SQL

→ Essential language to survive in data Industry
→ By learning SQL, we can prepare for four different roles in the IT Industry they are
1. Data Analyst
2. Data Scientist
3. Data Engineer
4. Software Engineer

Database is nothing but a collection of tabular data built across rows & columns.

Clause :- clause is a special keyword in SQL such as WHERE, GROUP BY, etc.. that has a predefined purpose such as filtering, grouping records etc..

Count COUNT(*) function where used to Retrieve the count/Number of Records/rows
(numerical count)

Distinct Values :- values that are different from each other.
Basicly, which retrieves the unique values from the column.

% Wild card search Ex:- "% THOR %"
which basicly, Retrieves all the records where "THOR" keyword Exists, ~~without~~ which doesn't care if there is any strings attaching before & after the keyword

→ With the help of USE function, we can indicate the query to use a particular database, especially when there are multiple databases.

→ "*" means all columns. Using "*" after the SELECT query will select all columns of a database.

→ use LIKE function and '%' to filter the rows based on a text value.

→ "ORDER BY" function which is used to sort the records, defaultly which sorts the records in ascending order, If we want to sort the records in descending order we can use KEYWORD "DESC" after ORDER BY Similarly, "ASC" for Ascending order.

SQL is not case sensitive

Ways to comment down the lines in SQL are "--", # (can be used before the lines to make it as a comment.

→ MAX, MIN, and AVG are the common summary analytics function of SQL.

→ We can define a custom column header name by using 'as' clause

→ GROUP BY clause will help us to create a summary of metrics such as average, count etc.. for selected columns.

→ GROUP BY and HAVING clauses are often used together.

→ the column you use in HAVING should be present in SELECT clause, whereas WHERE can use columns that is not present in select clause as well.

Order of Execution

FROM
↓
WHERE
↓
GROUP BY
↓
HAVING
↓
ORDER BY

Ex:- Explicitly derivering columns from table (Derived columns runs at the end of all above operations

→ IF function is often used in SQL queries, when we have more than 2 conditions, we need to use CASE and END function instead of IF function

→ SQL inbuilt functions such as YEAR, CURDATE etc..

→ "OFFSET" dictates the number of rows to skip from the beginning of the returned data before presenting results.

→ "LIMIT" clause is used to specify the number of records to return.

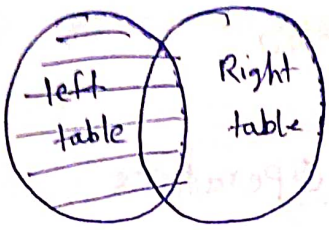
Why companies use multiple tables rather than one fact table to store data.

Because, → using multiple tables can save space by avoiding repetition
→ which, helps to organize the data better.
→ Finally, which will become simpler/ easier to make any changes in the tables.

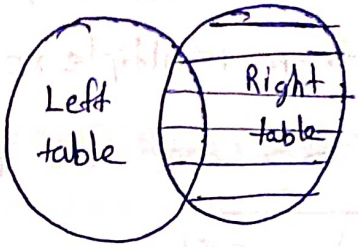
→ In SQL, mapping multiple tables can be done using JOIN clause.

JOINS

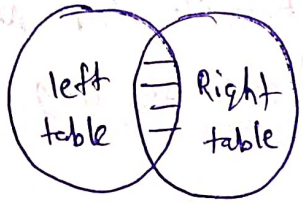
LEFT JOIN



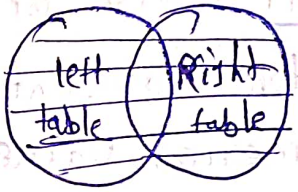
RIGHT JOIN



INNER JOIN



FULL JOIN



LEFT, RIGHT, FULL JOINS are also called as OUTER JOINS

→ UNION clause will enable to perform FULL JOIN

→ JOIN and ON clauses used together will enable to merge two tables

→ JOIN, ON & AND clause will enable to merge two tables based on multiple columns.

→ We can assign an abbreviated letter next to the table name to shorten the query length

Ex:- movies m,

→ CONCAT clause in SQL will help to combine two text strings.

