sql commands – d/v into 5 categories

ddl – data definition language – create a table

dml - data manipulation – insert / update

tcl – transaction control language - - whenever you are committing the data or rollback

session control – within time stamp – its used in hibernate

system control –

sql commands

create database db\_name ;

i.e create database dbnew; --- for creating a database

mysql> create database dbnew;

Query OK, 1 row affected (0.08 sec)

Char is fixed length , varchar – variable length – we use var char only

Binary data can be stored in – blob

mysql> show databases; -- to see the entire database available in system

+--------------------+

| Database |

+--------------------+

| dbnew | --- I have created this today

| information\_schema |

| my\_sql\_db |

| mysql |

| performance\_schema |

| sakila |

| sys |

| world |

+--------------------+

8 rows in set (0.06 sec)

mysql> use dbnew; -- in databases now I am using dbnew

Database changed

mysql> connect dbnew; --- we can use connect dbnew also – since its provide id and current db name

Connection id: 9 --- for frontend which id I am currently using ,for knowing that purpose I am using connect command

Current database: dbnew

mysql> show tables; -- since I have not created any tables in my new database so its showing empty

Empty set (0.08 sec)

Dbms helps use to perform crud operation on the data from the databases.

Prepose of using database instead of spreadsheet

++ data is stored in structural manner

++ its secure

++ redundancy can be avoided

++ its simple to use

++ its fast

++ multiple users can access the data

Its not mandatory to write the data type in capital letter, we can have in small letter – its not case sensitive

INT, VARCHAR(I am adding size – like this much characters can be there in this )

create table persons (

personId INT,

firstName VARCHAR(40),

lastName varchar(40),

address varchar(60),

city varchar(15)

);

Query OK, 0 rows affected (0.07 sec)

mysql> show tables; --- to check in dbnew how many tables are available

+-----------------+

| Tables\_in\_dbnew |

+-----------------+

| persons |

+-----------------+

1 row in set (0.04 sec)

mysql> desc persons ; --- it provides description of the you have created

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| personId | int | YES | | NULL | |

| firstName | varchar(40) | YES | | NULL | |

| lastName | varchar(40) | YES | | NULL | |

| address | varchar(60) | YES | | NULL | |

| city | varchar(15) | YES | | NULL | |

+-----------+-------------+------+-----+---------+-------+

5 rows in set (0.00 sec)

Add a record in the table

Use insert command to add the records

Insert into table-name (column-name)

Values( value ,value );

insert into persons (personId, firstName, lastName, address, city)values

(1, 'Ramlal', 'Sharma', '123 ABC apartment', 'Delhi'),

(2, 'Shamlal', 'Varma', '426 BSDK apartment', 'Noida');

mysql> insert into persons(personId, firstName, lastName, address, city)values

-> (1,"sac","h","gfjjfdghj","dfdhf"),

-> (3,"dhddggf","fdfg","gfgfjvjveufd","fgfgfshs");

Query OK, 2 rows affected (0.00 sec)

Its better to use notepad for queries because in word “ “ 🡪 makes some problem , so notepad is best

Fetch data from table

Select \* from table\_name ;

i.e select \* from persons;

mysql> select \* from persons;

+----------+-----------+----------+--------------------+----------+

| personId | firstName | lastName | address | city |

+----------+-----------+----------+--------------------+----------+

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi |

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi |

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida |

| 1 | sac | h | gfjjfdghj | dfdhf |

| 3 | dhddggf | fdfg | gfgfjvjveufd | fgfgfshs |

| 1 | sac | h | gfjjfdghj | dfdhf |

| 2 | dhddggf | fdfg | gfgfjvjveufd | fgfgfshs |

+----------+-----------+----------+--------------------+----------+

7 rows in set (0.00 sec)

Add a column from existing table ,

i.e I missed a column , I am adding now in persons table

alter command ---- alter table table\_name add column\_name data\_type;

alter table persons add age int; --- now I am adding age to the table column

mysql> alter table persons add id int; ---- output

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+-------+------+

| personId | firstName | lastName | address | city | id |

+----------+-----------+----------+--------------------+-------+------+

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida | NULL |

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi | NULL |

By miss I mentioned column\_name as id instead of age 🡪 so I need to alter column name

Alter table table\_name column rename current\_column\_name to new\_coloumn\_name;

Alter table persons rename column id to age;

mysql> alter table persons rename column id to age ;

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+-------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+--------------------+-------+------+

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida | NULL |

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi | NULL |

+----------+-----------+----------+--------------------+-------+------+

2 rows in set (0.00 sec)

Varchar and date DATA should be specified in “ ” double inverted comma

Insert 🡪 to add the data

Update command🡪 its used to modify existing records

Update table\_name set column\_name = value;

update persons set age=25 where personId=1;

update persons set age=27 where personId=2;

mysql> update persons set age=25 where personId=1;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update persons set age=27 where personId=2;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+-------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+--------------------+-------+------+

