Database & SQL

A database is an electronic storage in which data is organized systematically.

- The data is organized in a systematic manner known the entity, in which the data is stored as rows and columns, which are related to each other, hence called as **Relational elements.**
- Attributes and features are used to find the data from the database.
- The each row of the table represent the instance of the entity.
- The columns that specify or have the heading of the which the specific column is known as the **Fields,** and the data in a single row is known as **Records.**
- An each entity contain, unique values known as **Primary keys.** Which have an unique value that differ the data from the other data values.
- The **foreign key** is the unique ID in a table which represents or connects the primary key of the original table
- The set of legal values that are assigned to the attributes is known as the **Domain.**
- An Entity is an object with related attributes, in Relational data base system each interesting object in a project can be considered as Entity.

Types of databases are:

1. Object- oriented databases- where the data is stored in the form of objects.
2. Graph Databases - Where the data is stored in the form of graphs, nodes
3. Document Databases - This databases uses Java script, JSON to store the data into the tables known as collections.

> The NoSQl databses store data base in a structured and scalable way.

- The logical databases structure is represented using the ERD(Entity Relationship Diagram). It represents how the data is represented into the tables using DBMS(Like Mysql, or any).
- In a physical database structure the primary key of a table is used as the foreign key of the another table.

> 💡 **Big Data -** It is used to store both the structured, semi-structured and unstructured data, and is more powerful than the traditional data solving problems.

History

The history of databases has began in the **1960s**.

1970 - 1980 : - Flat flies, hierarchical and network

- The flat files models are used to store the entities of data only in a single file, in the form of text which is of a limited length.
- The hierarchical model is used in the same era of the flat files but in this model the data is arranged, and the one-to-many relation is used.
- The network are more complicated form of databased where an entity of data can have many number of owners and can be accessed through those. The language named **"SEQUEL"** is firstly used in this era to interact with the database. And later on when the relational databases are developed this language is developed into the SQL.

1980 - present :- **Relational**

- It is invented by E.F. Codd
- The data is stored in the form of tables, where the data in the tables have a column of attributes and a primary key attribute, which is the unique id of the data.
- This Database is the successor of the hierarchical and network databases.

1990 - present :- **Object-oriented, Object-relational, Web-Based**

- The object-oriented databases are used in order to save the data into the databases using the OOP like Java, C++. Where the data is represented in the form of classes and objects.
- The NoSQL Databases are now the trend, where many social media platforms and other MNCs use them to organize their data, as they are more flexible and scalable, and also faster than compared to the relational databases. And they also can process the unstructured and semi-structures data.

Structures Query Language(SQL)

CRUD - Create, read, Update, and Delete

The database interpret or get to know the instructions given my SQL using the Database Management System(DBMS).

The DBMS, takes the responsibility of transforming the instructions given via SQL into a form that is understood by the Databases

The Subsets of SQL are :

- **Data definition language** - It is used in order of the creations, altering and dropping of the data into the tables.
- **Data manipulation language -** This is used in order to manipulate the table, using the insert, update or delete any specific data or value from the database. Most of the CRUD operations comes under DML.
- **Data query language -** The DQL is used in order to read the data in the databases, the select command is used to read the data from the specified columns or row.
- **Data control Language -** The DCL commands are used in managing the access of the databases, the grant and revoke commands are used in order to give or remove the permission for others to access your database.

SQL Syntax

- **Creating a Database** :- CREATE DATABASE Database-name;
- **Creating a Table :-** CREATE TABLE table_name;
- **Add data to table :-** INSERT INTO table_name(Column_one, Column_two, Column_three,.....) VALUES(value1 , value2, value3.....)
- **Update data in the table :- UPDATE table_name** SET coloumn_name = “value” WHERE primary_key = value;
- Delete data :- DELETE from Table_name WHERE primary_key = Value;
- TRUNCATE - this command is used to remove all the data from the table but not the table.
- The maximum length of a database name while creating a database is 63 characters.
- we use the reference keyword, that is from where we are referencing the foreign key.

> 💡 If you doesn’t correctly enter the where clause while performing a DELETE query, then there is a chance that you might loose all the data that you have stored in the table.

Data types

- String - CHAR, VARCHAR
  - TINYTEXT - this datatype is used in order to store up-to 255 characters.
  - TEXT - used to define 65,000 characters.
  - MEDIUMTEXT - define columns 16.7 Million characters
  - LONGTEXT - 4GB of data
- Numeric - INT, TINYINT, BIGINT, FLOAT, REAL
  - The maximum value a TINYINT can store is 255, and the maximum number a INT can store is 4 billion.
  - and these data types can also accept positive and negative data types.

- Date and Time - DATE, TIME, DATETIME
- Binary - BINARY, VARBINARY

The mentioned below are known as Miscellaneous data types:

- Character large object(CLO) - For storing large block of text in form of text encoding.
- Binary large object(BLO) - for storing large binary objects like images.

Tables

Every table should abide by rules or constraints. these are know as the integrity constraints.

1. Key constraints - The primary key should present in every table.
2. Domain constraints - the rules of the data that has to be shared in a particular field, for example you cannot store the address of the student in a first name field.
3. referential integrity constraints - When the foreign key is used in another table, then the table from which the foreign key is referred must exist.

- The tables must have a key attribute that is used to uniquely identify an individual record of data
- The candidate key attribute is known as the unique value that is present in the each row of the table.
- Composite key attribute is known as the if the two or more attributes in a row combine to form a unique value.
- Alternate key, it is a key which also has unique value in each row but not selected as the primary key.
- The rows or records in a table are also known as tuples, and the columns are known as fields or attributes.
- The number of columns a table has is known as degree.
- The number of records or rows a table has is known as cardinality.

