and implemented as one of the most critical ETL tasks in tracking the history of dimension records.

• **Slowly Changing Dimension** is a term where dimensions change slowly over time, this dimension stores old values or data from a dimension attribute.

Slowly Changing Dimension:- It table change very slowly

Ex:- 1 Year, 2 Year, 3 Year.

• SCD Types: • Type 1 • Type 2 • Type 3

SCD: Type 1:- • This method does not take any special action if there is a change in a dimension.

• The new information simply overwrites the original information. In other words, no history is kept.
• **Ex**: Krishna is a customer with ABC Inc. He lives in Mumbai. So, the original entry in the customer table has the record shown in Table 1.

He gets transferred to Pune. So, the new entry looks like Table 2 as per Type 0.

Table 1

Customer Key	Name	City
1001	Krishna	Mumbai

Table 2

Customer Key	Name	City
1001	Krishna	Pune

Types 1:- Where Historical data can not be maintained

- Where we can replace the value such types call as types one.
- Where no kind of the table is change
- We only replace the value

In a Type 1 SCD **the new data overwrites the existing data**. Thus the existing data is lost as it is not stored anywhere else. This is the default type of dimension you create. You do not need to specify any additional information to create a Type 1 SCD.

SCD: Type 2:- • A new record is added to the table to represent the new information. Therefore, both the original and the new record will be present.

• The new record gets its own primary key

• **Ex**: For the same example, below table is based on Type 2.

Customer Key	Name	City	Start Date	End Date
1001	Krishna	Mumbai	1/1/2020	15/05/2020
1005	Krishna	Pune	16/05/2020	31/12/2030

Types 2:- In this types we are maintaining the historical data.

- Where we are not Replacing a Value Where we are Just adding Row

A Type 2 SCD **retains the full history of values**. When the value of a chosen attribute changes, the current record is closed. A new record is created with the changed data values and this new record becomes the current record.

SCD: Type 3:- • There will be two columns to indicate the particular attribute of interest, one indicating the original value, and one indicating the current value.

• There will also be a column that indicates when the current value becomes active.

• **Ex**: For the same example, below table is based on Type 3.

Customer Key	Name	Original City	Current City	Effective Date
1001	Krishna	Mumbai	Pune	5/16/2020

Types 3:- In this data types historical data is maintain but that is not maintained hundred present around fifty present data in maintained.

- We add addition column but you not adding a additional row.
- This data use by Businesman

The new current value column holds the new dimension data coming from the OLTP system. This type of slowly changing dimension is used when a change in a dimension value must be tracked but the old value must be retained as part of the record, usually for reporting.

Ex

Emp Key	Name	Location
101	Sehul	Mimbai

Table 1

Emp Key	Name	Location
101	Sehul	Chennai

Table 2

Emp Key	Name	Location	Start Date	End Date	Active
101	Sehul	Mumbai	8/10/2019	16/09/2020	0
102	Sehul	Chennai	17/09/2020	1/6/2021	0
105	Sehul	Bangalore	2/6/2020	31/12/2021	0
106	Sehul	Pune	1/1/2022	1/12/2025	0
107	Sehul	Nagpur	2/12/2025	5/12/2030	1

Table 3

Emp Key	Name	Prew Location	Curr Location	Start Date
101	Sehul	Nagpur	Bangalore	7/7/2022

Types of Databases

• **Staging/Landing(name can be anything else):**

Data is imported to this database to be treated as base data to derive the facts and dimensions from here.

Staging / Landing :- If you have N number of data source collecting that data from there and put that data in this database.

A staging database is **a user-created PDW database that stores data temporarily while it is loaded into the appliance**.

A staging area, or landing zone, is an intermediate storage area used for data processing during the extract, transform and load process. The data staging area sits between the data source and the data target, which are often data warehouses, data marts, or other data repositories.

- **RTPD(name can be anything else):**

This DB contains the facts and dimensions along with the scripts to generate these facts/dimensions.

RTPD:- Data Engineer are doing Coding here.

RTPD is **a program that provides error-corrected communications with a DAS unit over a variety of media**.

- **Reporting(name can be anything else):**

This DB contains the reporting tables which are directly used by the reporting tools. These tables are derived from the facts and dimensions.

Schemas in DW

SCHEMAS in Detawarehouse:- Number of folder Create in Single Database we call that Schemas in Datawarehouse.

The database schema is a structure of a database described in a formal language supported by the database management system. The term "schema" refers to the organization of data as a blueprint of how the database is constructed.