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi | 25 |

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida | 27 |

+----------+-----------+----------+--------------------+-------+------+

2 rows in set (0.00 sec)

Now I want to add another column in the persons tables

Alter table persons set citizen varchar(50);

Alter table persons set column citizen varchar(50);

mysql> alter table persons add column citizen varchar(50);

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+-------+------+---------+

| personId | firstName | lastName | address | city | age | citizen |

+----------+-----------+----------+--------------------+-------+------+---------+

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi | 25 | NULL |

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida | 27 | NULL |

+----------+-----------+----------+--------------------+-------+------+---------+

2 rows in set (0.00 sec)

Now I will add data to citizen ,

update persons set citizen=”Indian”;

Note:

Alter is used to add the column in table and update is used add the data inside the column

mysql> update persons set citizen="indian";

Query OK, 2 rows affected (0.01 sec)

Rows matched: 2 Changed: 2 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+-------+------+---------+

| personId | firstName | lastName | address | city | age | citizen |

+----------+-----------+----------+--------------------+-------+------+---------+

| 1 | Ramlal | Sharma | 123 ABC apartment | Delhi | 25 | indian |

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida | 27 | indian |

+----------+-----------+----------+--------------------+-------+------+---------+

2 rows in set (0.00 sec)

mysql> update persons set citizen="us " where firstName="Ramlal";

Query OK, 1 row affected (0.04 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update persons set city="USA" where firstName="Ramlal";

Query OK, 1 row affected (0.04 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+-------+------+---------+

| personId | firstName | lastName | address | city | age | citizen |

+----------+-----------+----------+--------------------+-------+------+---------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 | us |

| 2 | Shamlal | Varma | 426 BSDK apartment | Noida | 27 | indian |

+----------+-----------+----------+--------------------+-------+------+---------+

2 rows in set (0.00 sec)

To update multiple details in single query like city and citizenship use comma(,)

I.e update persons set city=”germany” , citizen=”german” where personId=2;

mysql> update persons set city="Germany",citizen="german" where age=27;

Query OK, 1 row affected (0.04 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+---------+------+---------+

| personId | firstName | lastName | address | city | age | citizen |

+----------+-----------+----------+--------------------+---------+------+---------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 | us |

| 2 | Shamlal | Varma | 426 BSDK apartment | Germany | 27 | german |

+----------+-----------+----------+--------------------+---------+------+---------+

mysql> insert into persons values(3,"seeta","kapoor","345 LMN Apartments","London",30,"UK");

Query OK, 1 row affected (0.04 sec)

mysql> select \* from persons;

+----------+-----------+----------+--------------------+---------+------+---------+

| personId | firstName | lastName | address | city | age | citizen |

+----------+-----------+----------+--------------------+---------+------+---------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 | us |

| 2 | Shamlal | Varma | 426 BSDK apartment | Germany | 27 | german |

| 3 | seeta | kapoor | 345 LMN Apartments | London | 30 | UK |

+----------+-----------+----------+--------------------+---------+------+---------+

3 rows in set (0.00 sec)

Note:

Practice makes man prefect , If I am not capable then I would never worked her

To delete the database – use drop command

Drop database database\_name;

Tables:

Data is stored interms of rows or columns

Alter command – to alter or to add column inside the table

i.e alter table persons add column column\_name datatype;

show tables – to check what all tables are available in database.

Insert command to insert into persons(table\_name) values ();

To add data inside the column or make change on existing data we use update command

i.e update persons set age=56 , personId=65 where firstName=”Ramlal”;

update has been done in 2 levels , updating the existing column – alter command

To remove a column from existing table

Alter table table\_name drop column column\_name;

Alter table persons drop citizen;

mysql> alter table persons drop column citizen;

Query OK, 0 rows affected (0.11 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+---------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+--------------------+---------+------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 |

| 2 | Shamlal | Varma | 426 BSDK apartment | Germany | 27 |

| 3 | seeta | kapoor | 345 LMN Apartments | London | 30 |

+----------+-----------+----------+--------------------+---------+------+

3 rows in set (0.00 sec)

Insert multiple records at once

Insert into persons values (column\_data),

(column\_data),

(column\_data);

Insert data into specific columns

Insert into persons(personId, firstName,LastName) values

(4,”geeta”,”kumari”),

(5,”Babita”,”ji”);

mysql> insert into persons(personId,firstName,lastName) values

-> (4,"geeta","kumari"),

-> (5,"babita","ji");

Query OK, 2 rows affected (0.02 sec)