CROSS JOIN :-

Is useful, when we do not have any common column between two tables.

→ Entity Relationship Diagram helps us to Understand the relationship b/w the tables.

→ Group_Concat function helps us to combine text from multiple rows to one row

→ "<< >>" are the Basic Numerical Operators used in SQL

→ Numerical queries "AND, OR, BETWEEN, IN"

→ "LIMIT" clause can be used to fetch the top 'N' or bottom 'N' amount of records. 'N' can be any numerical value.

→ "OFFSET" clause will help us to skip certain Number of rows in the result.

SubQueries can be act as which are generated output that will be used as input to main query. where, queries that provide a single record, list or even a table as output can be used as a subquery.

→ IN, ANY, & ALL clauses expect a list as input.

"ANY" clause executes the condition for any of the values on the list that meets the condition which is the minimum value by default.

"ALL" clause executes the condition where all the values on the list meet the condition which is the maximum value of the list.

→ A subquery is called a co-related query when its execution depends upon the statement(s) written after the bracket.

→ To optimize the query performance, by choosing a subquery (or) co-related query depending on its Performance. which can be known by the help of "EXPLAIN ANALYSE" clause, (BY. Running this clause on before any query) will provide the query execution Plan through which one can understand the query Performance.

We can't use Calculative columns ~~(A+B)~~ as an conditional statement in WHERE clause

Common Table Expression [CTE]

→ which creates a temporary table within a query.

→ WITH and AS clauses are used in combination to create CTE

→ We can create multiple CTE's inside a query.

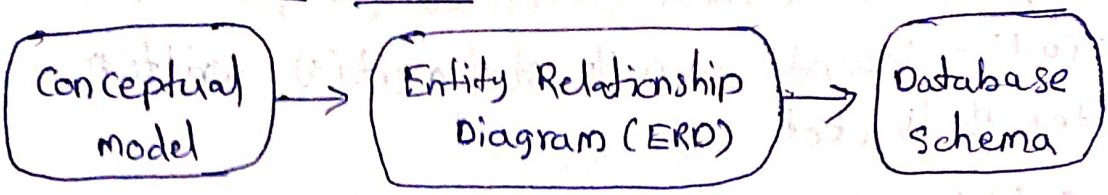
Benefits to use CTE

→ Simple Queries (which Increases, Query Readability

→ Same Resultset can be referenced multible times (means, within the scope of "with" statement/clause we can use as many times as we want) (CTE) [Query Reusability]

→ gives potential candidates for views [Visibility for Creating Data Views]

Database Design



Data Integrity :- Basically, It is a way to maintain the Accuracy and Consistency of data over its life cycle.

Normalization :- ~~which is a proce~~ It's a process of Organizing database so that we can avoid duplication & Increase data integrity

Types of Normalizations :- 1 NF, 2 NF, 3 NF

~~link~~ **link table** :- ~~it's~~ a term used to describe a table that acts as a link between two tables.

Numeric

Whole Numbers (Can include Negative numbers)

To represent whole Numbers we use Integer datatypes.

→ TINYINT
→ SMALLINT
→ INT
→ BIGINT

Numbers with Decimal Point (can include negative numbers)

To represent No with decimal points we use Floating data types

→ FLOAT
→ DOUBLE
→ DECIMAL

difference b/w CHAR and VARCHAR

VARCHAR uses the spaces ~~according on~~ based the string size and saves the Reserve space, but in case CHAR it uses the Entire reserved space, Even though ~~the~~ their is No values existed, which Actually is a loss of space.

