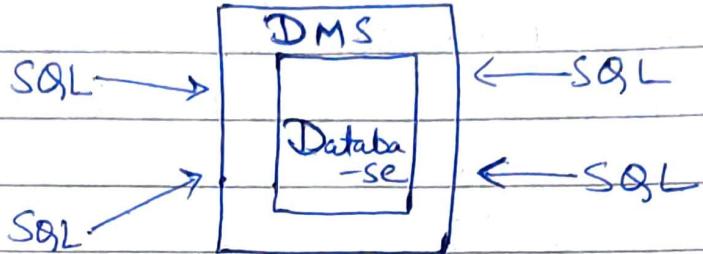


MySQL

Date _____ Page 1)

Database: A database is a collection of information (data) in an organised manner.

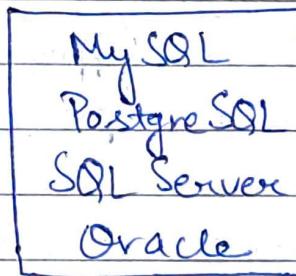
A database management system allows us to interact with data stored inside databases using SQL.



SQL stands for structured query language and is a common language using which we can communicate with databases.

→ What is MySQL?

MySQL is a type of relational database management system (RDBMS) which provides a user interface for users to interact with the database using SQL queries.



→ What is RDBMS?

A relational database is a collection of data organised into tables. Table contains columns of data categories, and rows with particular instance of that data category.

Tables within a relational database can be linked together.

- Creating a database in MySQL workbench —
 - (CREATE DATABASE (database-name));
- Ex: CREATE DATABASE localdb;
- Showing the list of all available databases present in the system
 - SHOW DATABASES;
- Deleting / Dropping a database from the workbench —
 - DROP DATABASE (database-name);
- After creation, to use a particular database —
 - USE (database-name);
- Eg: USE localdb;

Data Definition Language (DDL): DDL is used to deal with database schemas and descriptions. It also defines how should data reside inside databases.

Eg: Create, Alter, Drop, Truncate, Comment

→ Data Types in MySQL :

- * INT : Whole number {For Eg: Age → 22, 28, }.
- * FLOAT (M,D) : Decimal numbers (approx) {For Eg: Height → 1.78, 2.28 }
- * DECIMAL (M,D) : Decimal numbers (precise) {For Eg: Price → 10.25, 8.40}
- * BIGINT,

In float rounding off is done, but in decimal data type rounding off is not allowed.

(M,D) → Means M → total digits, D → decimal place

For writing 1.78 in float we declare as Float (3, 2).

- * CHAR (N) : Fixed length character
- * VARCHAR (N) : Varying length character {Eg: Names}.
- * ENUM ('M', 'F') : Value from a defined list {Gender}
- * BOOLEAN : True or False

* DATE : (YYYY-MM-DD)

* DATETIME : (YYYY-MM-DD HH-MM-SS)

* TIME : (HH-MM-SS)

* YEAR : (YYYY)

→ Primary Key : A primary key is a column or set of columns, which uniquely identifies a record within a table.

A primary key must be unique and it cannot be NULL.

A table can have atmost one primary key.

→ Foreign Key : A foreign key is used to link two tables together. A foreign key is a column whose values match with the values of another table's primary key column.

The table with the primary key is called parent/reference table and the table with the foreign key is called the child table.

A table can have multiple foreign keys.

→ Creating a table inside a database

create table table-name (

column-name1 data-type,

column-name2 data-type, ---);

→ To check if the table is created or not we can use, the command : show tables ;

→ Adding a column to already existing table - data-type;
alter table table-name add column column-name;

→ Deleting a column from the table:

alter table table-name drop column column-name;

→ Removing all the records from a table :

truncate table table-name ;

→ Delete the entire table

drop table table-name ;

→ Adding a primary key to the table when table is already created :

alter table table-name add primary key (column-name);

→ Adding a primary key to the table while creating the table :

create table table-name (

column-1 datatype primary key,
column-2 datatype);

Eg: create table employee (

id int primary key,
name varchar(20));

→ Removing the primary key from the table:

alter table table-name drop primary key;

→ Adding a foreign key to a table that is already existing and was created earlier :

alter table table-name2 add constraint constraint-name

foreign key (column-name) references table-name1 (column-name);

Eg: Suppose we have an employee table having id column as the primary key, now we want to make another table Employee-info and this table has a column named employee-id which has exact same value as in the id column of employee table. So, we can make employee-id column as the foreign key that references id column of employee table. This could be done as —

SQL query →

alter table employee-info add constraint forkey
foreign key (employee-id) references employee (id);

→ Dropping a foreign key from the table —

alter table table-name drop foreign key constraint-name;

Eg: alter table employee_info drop foreign key forkey;

→ Adding a unique constraint to a column in

alter table table-name add constraint constraint-name
unique (column-name);

Eg: alter table employee add constraint unique_id unique(id);

→ Removing the unique constraint assigned to a column:

alter table table-name drop index constraint-name;

Eg: alter table employee drop index unique_id;

→ Changing the name of a column inside a table:

alter table table-name change 'old-column-name' 'new-column-name'
datatype;

Eg: alter table employee change 'id' 'E-id' int;

→ Changing the data type of a column :

alter table table-name modify column-name datatype^{new};

Eg:

alter table employee modify house_no varchar(30);

Exercise-1

(1)

Create a database named new-database.

(2)

Inside the database create three tables named 'addresses', 'people' and 'pets'.

The schema of the table is given as follows—

Addresses

id	H-no.	city	zipCode

People

id	f-name	l-name	addId

name	species	owner-id	Pets

- (3) Add a primary key to the ^{id column} in pets and people tables.
- (4) Add a foreign key to the owner-id field in the pets table referencing the id field in the people table.
- (5) Add a column named email in the people table.
- (6) Add a unique constraint to the email column in the people table.
- (7) Rename the name column in the pets table to 'first-name'.
- (8) Change the zip code data type to char(7) in addresses table.

Ans → 1 → create database new-database ;

To use it: use new-database ;

Ans → 2 → Creating addresses table →

```
create table addresses (id int, H-no int, city varchar(20),
zip code t varchar(7));
```

Creating people table -

create table people (id int, f-name varchar(20),
l-name (20), add-id int);

Creating pets table -

create table pets (name varchar(20), species Varchar(20),
owner-id int);

Ans → 3 →

alter table people add primary key (id);

alter table pets add primary key (owner-id);

Ans → 4 → alter table pets add constraint fkeypets foreign key
(owner-id) references people (id);

Ans → 5 → alter table people add column email varchar(25);

Ans → 6 → alter table people add constraint u-email unique (email);

Ans → 7 → alter table pets 'name' 'first-name' varchar(20);

Ans → 8 → alter table addresses modify zipcode char(7);