Records: 2 Duplicates: 0 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+---------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+--------------------+---------+------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 |

| 2 | Shamlal | Varma | 426 BSDK apartment | Germany | 27 |

| 3 | seeta | kapoor | 345 LMN Apartments | London | 30 |

| 4 | geeta | kumari | NULL | NULL | NULL |

| 5 | babita | ji | NULL | NULL | NULL |

+----------+-----------+----------+--------------------+---------+------+

5 rows in set (0.00 sec)

Delete record

Delete all records from a table

Delete from table\_name;

Delete specific records from a table

Delete from table\_name where condition;

i.e delete from persons where personId=6;

mysql> delete from persons where personId=6;

Query OK, 2 rows affected (0.01 sec)

mysql> select \* from persons;

+----------+-----------+----------+--------------------+---------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+--------------------+---------+------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 |

| 2 | Shamlal | Varma | 426 BSDK apartment | Germany | 27 |

| 3 | seeta | kapoor | 345 LMN Apartments | London | 30 |

| 4 | geeta | kumari | NULL | NULL | NULL |

| 5 | babita | ji | NULL | NULL | NULL |

+----------+-----------+----------+--------------------+---------+------+

5 rows in set (0.00 sec)

There are 2 ways we can delete a data

To delete entire database or table

Use drop command

Drop database dbnew;

Drop table persons;

To delete the data inside the table use delete command

Delete from persons ; -🡪 entire data inside the table is deleted

Delete from persons where personId=6; 🡺 specific data is deleted

To delete entire data inside a table we have 2 commands

Delete and truncate

Select \*from persons where personId=1;

mysql> Select \*from persons where personId=1;

+----------+-----------+----------+-------------------+------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+-------------------+------+------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 |

+----------+-----------+----------+-------------------+------+------+

1 row in set (0.00 sec)

Sql is used to store unique data

Primary key – it is used to specify the uniqueness of data in a table

++ no reptation or redundancy on that particular data

Ex—empId 🡪 is unique 🡪to maintain difference b/w the employees

mysql> Alter table persons add primary key(personId);

Query OK, 0 rows affected (0.15 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from persons;

+----------+-----------+----------+--------------------+---------+------+

| personId | firstName | lastName | address | city | age |

+----------+-----------+----------+--------------------+---------+------+

| 1 | Ramlal | Sharma | 123 ABC apartment | USA | 25 |

| 2 | Shamlal | Varma | 426 BSDK apartment | Germany | 27 |

| 3 | seeta | kapoor | 345 LMN Apartments | London | 30 |

| 4 | geeta | kumari | NULL | NULL | NULL |

| 5 | babita | ji | NULL | NULL | NULL |

+----------+-----------+----------+--------------------+---------+------+

5 rows in set (0.00 sec)

mysql> desc persons;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| personId | int | NO | PRI | NULL | |

| firstName | varchar(40) | YES | | NULL | |

| lastName | varchar(40) | YES | | NULL | |

| address | varchar(60) | YES | | NULL | |

| city | varchar(15) | YES | | NULL | |

| age | int | YES | | NULL | |

+-----------+-------------+------+-----+---------+-------+

6 rows in set (0.01 sec)

If I add some data now

mysql> insert into persons(personId,firstName,lastName) values

-> (6,"gdhghg","gjgdj"),

-> (6,"gdhghg","gjgdj");

ERROR 1062 (23000): Duplicate entry '6' for key 'persons.PRIMARY'

By using primary key we make sure we don’t allow duplicate entry;

Primary key :

1. A single or combination fields
2. Identifies records in a table uniquely
3. Cannot be null or empty

Rules:

1. Key column must be unique
2. Each table will have only one primary key
3. Can not be null or empty
4. Duplicate entries with existing primary key value not allowed
5. Recommended to use INT or BIGINT as primary key in the column

Create table employees

(

Emp\_id int not null auto\_increment,

Emp\_name varchar(30),

Emp\_designation varchar(30) ,

Primary key(Emp\_id)

);

mysql> Create table employees

-> (

-> Emp\_id int not null auto\_increment,

-> Emp\_name varchar(30),

-> Emp\_designation varchar(30) ,

-> Primary key(Emp\_id)

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> desc employees;

+-----------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+----------------+

| Emp\_id | int | NO | PRI | NULL | auto\_increment |

| Emp\_name | varchar(30) | YES | | NULL | |

| Emp\_designation | varchar(30) | YES | | NULL | |

+-----------------+-------------+------+-----+---------+----------------+

3 rows in set (0.04 sec)

Insert into employees values

(”radha”,”developer”),

(sham”,”developer”),

(“radha”,”tester”),

(“sham”,”manager”);

insert into employees(Emp\_name,Emp\_designation) values

-> ("radha","developer"),

-> ("sham","developer"),

-> ("radha","tester"),

-> ("sham","manager");

Query OK, 4 rows affected (0.00 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> select \* from employees;