∴ ~~So~~ VARCHAR is Preferably than CHAR

Ex:-

VARCHAR (50) — String, Cricket, In case of VARCHAR it takes only 7 character Space even though it is having size is 50

CHAR (50) — Here, it consumes all the spaces ~~to~~ reserved

String

Fixed length

Ex:- CHAR (3)

"USD"

"INR"

"A "

↓

where Empty space will be filled

Variable length

Ex:- VARCHAR(150)

"KGF 2"

"The Pusha The RISE"

ENUM

(which is basically used when we know the No of categories that existed in the column)

Ex:-

ENUM ('small', 'medium', 'large')

ENUM ('billions', 'millions', 'thousands')

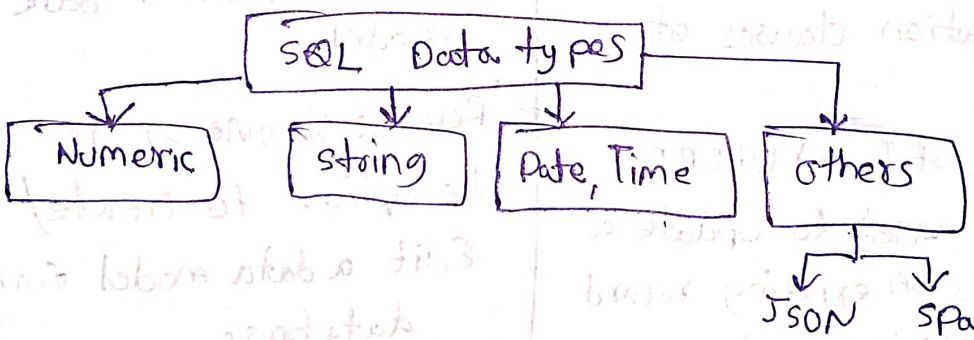
2

→ DATE, YEAR and DATETIME are the major data types under Date, Time category.

→ JSON is a popular & Efficient data type to store massive amount of data.

"→" operator is used to Extract a JSON object.

→ SPATIAL datatype is used to represent geospatial data types likes latitude, & longitude etc..



Primary key

→ It's an unique identifier which cannot have any duplicates.

→ Primary key that already exists in database is called Natural key.

Primary key that is generated by user artificially is called Surrogate key.

Composite key is a Primary key that is generated by combining multiple columns.

Foreign key

→ Foreign key column is a way to Establish relationship between two tables

→ the Parent table Contains the Primary key which is Connected to the child table which Contains the foreign key.
(unique value, column)
(where unique values of Primary key columns are repeated)

Primary key existence table are called as Parent table

Foreign key existence table are called as child table → which is always a part of child table

→ the key benefit of creating a relationship is to prevent having undesirable records in the database.

INSERT, UPDATE and DELETE are the primary database modification clauses of SQL

UPDATE, SET and WHERE clauses used to update a value to an existing record or multiple records.

Forward Engineering is the option to create a database from a data model.

'Reverse Engineer' is the option to create/ Edit a data model from database

→ 'synchronize' updates between the model & actual database (when used) If required to do any changes after creating a database)

Examples of Relational Database system [RDBMS]