- If there are no unique values in the table to become a primary key, then you can create a composite primary key by adding to columns in a table.

Constraints

- NOT NULL :- this constraint is used in order to make a column field not empty. that is, when an variable is declared using this not null constraint then the table must be assigned a value, and if no value is assigned to the field, then it displays an error.

```
CREATE TABLE Employees (
    ID int NOT NULL,
    Name varchar(255) NOT NULL,
    Position varchar(255)
);
```

- DEFAULT :- This constraint is used in order to place a default value into the field if no value is assigned into it.

```
CREATE TABLE Team (
    CustomerName varchar(255) NOT NULL,
    OrderDate DATE DEFAULT "Barcelona"
);
```

Operators

Arithmetic Operators :- We use arithmetic operators in SQL to carry out mathematical operations.

- +,-,/,%,*
- using the arithmetic operators in SQL we can also performs operations on fields.

Comparison operators :- This operators are used in order to compare two values.

- ==, >,<,≤,≥

- <> this operators results the values that are not equal to the value that you give.

Clauses

There are different clauses in SQL, a few of them are:

ORDER BY -We use this clause in order to reorder the one or more columns in a table.

Syntax - SELECT Column1,Column2,Column3... FROM Table_name ORDER BY column_name ASC/DESC;

- The ordering of the data is depended on the data type of it, if the data type is numeric then the data will be organized in numerical order.
- If the data is a string based then it will be ordered in alphabetical order.
- the order by clause sorts the data in ascending order by default.

WHERE - It is used in order to specify a condition.

- In the where clause we have 3 other comparison operators, they are
  - BETWEEN - Used in order to get the results that are in between of the required numeric values or date or any.
  - LIKE - to specify a pattern which the criteria search. it checks for if the data in the columns start or end with a particular character or string using the modulus operator as the before or the after characters. 'mo%' this check if there are any strings that start with MO.
  - IN - to specify multiple values for columns. that is it used to search multiple columns with the specified values.
  - OR - it is used to combine multiple conditions .
  - UNIQUE - searches for every row in the table and results the values that are unique, that is those which aren't duplicates.

DISTINCT - This clause is used in order to return the data without any duplicates.

- The distinct clause also returns the null value records, as it considers the null to be a unique value.

- There are a few aggregate functions in sql like

Schema

- The plan that one must have before creating a database.
- It is the organizing the information and relationships in a database(General definition).
- Database Schema, it is a collection of data structures or an abstract design of how a data is store din a database.
- Irrespective of the DBMS, the schema is binded to this two main concepts, organizing of the data in the tables, and relating the tables.
- Tables, columns, keys, datatypes are all schema objects
- Schemas are easily accesible and they provide more security, we can also transfer the ownership of the schema easily.
- The process of schema designing is known as the data modeling.
- Internal or physical schema :- the physical schema describes how the data is stored in the database and how.
- The Logical data is known as the relationships of the tables in a database.

Relations in tables

There are three types of relations in tables:

1. **One-to-one** - in this type of relation the row or record in one table is connected to other row or record in another table.
2. **One-to-many** - in this type of relation the row in one table is connected to two or more rows in another table
3. **Many-to-many** - In this type of relation a single record can be connected to the two or more records in the second table, also the record in the second table can also be connected to the record in the first table with two or more records.

Attributes

There are 6 types of attributes in a Relational database system:

1. Simple attributes - it could not be classified more further
2. Single-valued attributes - The attributes in which only one value is present.
3. Composite attributes - it is divided into different components, such as a name can be split into first name and last name.
4. Multi-values attributes - the attributes that can store multiple values in them.
5. key attributes - the field that have a unique value.
6. Derived attributes - The derived attributes are those which can derived from an another attribute, such as if there is a attribute named date of birth than we can get the age of the person from that.

Normalization

The process of creating tables that minimizes the problems or challenges facing by database systems. It creates tables that are only used for single purpose.

The normalization process makes it easier for engineers to access databases.

The most common problems caused in the database systems are:

1. **Insert anomaly** - when you wanted to insert a few columns of records into the table, it results in also inserting more records into the table of the row that you are inserting in. For suppose if you insert the data of 3 columns in a 4 columned table, there comes a problem, that tells to insert the 4th column.
2. **update anomaly** - You might get any error while you update a few records, but there might be a few records that won't get updated.
3. **delete anomaly** - this anomaly is occurred when you are intended to delete a few columns of data from a record, and if the other columns that are present in the row also get deleted.

The three database normalization forms:

These form as used for building tables in such a way to optimize the database structure.

1. **First Normal form** - it eliminates unnecessary duplicates from the database or it eliminates the repeating groups of data from the database.

Data atomicity - Data atomicity means there must be only one value in the column field.

- The repeated data in the database or table might cause data redundancy and inconsistency.
- *The main objective of the 1NF is to design a table in a database such that there should be no duplicate data in the table.*

2. **Second normal form -** *If a table need to comply with the 2NF rule, then it must have at least one primary key in it, and the other non-primary key attributes must relying on the primary key field in order to retrieve the data, as there are no other unique fields in the table.*

   Functional Dependency - In a table if there is only one column field that is unique or if there is no other alternate key, then the other fields in the table are functionally dependent to the unique field.

   Partial Dependency - In a table if there are no fields that contain unique values then, we create a composite key, that is we create a primary key by combining two columns in a table, then the other fields are partially dependent on the both columns.

3. **Third normal form -** *All the attributes in the table are determined by only the primary key.*

   Transitive Dependency - In a table, the non-key attribute cannot functionally depend on the other non-key attributes.