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 1 | radha | developer |

| 2 | sham | developer |

| 3 | radha | developer |

| 4 | sham | developer |

| 5 | radha | tester |

| 6 | sham | manager |

+--------+----------+-----------------+

mysql> delete from employees where Emp\_id=1 ;

Query OK, 1 row affected (0.05 sec)

mysql> select \* from employees;

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 2 | sham | developer |

| 3 | radha | developer |

| 4 | sham | developer |

| 5 | radha | tester |

| 6 | sham | manager |

+--------+----------+-----------------+

5 rows in set (0.00 sec)

mysql> delete from employees where Emp\_id=2;

Query OK, 1 row affected (0.01 sec)

mysql> select \* from employees;

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 3 | radha | developer |

| 4 | sham | developer |

| 5 | radha | tester |

| 6 | sham | manager |

+--------+----------+-----------------+

4 rows in set (0.00 sec)

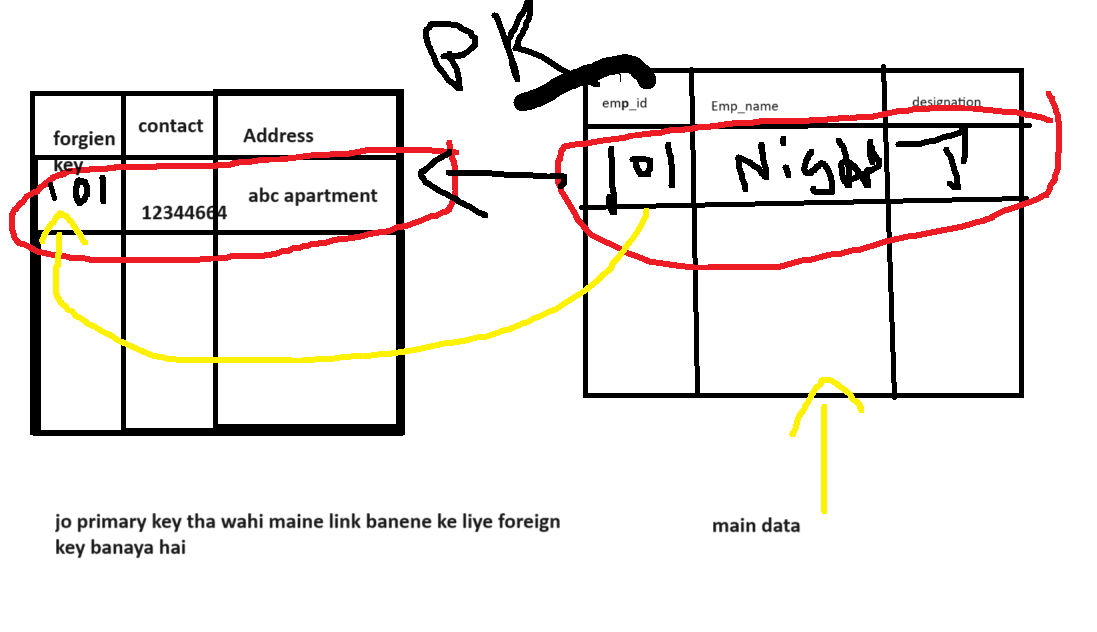
Emp\_id , mobile\_no 🡪 this are the places where data takes unique identification

Foreign key:

It’s a key that is used to link one or more tables together.

Ex – we have employee details in 2 catagories i.e one is personal details and professional details (official info – grade , skills , joining date ,designation)

Ex-2 hospital management system – patient data personal info , medical info(like what was the problem they were facing , medication , treatment , test details , medical history.. etc )



To update the data in employees

Update employees set Emp\_designation=”senior developer” where Emp\_designation=”developer”

mysql> Update employees set Emp\_designation='senior developer' where Emp\_designation='developer';

Query OK, 2 rows affected (0.01 sec)

Rows matched: 2 Changed: 2 Warnings: 0

mysql> select \* from employees;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 3 | radha | senior developer |

| 4 | sham | senior developer |

| 5 | radha | tester |

| 6 | sham | manager |

+--------+----------+------------------+

4 rows in set (0.00 sec)

mysql> Update employees set Emp\_designation="developer" where Emp\_id=3;

Query OK, 1 row affected (0.02 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from employees;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 3 | radha | developer |

| 4 | sham | senior developer |

| 5 | radha | tester |

| 6 | sham | manager |

+--------+----------+------------------+

4 rows in set (0.00 sec)

Foreign key—its used to link one or more tables together

++ Its also called as referencing key

Foreign key creates a parent child relationship in 2 tables.

++ the parent table holds the actual or initial column value and

++ column values of child table will refer the parent table key

When can we create foreign key

1. When we are creating a table – create table
2. When we are altering a table – alter table

[constraint constraint\_name]

Foreign key[foreign key \_name] (column\_name, ….)