sales transactions, Banking system,
Student Management system

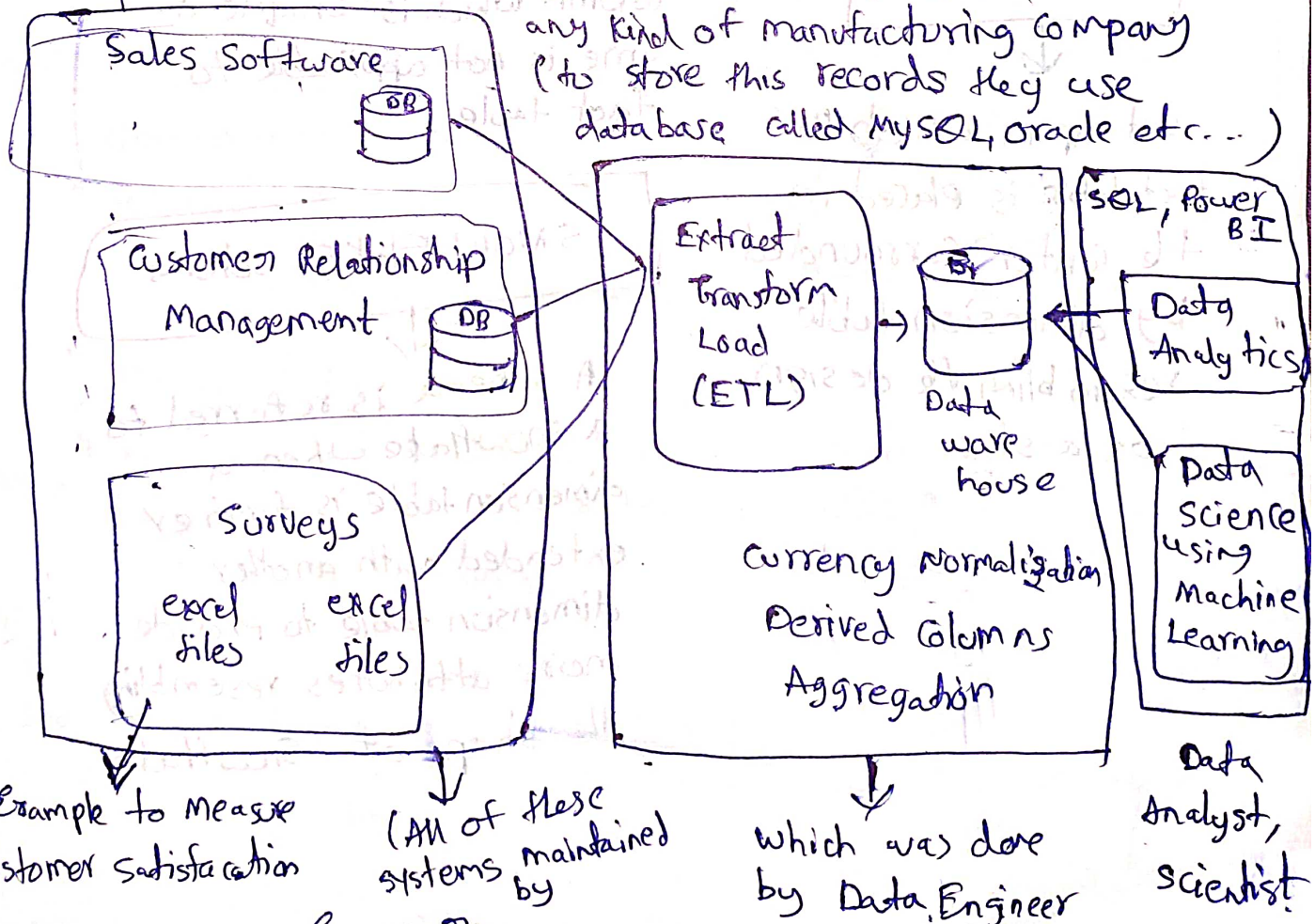
Three Popular Relational-Database Servers are :- Oracle, Microsoft SQL server, MySQL

For Big or Enterprise level application Oracle & Microsoft SQL server is preferable.

For smaller & Medium level Enterprise MySQL is Preferable

Non-Relational Database Server are :- MongoDB, Couch DB relax, apache Cassandra

→ Retail stores, should have this kind of softwares to maintain all their invoice sales record transactions from any kind of manufacturing company (to store this records they use database called MySQL, oracle etc...)

