MY SQL PROJECT

BY

PRIYADHARSHINI

WHAT IS MYSQL ?

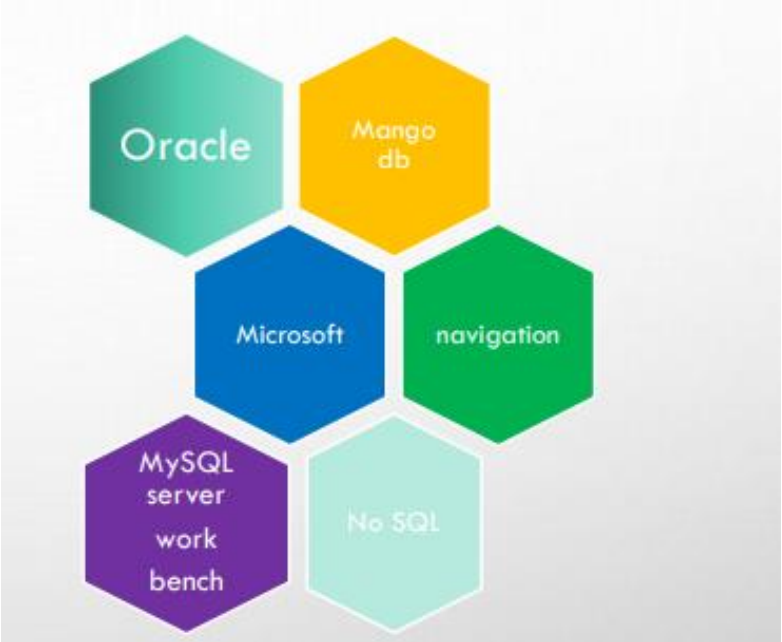
- ✓ MY SQL is the most popular open source relational database management system.
- ✓ MY SQL used for developing various web based software development.
- ✓ Developed by company MYSQL AB. (Based on CN C++).

MYSQL Workbench

- ✓ It is a visual database design tool that integrates are
- ✓ SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MYSQL database system.
- ✓ MYSQL workbench 8.0. CE version.

MYSQL SERVERS

- ✓ Oracle
- ✓ Mango DB
- ✓ Microsoft server
- ✓ No SQL



DIFFERENCE BETWEEN RDBMS AND DBMS

RDBMS	DBMS
Data stored is in table format	Data stored is in the file format
Multiple data elements are accessible together	Individual access of data elements
Data in the form of a table are linked together	No connection between data
Normalisation is not achievable	There is normalisation
Support distributed database	No support for distributed database
Data is stored in a large amount	Data stored is a small quantity

DBMS KEYS


Super Key
Primary Key
Candidate Key
Foreign Key
Alternate Key

PRIMARY KEY

A primary key is a special attribute within a table that uniquely identifies each row or record in that table.

CANDIDATE KEY

It is a super key no repeated data is called a candidate key.

SUPER KEY

A Single key or a group of multiple keys that can uniquely identify tuples in table.

ALTERNATE KEY

All the key which are not primary keys are called alternate keys.

FOREIGN KEY

It is a key it acts as a primary key in one table and if acts as secondary key in a another table.