References parent\_table\_name(col\_name,…)

Or

Foreign\_key (foreign key\_name) References parent\_table\_name

//// parent table

Create table employees

(

Emp\_id int not null auto\_increment,

Emp\_name varchar(30),

Emp\_designation varchar(30) ,

Primary key(Emp\_id)

);

//// child table

Create table emp\_info(

Emp\_id\_fk int,

Emp\_address varchar(40),

Emp\_contact varchar(40),

Constraint fk\_employees foreign key (Emp\_id\_fk)

References employees(Emp\_id)

);

constraint constraint\_name(fk\_employees) then I am going to specify which key I am using , I am using foreign key(name of foreign key)

her constraint name is going to specify that I am going to create a foreign key for this particular table

current table which foreign key will be Emp\_id \_fk

references --- which option or which table you are referring through this foreign key

mysql> desc employees;

+-----------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+----------------+

| Emp\_id | int | NO | PRI | NULL | auto\_increment |

| Emp\_name | varchar(30) | YES | | NULL | |

| Emp\_designation | varchar(30) | YES | | NULL | |

+-----------------+-------------+------+-----+---------+----------------+

3 rows in set (0.02 sec)

mysql> desc emp\_info

-> ;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| Emp\_id\_fk | int | YES | MUL | NULL | |

| Emp\_address | varchar(40) | YES | | NULL | |

| Emp\_contact | varchar(40) | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

Parent\_table (no null value and no multiple entries)

When I check the desc emp\_info ; --- > its shows MUL 🡪 there can be multiple entries of same emp\_id in this particular table(child table)

Insert into emp\_info values

(3,”abc apartment”,”12344552323”);

Insert into emp\_info values

(3,”pqr apartment”,”12344552323”);

mysql> insert into emp\_info values

-> (3,"ggj apart","43874987979"),

-> (3,"jjb apart","43874987979");

Query OK, 2 rows affected (0.07 sec)

Records: 2 Duplicates: 0 Warnings: 0

mysql> select \* from emp\_info;

+-----------+-------------+-------------+

| Emp\_id\_fk | Emp\_address | Emp\_contact |

+-----------+-------------+-------------+

| 3 | ggj apart | 43874987979 |

| 3 | jjb apart | 43874987979 |

+-----------+-------------+-------------+

Foreign key duplicates are allowed as you can see in the above output

mysql> insert into emp\_info values

-> (98,"ggj apart","43874987979");

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`dbnew`.`emp\_info`, CONSTRAINT `fk\_employees` FOREIGN KEY (`Emp\_id\_fk`) REFERENCES `employees` (`Emp\_id`))

Create a table customers with primary key as user\_id ,this holds the information for all users of amazon,

Create another table as order\_info which refers to primary key of customers

Create table customers

(

User\_id int not null ,

User\_name varchar(40),

User\_email varchar(50),

Primary key(User\_id)

);

Insert into customers values

(101,”jack”,”[jack@gmail.com](mailto:jack@gmail.com)”),

(102,”jaya”,”[jaya@gmail.com](mailto:jaya@gmail.com)”),

(103,”diya”,”[diya@gmail.com](mailto:diya@gmail.com)”);

Create table order\_info

(  
order\_id int not null auto\_increment primary key,

Amount decimal(8,2),

User\_id int,

Constraint fk\_customers foreign key (User\_id)

References customers (User\_id)

);

Or

Foreign key User\_id references customers(User\_id)

Insert into order\_info (Amount , User\_id)values

(11183.45,101),

(4347.64,101),

(98398.74,102),

(57457.44,103);

mysql> Create table customers

-> (

-> User\_id int not null ,

-> User\_name varchar(40),

-> User\_email varchar(50),

-> Primary key(User\_id)

-> );

Query OK, 0 rows affected (0.02 sec)

mysql> insert into customers values

-> (101,"jack","jack@gmail.com"),

-> (102,"jay","jay@gmail.com"),

-> (103,"diya","diya@gmail.com");

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> Create table order\_info

-> (

-> order\_id int not null auto\_increment primary key,

-> Amount decimal(8,2),

-> User\_id int,

-> Constraint fk\_customers foreign key (User\_id)

-> References customers (User\_id)

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> Insert into order\_info (Amount , User\_id)values

-> (11183.45,101),

-> (4347.64,101),

-> (98398.74,102),

-> (57457.44,103);

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> desc customers;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| User\_id | int | NO | PRI | NULL | |

| User\_name | varchar(40) | YES | | NULL | |

| User\_email | varchar(50) | YES | | NULL | |

+------------+-------------+------+-----+---------+-------+

3 rows in set (0.01 sec)

mysql> desc order\_info;

+----------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------+--------------+------+-----+---------+----------------+

| order\_id | int | NO | PRI | NULL | auto\_increment |

| Amount | decimal(8,2) | YES | | NULL | |

| User\_id | int | YES | MUL | NULL | |

+----------+--------------+------+-----+---------+----------------+

3 rows in set (0.01 sec)

mysql> select \* from customers;

