

Database:-

A database is an organized collection of data, stored and retrieved digitally from a remote or local computer system.

Database can be vast and complex, and such databases are developed using fixed design and modeling approaches.

DBMS

DBMS stands for Database management System. DBMS is a ~~software~~ system software responsible for 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 user application software.

RDBMS:-

- RDBMS stands for Relational Database Management System.
- The key difference between here, compared to RDMS stores data in the form of collection of tables, and relations can be defined between the common fields of these tables.

→ most common modern database management system are based on MySQL, mixed of SQL Server, Oracle, IBM DB2, and Amazon Redshift are based on RDBMS.

SQL

- SQL stands for Structured Query language.
- It is a standard language for retrieving and manipulating Structured databases.
- ~~And the features~~

MySQL

It is a relational database management system, like SQL Server, Oracle or IBM, DB2, that is used to manage SQL databases.

Tables and Fields:

- A table is an organized collection of data stored in the form of rows and columns.
- Columns can be categorized as vertical and rows as horizontal.

- The columns in a table are called fields while the rows can be referred to as records.

Constraints

- Constraints are used to specify the rules concerning data in the table. It can be applied for single or multiple fields in an SQL table during the creation using the ALTER TABLE command.
- **NOT NULL** :- Restricts Null values from being inserted into a column.
- **Check** :- Verifies that cell values in a field satisfy a condition.
- **Default** :- Automatically assigns a default value if no value has been specified for the field.
- **Unique** :- Ensures unique values to be inserted into the field.
- **Index** :- Indexes a field providing faster retrieval of records.

- **Primary Key:** - Uniquely identifies each record in a table.
- **Foreign Key:** - Ensures referential integrity for a record in another table.

Primary Key:-

Primary key constraint uniquely identifies each row in a table, i.e. must contain unique values and has an implicit NOT NULL constraint.

A table in MySQL is restricted to have one and only one primary key, which is composed of single or multiple fields (columns).

Example:-

```
create table Students ( # Create table
    ID INT NOT NULL
    Name VARCHAR(255)
    # PRIMARY KEY (ID)
);
```

Table with a single field as primary key.

```
CREATE TABLE STUDENTS (
    ID INT NOT NULL,
    LastName VARCHAR(255),
    FirstName VARCHAR(255) NOT NULL,
    CONSTRAINT PN-Student
    PRIMARY KEY (ID, FirstName)
);
```

Create table with multiple fields as Primary key.

Set a column to a Primary key
After Table STUDENTS add Primary key (ID);
After Table STUDENTS add Constraint
PN-STUDENT PRIMARY KEY (ID, FirstName)

→ Unique Constraints

A unique constraint ensures so that all values in a column are different.

This provides uniqueness for the column(s) and helps identify each row uniquely.

Unique Primary Key, there can be multiple unique constraints defined per table.

The code syntax for Unique is quite similar to that of PRIMARY KEY and can be used interchangeably.

* Create table with a single field as unique */

Create table Students (

ID INT NOT NULL UNIQUE

NAME VARCHAR(255)

);

Table with multiple fields as Unique

CREATE TABLE Students (

ID INT NOT NULL

lastName VARCHAR(255)

firstName VARCHAR(255)

NOT NULL

CONSTRAINT PU-STUDENT

UNIQUE (ID, firstName)

);

Set a Column as Unique

ALTER TABLE Students ADD UNIQUE
(ID);

ALTER TABLE Students ADD CONSTRAINT
PU-STUDENT UNIQUE (ID, Student)

Foreign key

- A FOREIGN KEY comprises of single or collection of fields in a table that essentially refers to the PRIMARY KEY in another table.
- FOREIGN KEY constraint ensures referential integrity in the relation between two tables.
- The table with the foreign key constraint is labelled as the child table, and the table containing the candidate key is referred as referenced or parent table.

Example:-

```
CREATE TABLE Students (
    ID INT NOT NULL
    Name VARCHAR(255)
    LibraryID INT
    FOREIGN KEY (LIBRARY_ID)
    REFERENCES Library (LibraryID)
```

~~After~~ table

```
After table Students (
    ID INT NOT NULL PRIMARY KEY
```

NAME VARCHAR (255)

LIBRARY_ID INT FOREIGN KEY

(LIBRARY_ID) REFERENCES

library (library ID)

Join

The SQL Join Clause is used to combine records (rows) from two or more tables in a SQL database on a related column between the two.

Types of Join:-

- Inner Join :- Retrieves records that have matching values in both tables involved in the join. This is the widely used join for queries.
- Left Join :- Retrieves all the records from the left and the matched records from the right table.

- Right Outer Join :- Retrieves all the records from the right hand table and the matched records from the left.
- Full Outer Join :- Retrieves all the records when there is match in either left or right table.
- Self Join
A Self Join is a condition case of regular Join where a table is joined to itself based on some relation between its own columns.
→ Self Join uses the inner join and/or left join clause and a table alias is used to assign different names to the table within the query.

SELECT A.emp_id AS "Emp-ID", A.emp-name AS "Employee", B.emp-id AS "Sup-ID", B.emp-name AS "Supervisor"

FROM employee A, employee B
where A.emp-sup = B.emp-id;

Cross Join:-

Cross Join can be defined as a cartesian product of the two tables included in the join.

The table join after join contains the same number of rows as the cross - product of the number of rows in the two tables.

If a WHERE clause is used in cross join then the query will work like an INNER JOIN.

S

```
Select stu.name , sub.subject  
from Students AS stu  
CROSS JOIN Subjects AS sub;
```