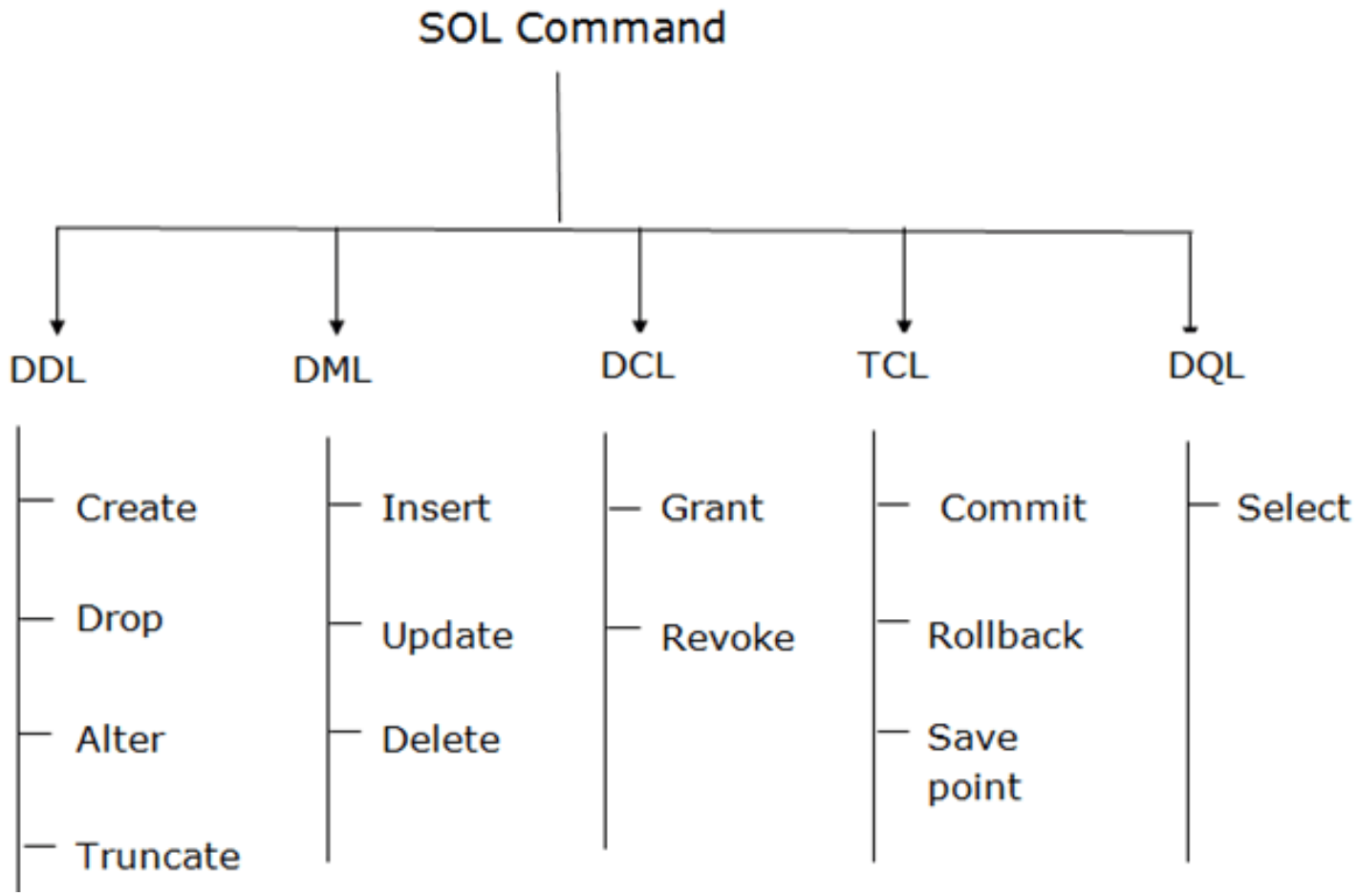
DATABASE

- A database is an organized collection of structured information, or data, typically stored electronically in a computer system
- A database is usually controlled by a database management system (DBMS).
- The data can then be easily accessed, managed, modified, updated, controlled, and organized. ▪ most databases use structured query language (SQL) for writing and querying data.

CONTENT

- ✓ MAIN COMMANDS
- ✓ GENERAL FUNCTIONS
- ✓ CALCULATE FUNCTION
- ✓ STRING FUNCTION
- ✓ DATE FUNCTION
- ✓ LOGICAL FUUNCTION
- ✓ RDBMS SYSTEM
- ✓ OIN QUERIES

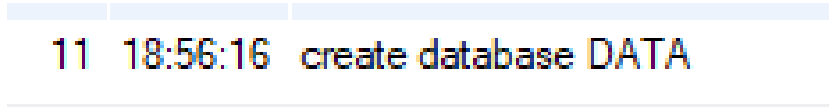
MY SQL MAIN COMMANDS



CREATE,USE,SHOW,DROP

QUERY:

- ✓ CREATE DATABASE TABLE;



- ✓ USE DATABASE ;

- ✓ SHOW DATABASES;

- ✓ DROP DATABASE TABLE;