+---------+-----------+----------------+

| User\_id | User\_name | User\_email |

+---------+-----------+----------------+

| 101 | jack | jack@gmail.com |

| 102 | jay | jay@gmail.com |

| 103 | diya | diya@gmail.com |

+---------+-----------+----------------+

3 rows in set (0.00 sec)

mysql> select \* from order\_info;

+----------+----------+---------+

| order\_id | Amount | User\_id |

+----------+----------+---------+

| 1 | 11183.45 | 101 |

| 2 | 4347.64 | 101 |

| 3 | 98398.74 | 102 |

| 4 | 57457.44 | 103 |

+----------+----------+---------+

4 rows in set (0.00 sec)

Certain Clauses:

1. Where 🡪 can be used with select ,insert ,update and delete commands

++specifies the condition that must be fulfilled for selected record

Ex- select \* from employees; 🡪 it will entire information

mysql> select \* from employees;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 3 | radha | developer |

| 4 | sham | senior developer |

| 5 | radha | tester |

| 6 | sham | manager |

+--------+----------+------------------+

4 rows in set (0.02 sec)

If I want to fetch specific data

Select \* from employees where Emp\_id=3;

mysql> Select \* from employees where Emp\_id=3;

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 3 | radha | developer |

+--------+----------+-----------------+

1 row in set (0.00 sec)

I will add one more info in the table

Insert into employees(Emp\_name,Emp\_designation) values

(“Nisha”,”manager”);

mysql> Insert into employees(Emp\_name,Emp\_designation) values

-> ("Nisha","manager");

Query OK, 1 row affected (0.01 sec)

mysql> Select \* from employees ;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 3 | radha | developer |

| 4 | sham | senior developer |

| 5 | radha | tester |

| 6 | sham | manager |

| 7 | Nisha | manager |

+--------+----------+------------------+

5 rows in set (0.00 sec)

Select \* from employees where Emp\_designation=”manager”;

mysql> Select \* from employees where Emp\_designation="manager";

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 6 | sham | manager |

| 7 | Nisha | manager |

+--------+----------+-----------------+

2 rows in set (0.01 sec)

mysql> Select \* from employees where Emp\_name="radha";

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 3 | radha | developer |

| 5 | radha | tester |

+--------+----------+-----------------+

2 rows in set (0.00 sec)

Now I need to write a query in such a way that only one data I need to get

So I will use where & AND – operator 🡪 once both the conditions matches then only I can get the output.

AND OPERATOR for combining multiple conditions

Select \* from employees where Emp\_name=”radha” and Emp\_designation=”tester”;

mysql> Select \* from employees where Emp\_name='radha' and Emp\_designation="tester";

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 5 | radha | tester |

+--------+----------+-----------------+

1 row in set (0.00 sec)

OR operator

Select \* from employees where Emp\_name=”radha” or Emp\_designation=”tester”;

mysql> Select \* from employees where Emp\_name="radha" or Emp\_designation="tester";

+--------+----------+-----------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+-----------------+

| 3 | radha | developer |

| 5 | radha | tester |

+--------+----------+-----------------+

2 rows in set (0.01 sec)

Order by clause 🡪

While we are retrieving the data using select query it used to

If we want the records to sort in ascending or descending 🡪 we use order by

Select expressions

From table\_name

Where condition

Order by expression

mysql> select \* from employees order by Emp\_name asc;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 7 | Nisha | manager |

| 3 | radha | developer |

| 5 | radha | tester |

| 4 | sham | senior developer |

| 6 | sham | manager |

+--------+----------+------------------+

5 rows in set (0.01 sec)

mysql> select \* from employees order by Emp\_name ;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 7 | Nisha | manager |

| 3 | radha | developer |

| 5 | radha | tester |

| 4 | sham | senior developer |

| 6 | sham | manager |

+--------+----------+------------------+

5 rows in set (0.00 sec)

mysql> select \* from employees order by Emp\_name desc;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 4 | sham | senior developer |

| 6 | sham | manager |

| 3 | radha | developer |

| 5 | radha | tester |

| 7 | Nisha | manager |

+--------+----------+------------------+

5 rows in set (0.00 sec)

mysql> select \* from employees order by Emp\_id desc;

+--------+----------+------------------+

| Emp\_id | Emp\_name | Emp\_designation |

+--------+----------+------------------+

| 7 | Nisha | manager |

| 6 | sham | manager |

| 5 | radha | tester |

| 4 | sham | senior developer |

| 3 | radha | developer |

+--------+----------+------------------+

5 rows in set (0.01 sec)

Order does not reflect original data in the table

Desc—to get data in descending

Asc – to get data in ascending

Group by clause

Whenever you want to collect data from multiple records and group the result by one or more column