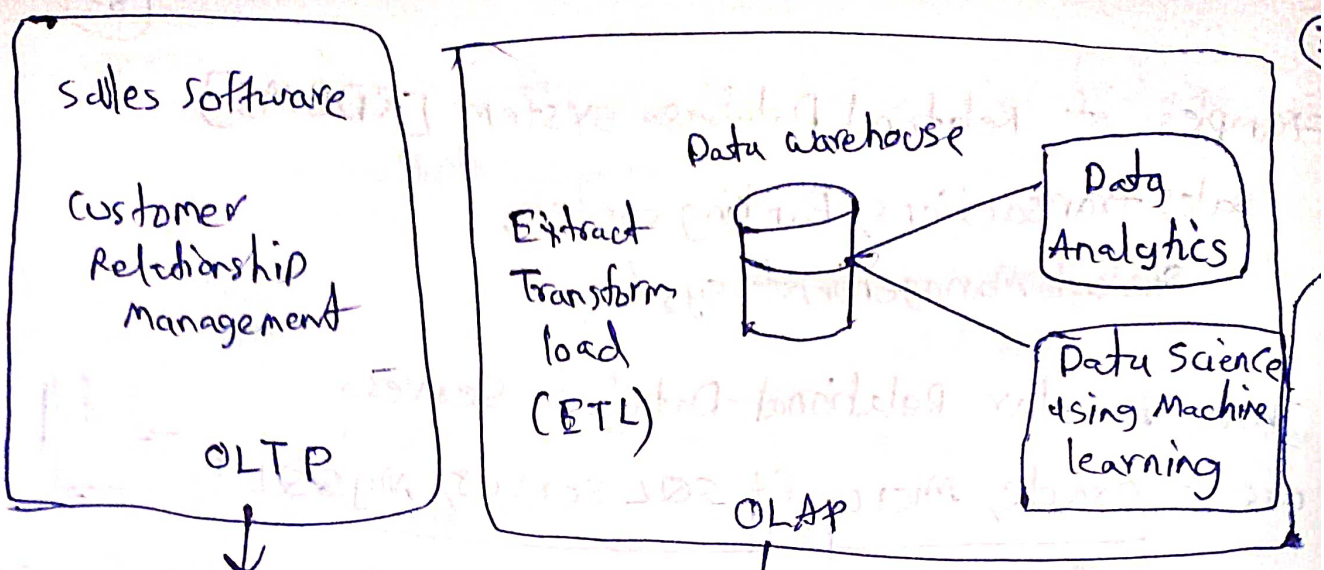


Example to measure Customer Satisfaction (Software Engineer)

(All of these systems maintained by

which was done by Data Engineer

Data Analyst, Scientist



online transaction Process

Online Analytical Processing

Fact table

Contains the transaction data (sales data)

Dimension table

→ Contains the attributes of dataset such as customer, region, etc.

→ Contains atleast one column which is unique & same is not applicable to fact table.

STAR schema

Set-up in which the fact table is placed in the center surrounded by dimension table resembling the design of a star.

SNOW FLAKE schema

A schema is referred to a snowflake when a dimension table is further extended with another dimension table to provide more attributes resembling the shape of Snowflake.

DETERMINISTIC function means the output will be always the same for a given input.

NOT DETERMINISTIC function means the output will differ depending upon the time of Execution even with the same input.

↑ Stored procedure is a way to automate repeated tasks such as creating same report for different customers.

- where, the query that needs to be executed in a Stored Procedure is copied between BEGIN and END clause
- Also, one can Enter multiple values as input (by using TEXT datatype) to run a query and retrieve an aggregated report.

One can indicate the input parameter(s) and output parameter by using IN and OUT clauses respectively

-Benefits of Stored procedure

1. Convienience
2. Security (We can given limited access to those user who can run these stored Procedures)
3. Maintainability
4. Performance (Stored Procedures are Compiled and also usually, It works faster than native queries)
5. Developer productivity

Stored procedures are best suited to automate tasks like Top N and Bottom N

Performance Optimization / Improvement (Improvem!)

which

- PERFORM ANALAYZE will gives the break down of each query / line, actual time of Execution
- currently, while performing Pre-invoice document report. the time taken by machine to run the query was nearly 11.23... sec; ~~So to~~ which Because of user-defined functions, the time taken to run the query was high to reduce the time period, we have been included a dim date table which, saves the time by ~~per~~ simply doing the mapping rather than executing the query to all records [get_fiscal_year] function.
  - ↑ Duration and Fetch are two key metrics to understand performance of a query.
  - ↑ Duration is the time taken for a query to get executed.
  - ↑ Fetch is the time taken to retrieve the data from the database server

Views :-

→ Views are virtual tables which provide you a transformed table on the fly without taking up the storage space.

→ CTE's (Common table Expressions) are like views but they are temporary & restricted to the Particular session, whereas views can be used in any session (which act as a stored in database (virtually))

→ We cannot directly add the derived columns in the base query hence CTE's sub Queries or Views are required

Window Function :-

A window function performs a calculation across a specified set of tables rows with reference to the current row.

→ over() clause is a window function which will execute the aggregation function formula across a specified set of rows.