- Schemas are like containers to categorise the database objects.
- We can use schemas when we don't want to have different databases for Landing/RTPD/Reporting.
- There are other categorization that we want to have in our database which can be achieved by schemas.

Server Means:- a computer that stores information that a number of computers can share

In computing, a server is a piece of computer hardware or software that provides functionality for other programs or devices, called "clients". This architecture is called the client–server model.

Server Means :- Who serve Something

- SSMS this is our Client
- SQL Express is our Server

Client Means:- one of a number of computers that is connected to a special computer (server) that stores shared information

In computing, a client is a piece of computer hardware or software that accesses a service made available by a server as part of the client–server model of computer networks. The server is often on another computer system, in which case the client accesses the service by way of a network.

SSMS:- **SQL Server Management Studio** (SSMS) is an integrated environment for managing any SQL infrastructure.

SQL:- **Structured Query Language** (***SQL***) is a standardized programming language that is used to manage relational databases and perform various operations on the data in.

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

Other note only practice use

Types of Database

Database is an organized collection of structured of information and there is different types of database landing and staging , RTPD, Reporting this three concept in database.

Fist is Staging is doing a collect all data in one place and understanding and analyzing easily that data.

Second RTPD Stand for ready to process data convert in Fact and Dimension with a help of codding and transfer that data in reporting that way as name Suggest RTPD .

Third Reporting in a reporting we convert in a data any visualization form and reporting table

Schema means Create a Number of folder in create a one data base. That call Schema that is Schema in warehouse .

SQL Introduction

SQL Stand for Standard query there is assign and maniculate a database and SQL is a domain specific language

SQL

SQL (PDF)

Definition :- Structured Query Language(SQL) is the database language using which we can perform certain operations on the existing database and, we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

Data Definition Language(DDL) :- DDL consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

CREATE – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).

DROP – is used to delete objects from the database.

ALTER-is used to alter the structure of the database.

TRUNCATE–is used to remove all records from a table, including all spaces allocated for the records are removed.

Data Query Language(DQL) :- DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it.

Example of DQL:

SELECT – is used to retrieve data from the database

Data Manipulation Language(DML) :- The SQL commands that deals with the manipulation of data present in the database belong to DML and this includes most of the SQL statements.

Examples of DML:

INSERT – is used to insert data into a table.

UPDATE – is used to update existing data within a table.

DELETE – is used to delete records from a database table.

Data Control Language(DCL) :- DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions and other controls of the database system.

Examples of DCL commands:

GRANT-gives users access privileges to the database.

REVOKE-withdraw user's access privileges given by using the GRANT command.

Transaction Control Language(TCL)

TCL commands deal with the transaction within the database.

Examples of TCL commands:

COMMIT– commits a Transaction.

ROLLBACK– rollbacks a transaction in case of any error occurs. (Akruj)

SAVEPOINT

JOINS

A SQL join is a way to combine rows from two or more tables based on a common column between them.

1.Inner joins **combine records from two tables whenever there are matching values in a field common to both tables**. You can use INNER JOIN with the Departments and Employees tables to select all the employees in each department.

2. LEFT JOIN command **returns all rows from the left table, and the matching rows from the right table**. The result is NULL from the right side, if there is no match.

3. RIGHT JOIN keyword **returns all records from the right table (table2), and the matching records from the left table (table1)**. The result is 0 records from the left side, if there is no match.

4. FULL OUTER JOIN keyword **returns all matching records from both tables whether the other table matches or not**. So, if there are rows in "Customers" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customers", those rows will be listed as well.

5. CROSS JOIN is **used to combine each row of the first table with each row of the second table**. It is also known as the Cartesian join since it returns the Cartesian product of the sets of rows from the joined tables.

Cross Join :- Each row of Fist table is Combine with each Row of Second Table is Call as **Cartesian join** or Cross Join

Constraints

SQL Constraints are rules used to limit the type of data that can go into a table, to maintain the accuracy and integrity of the data inside the table. Constraints can be divided into the following two types

Column level constraints - Limits only column data.

Table level constraints - Limits whole table data.

Constraints are used to make sure that the integrity of data is maintained in the database. The following are the most used constraints that can be applied to a table.

1.**NOT NULL** 2. **UNIQUE** 3. **CHECK** 4.**DEFAULT** 5.**PRIMARY KEY** 6.**FOREIGN KEY**

1. NOT NULL CONSTRAINTS

• NOT NULL constraint restricts a column from having a NULL value. Once NOT NULL constraint is applied to a column, you cannot pass a null value to that column.