Generally used with select query

Along with group by we can use aggregate function

Select \* from table where

Condition(s)

Group by expression1,expression2,…..;

aggregate function 🡪 count,sum,min,max,avg

select emp\_designation , count(\*)

from employees group by emp\_designation;

mysql> select emp\_designation , count(\*)

-> from employees group by emp\_designation;

+------------------+----------+

| emp\_designation | count(\*) |

+------------------+----------+

| developer | 1 |

| senior developer | 1 |

| tester | 1 |

| manager | 2 |

+------------------+----------+

4 rows in set (0.01 sec)

Like i am counting no of employees in there respective designation

select emp\_designation , count(emp\_designation)

from employees group by emp\_designation;

mysql> select emp\_designation , count(emp\_designation)

-> from employees group by emp\_designation;

+------------------+------------------------+

| emp\_designation | count(emp\_designation) |

+------------------+------------------------+

| developer | 1 |

| senior developer | 1 |

| tester | 1 |

| manager | 2 |

+------------------+------------------------+

4 rows in set (0.00 sec)

Select Emp\_name ,count(Emp\_name) from employees group by Emp\_name;

How many employee name have same name – I am counting that

mysql> Select Emp\_name ,count(Emp\_name) from employees group by Emp\_name;

+----------+-----------------+

| Emp\_name | count(Emp\_name) |

+----------+-----------------+

| radha | 2 |

| sham | 2 |

| Nisha | 1 |

+----------+-----------------+

3 rows in set (0.00 sec)

Having clause

* It is always used with group by clause
* It always return the records where the condition is true

Select \* from table where

Condition(s)

Group by expression1, expression2,…

Having condition;

Select Emp\_name, count(Emp\_name) from employees

Group by emp\_name having emp\_name=”Nisha”; -- . group by and having is used together

mysql> Select Emp\_name, count(Emp\_name) from employees

-> Group by emp\_name having emp\_name='radha';

+----------+-----------------+

| Emp\_name | count(Emp\_name) |

+----------+-----------------+

| radha | 2 |

+----------+-----------------+

1 row in set (0.00 sec)

As you can see it is providing information of specific employee name

mysql> Select Emp\_name, count(Emp\_name) from employees

-> where Emp\_designation="manager" Group by emp\_name having emp\_name='sham';

+----------+-----------------+

| Emp\_name | count(Emp\_name) |

+----------+-----------------+

| sham | 1 |

+----------+-----------------+

1 row in set (0.00 sec)

mysql> Select Emp\_name, count(Emp\_name) from employees

-> where Emp\_designation="manager" ;

ERROR 1140 (42000): In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'dbnew.employees.Emp\_name'; this is incompatible with sql\_mode=only\_full\_group\_by

Joins:

Joins are used in select query

++ now why do we need joins

++ purpose of using joins

* It is used to fetch or retrieve the data from multiple tables

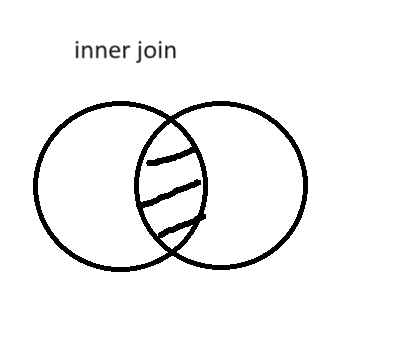
Joins are used to fetch the data from 2 or more tables (combining the data from multiple tables while retrieving the data).

Types of joins

1. Inner join (simple join)
2. Left outer join (left join)
3. Right outer join (right join)

Inner join (simple join)🡪

Used to return all rows from multiple tables where join condition is satisfied .



It will select all the data from both the tables and returns only matched data

Syntax:

Select columns

From table1

Inner join table2

On table1.column = table2.column;

Ex:

Select employees.Emp\_id,employees.Emp\_name,emp\_info.Emp\_address

From employees

Inner join emp\_info

On employees.Emp\_id = emp\_info.Emp\_id\_fk;

mysql> Select employees.Emp\_id,employees.Emp\_name,emp\_info.Emp\_address

-> From employees

-> Inner join emp\_info

-> On employees.Emp\_id = emp\_info.Emp\_id\_fk;

+--------+----------+-------------+

| Emp\_id | Emp\_name | Emp\_address |

+--------+----------+-------------+

| 3 | radha | ggj apart |

| 3 | radha | jjb apart |

+--------+----------+-------------+

2 rows in set (0.00 sec)

Primary -- > not null + unique

mysql> Select employees.\*,emp\_info.\*

-> From employees

-> Inner join emp\_info

-> On employees.Emp\_id = emp\_info.Emp\_id\_fk;

+--------+----------+-----------------+-----------+-------------+-------------+

| Emp\_id | Emp\_name | Emp\_designation | Emp\_id\_fk | Emp\_address | Emp\_contact |