→ If nothing is specified inside ~~rows~~ over() function, it will take all the rows as one window

row_number() rank(), dense_rank() function's will be helpful to get the ranks based on ~~their~~ specified criteria, using over() function (window function)

→ row_number() function will assign the row number for the specified window of a table.

→ rank() and dense_rank() are used to find the rank of an entity within a specified window of a table.

→ rank() will skip ranks if the ranks are same while dense_rank() does not skip any rank.

Use Cases of SQL in Industry

1. Ad Hoc Analysis
2. Report Generation
3. Exploratory Data Analysis (EDA) & Machine learning
4. Inside BI tool
5. ETL and Data Migration

CRUD
C : Create
R : Read
U : Update
D : Delete

Database Triggers :-

A trigger can be created to update table records on multiple tables when a new record is inserted in a particular table.

In real-time, new records are added to data-tables on a regular basis based on the business requirements by data Engineers.

Common use cases of triggers are:

- To create aggregated/derived data
- Create historical update logs
- Data validation

Database Events

→ one can create an Event to perform various actions and deleting session logs is one of them.

→ Create event clause is used to create an event.

→ Show events clause will display the list of events created

→ Drop events clause will delete the event

[Basically, Database Events are used to run defined schedule interval / time

Ex:- If we want to run the SQL code every day in 5AM in morning / after days / Every 1 month.

(Into but SQL code)

where SQL events comes into the play. "Database Events" which will run on a time that's been scheduled before

Advantages of using Database Events

→ Deleting old data [At a Regular Interval]
→ Database scheduled Maintenance
→ Generate Aggregate data
→ To clear logs

Difference blw temporary tables and common table Expressions (CTE's)

→ temperory table can be created as followed below

```
create temporary table table_name
then SQL code
```

→ which can exists only that particular session (which means, ~~after~~ if we close the SQL workbench, & then again if we open and tried to run the same SQL query which does not work because, the temperory table is will exists for only that session. If we want to Execute the query, then we have re-run the "Creation of temporary table syntax"

→ Incase of CTE's which can be written as followed below

```
with table_name as (
        SQL Code/query
)
```

& we can refer that table & use for our manipulations.

→ the CTE table will work only untill its scope which means we ~~should~~ can run the query's only ~~and with~~ under the CTE's table, where we cannot run/execute the query in any another SQL tab.

Benefits of CTE's

→ CTE's are reusable.

Ex:- for reusable CTE's

```
with cte1 as (select from table1),
     cte2 as (select.. from cte1),
     cte3 as (select.. from cte1)

Select... from ct3
```

→ CTE support recursion

Benefits of Sub-queries

→ Sub-queries can be used in where clause.

Ex:- select * from movies
where imdb_rating > (select avg(imdb_rating) from movies);

→ Also, can be used in SELECT clause

Ex:-

```
select
 actor_id, name,
 (select count(*) from movie_actor
  where actor_id = actors.actor_id)
  as movies_count
from actors
order by movies_count desc
```

	Subquery	CTE	Temporary table	Views
Validity	Scope of the statement	Scope of statement	Session	Forever
Readability	low	high	high	high
Ideal use case	→ In where & select clause	→ Reuse sub result → Recursive use case → At all places where we can replace subquery with CTE	→ Perform multi pass processing steps on a data set	→ Derived tables that will be used in multiple queries

Database Index :-

→ It is a way to speed up SQL queries.

→ To create a new index, one can use CREATE INDEX statement.

→ SHOW INDEXES in <table name> will display all the indexes in a given table.

→ Adding an index comes at the cost of extra memory space & slow writes. Hence add it only when necessary.

→ Most of the indexes use B Tree data structure under the hood

Composite Index :- (which is a best practice to optimise query performance tips)

In simpler words, whenever we have more than one column as an index, we call it as a Composite Index

→ Order of columns in a composite index is important & it should be decided based on our query needs.

⇒ Database Index types

1. Primary Key Index (which cannot contain Null values)
2. Unique Index (can contain Null values)
3. Regular or Normal Index (Values do not have to be unique, they can be Null as well)
4. Fulltext Index (It can be useful when we want to perform advanced search operations on text columns.)
   → MATCH and AGAINST can be used to get benefit of FULLTEXT index