• It enforces a column to contain a proper value. One important point to note about this constraint is that it cannot be defined at the table level.

• In the following example, ID and NAME are not accepted the Null values but Age can accept there is NOT NULL constraint not used.

2.UNIQUE CONSTRAINTS

UNIQUE constraint ensures that a field or column will only have unique values. A UNIQUE constraint field will not have duplicate data

3.CHECK CONSTRAINTS

CHECK constraint is used to restrict the value of a column between a range. It performs check on the values, before storing them into the database. Its like condition checking before saving data into a column

4.DEFAULT CONSTRAINTS

The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

5.PRIMARY KEY CONSTRAINTS

The primary key constraint uniquely identifies each record in a database. A Primary Key must contain unique value and it must not contain a null value.

6.FOREIGN KEY CONSTRAINTS

FOREIGN KEY is used to relate two tables. The FOREIGN KEY constraint is also used to restrict actions that would destroy links between tables.

If you try to insert any incorrect data, DBMS will return an error and will not allow you to insert the data.

DELETE COMMAND

• Delete is a Data Manipulation Language (DML) command.

• In Delete command we can specify particular tuple that we want to delete.

• Delete command use WHERE clause.

• In Delete command data restore using ROLLBACK.

• Delete command slower than Truncate.

• Delete command lock the row/tuple before deleting it.

• Delete command eliminate the tuples one-by-one.

• In Delete command transaction log for each deleted tuple.

• Delete command activates the trigger to fire.

• **Syntax**: DELETE FROM table_name WHERE [conditon]

TRUNCATE COMMAND

• Truncate is a Data Definition Language (DDL)command.

- Truncate command deletes all the tuples from a table.
- Truncate command not use WHERE clause.
- In Truncate command ROLLBACK is not work that's why data can not restore .
- Truncate is faster than Delete.
- Truncate command lock data page before deleting table data.
- Truncate command delete the entire data page containing the tuples.
- In Truncate command transaction log for each deleted data page.
- Truncate command does not activate the triggers to fire.
- **Syntax**: RUNCATE TABLE table_name;

Bulen opration

1. **Conjection – And**
2. **Dirjunction- Or**
3. **Negation- Not**
4. **Nand- Not+And**
5. **Nor – Not+Or**
6. **Exor**
7. **Exnor**

P	Q	P- AND - Q	P- OR- Q	NOT - P	NOT- Q
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

Comparison operators can compare numbers or strings and perform evaluations. Expressions that use comparison operators do not return a number value as do arithmetic expressions. Comparison expressions return either 1 , which represents true, or 0 , which represents false.

Ex:-

Comparison Operator Symbol	Name
==	compares operands
<>	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
!~	does not contain
~	string comparison
like	like

Arithmetic operations is **a branch of mathematics, that involves the study of numbers, operation of numbers that are useful in all the other branches of mathematics**. It basically comprises operations such as Addition, Subtraction, Multiplication and Division.

Binary operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation

Only Note Index Only there not syllabus

INDEX

Clustered indexes are **indexes whose order of the rows in the data pages corresponds to the order of the rows in the index**. This order is why only one clustered index can exist in any table, whereas, many non-clustered indexes can exist in the table.

Clustered indexes:- Dictonary Types

Non Cluster Indexes:- Normal Text Book

STORE PROCEDURE

Store Procedure is a Prepaid SQL Code each can be save and reuse again again Whenever user Requirement.

Main purpose of Store procedure Hide the direct SQL Queries from the code and improve the performances of database operation

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

UNION

UNION operator is **used to combine the result-set of two or more SELECT statements**. Every SELECT statement within UNION must have the same number of columns. The columns must also have similar data types. The columns in every SELECT statement must also be in the same order. 1.Number of Columns must be Same in Number. 2. The domain of every columns must be Same.

UNION ALL

The SQL Union All operator **combines the result of two or more Select statement similar to a SQL Union operator with a difference**. The only difference is that it does not remove any duplicate rows from the output of the Select statement.

VIEW

View means any command and Code we are giving to the User to see only the particular things witch we have to show them

SubQuery

Query inside Query that is Called as SubQuery
Inside the Query there are another query that is called as SubQuery (SubQuery Called = Inner Query = NestedQuery)

A subquery is **a query that is nested inside a SELECT , INSERT , UPDATE , or DELETE statement, or inside another subquery**.