+--------+----------+-----------------+-----------+-------------+-------------+

| 3 | radha | developer | 3 | ggj apart | 43874987979 |

| 3 | radha | developer | 3 | jjb apart | 43874987979 |

+--------+----------+-----------------+-----------+-------------+-------------+

2 rows in set (0.00 sec)

Left join🡪

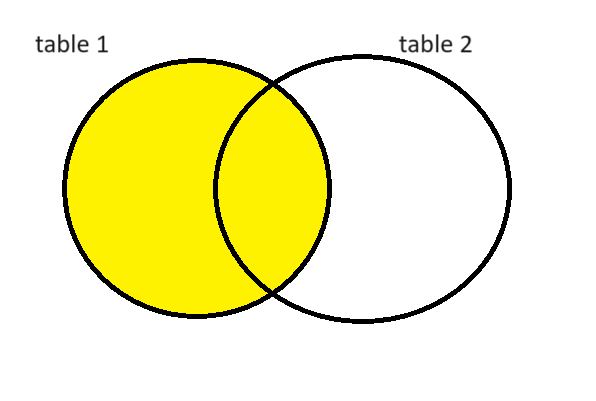
It is going to select all the data from table A and matching data from table B

Or

Used to fetch records from multiple tables

It retrieves all the records from 1st table even no matching records found in second table

If no matches are found from right side table, then returns null



Syntax:

Select columns

From table1

Left join table2

On condition.

Select employees.Emp\_id,employees.Emp\_name

From employees

Left join emp\_info

on employees.Emp\_id=emp\_info.Emp\_id\_fk;

mysql> Select employees.Emp\_id,employees.Emp\_name

-> From employees

-> Left join emp\_info

-> on employees.Emp\_id=emp\_info.Emp\_id\_fk;

+--------+----------+

| Emp\_id | Emp\_name |

+--------+----------+

| 3 | radha |

| 3 | radha |

| 4 | sham |

| 5 | radha |

| 6 | sham |

| 7 | Nisha |

+--------+----------+

6 rows in set (0.00 sec)

mysql> Select employees.Emp\_id,employees.Emp\_name,emp\_info.Emp\_contact

From employees

Left join emp\_info

on employees.Emp\_id=emp\_info.Emp\_id\_fk;

mysql> Select employees.Emp\_id,employees.Emp\_name,emp\_info.Emp\_contact

-> From employees

-> Left join emp\_info

-> on employees.Emp\_id=emp\_info.Emp\_id\_fk;

+--------+----------+-------------+

| Emp\_id | Emp\_name | Emp\_contact |

+--------+----------+-------------+

| 3 | radha | 43874987979 |

| 3 | radha | 43874987979 |

| 4 | sham | NULL |

| 5 | radha | NULL |

| 6 | sham | NULL |

| 7 | Nisha | NULL |

+--------+----------+-------------+

6 rows in set (0.01 sec)

As you can see above left join is going to select all the data from left table and matched data from right table like (contact)

Right join 🡪

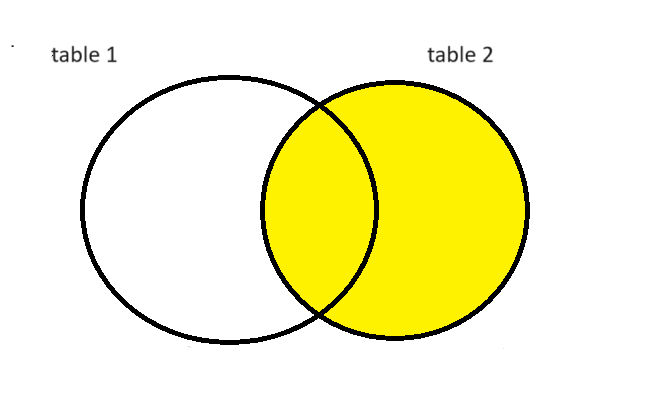
It is going to select all the data from the right table along with the matching data with the left table

Or

Used to retrieve data from multiple tables

Returns all records from right table and those from left table where condition is satisfied

If unmatched records in left table it returns null



Select columns

From table 1

right join table2

on conditions;

select employees.Emp\_id,employees.Emp\_name,emp\_info.Emp\_contact

from employees

right join emp\_info

on employees.emp\_id=emp\_info.emp\_id\_fk;

mysql> select employees.Emp\_id,employees.Emp\_name,emp\_info.Emp\_contact

-> from employees

-> right join emp\_info

-> on employees.emp\_id=emp\_info.emp\_id\_fk;

+--------+----------+-------------+

| Emp\_id | Emp\_name | Emp\_contact |

+--------+----------+-------------+

| 3 | radha | 43874987979 |

| 3 | radha | 43874987979 |

+--------+----------+-------------+

2 rows in set (0.00 sec)