TABLE INSERT VALUES

QUERY:

```
insert into salary_ det values
(18001,   '17001'   ,'2022-06-10'   ,241,   35000),
(18002,   '17002','2022-06-12',   241,   14000),
(18003   ,'17003',   '2022-06-15'   ,241   ,28000),
(18004,   '17004',   '2022-06-20'   ,242   ,18000),
(18005   ,'17005',   '2022-06-23'   ,241   ,30000),
```

ALTER- ADD,MODIFY,UPDATE,RENAME,DROP

QUERY:

- alter table employee modify varchar(35) ;
- alter table employee drop gender ;
- update employee set age = 32 where employee_ id = 17004;
- alter table Employee rename to Emp_ info;
- alter table Employee add gender varchar(8);

TABLE 2

- create table salary _ det(salary _ id int, emp _ id int, salary _ date branch _ id int, amount int, primary key(salary _ id));
- select*from salary _ det;

salary_id	emp_id	salary_date	branch_id	amount
18001	17001	2022-06-10	241	35000
18002	17002	2022-06-12	241	14000
18003	17003	2022-06-15	241	28000
18004	17004	2022-06-20	242	18000
18005	17005	2022-06-23	241	30000
18006	17006	2022-07-06	241	23000
18007	17007	2022-07-07	243	28000

TABLE CREATION

create table employee (employee _ id int ,employee _ name varchar(20),designation _ id int, dep _ no int, date _ of_ join date, primary key (employee _id)); select * from employee;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05

TABLE 3 & TABLE 4

create table salary (Designation _ ID int, Designation varchar(30));

Select*from salary;

Designation_ID	Designation
3001	Manager
3002	Junior Associates
3003	Senior Manager
3004	HR
3005	General Manager
3006	Team Lead
3007	Senior HR

create table dep _ det (Dep _ NO int, Dep _ name varchar(35),Branch _ ID int, Branch _ Name varchar(25));

Dep_NO	Dep_name	Branch_ID	Branch_Name
50	Production Department	241	Annan Nagar
60	HR Department	242	Velachery
70	Sales Department	243	Guindy
80	Finance Department	244	KMC

MYSQL GENERAL FUNCTIONS

- Where
- Or
- And
- In
- Not in
- >
- <
- <=
- >=
- !=

- Count
- Distinct
- Count with distinct
- Order by ascending
- Order by desc
- Group by
- Limit
- Like(_%)
- Not like
- Between

WHERE

- select*from employee where dep _ no=70;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17007	Jaga	3003	70	2022-06-06
17010	Kabilan	3006	70	2022-06-09
17011	Manasi	3001	70	2022-06-10
17017	Swetha	3002	70	2022-06-16
17018	Selvi	3002	70	2022-06-17
17019	Pooja	3002	70	2022-06-18
17020	Lakshmi	3002	70	2022-06-19

OR

- select*from employee where designation _ id=3002 or designation _ id=3005;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17002	Guru	3002	50	2022-05-12
17005	Moorthy	3005	50	2022-05-23
17009	Arthi	3005	50	2022-06-08
17012	Suja	3002	50	2022-06-11
17015	Sindhu	3005	80	2022-06-14
17016	Madhavi	3002	50	2022-06-15
17017	Swetha	3002	70	2022-06-16

AND

- select*from employee where designation_ id=3002 and dep _ no=50;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17002	Guru	3002	50	2022-05-12
17012	Suja	3002	50	2022-06-11
17016	Madhavi	3002	50	2022-06-15
NULL	NULL	NULL	NULL	NULL

IN

- select*from employee where dep _ no in(50,60,80);

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05
17008	Pavithra	3007	60	2022-06-07

NOT IN

- select*from employee where dep _ no not in(50,60,80);

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17007	Jaga	3003	70	2022-06-06
17010	Kabilan	3006	70	2022-06-09
17011	Manasi	3001	70	2022-06-10
17017	Swetha	3002	70	2022-06-16
17018	Selvi	3002	70	2022-06-17
17019	Pooja	3002	70	2022-06-18
17020	Lakshmi	3002	70	2022-06-19

>

- select*from employee where dep _ no>60;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17007	Jaga	3003	70	2022-06-06
17010	Kabilan	3006	70	2022-06-09
17011	Manasi	3001	70	2022-06-10
17015	Sindhu	3005	80	2022-06-14
17017	Swetha	3002	70	2022-06-16
17018	Selvi	3002	70	2022-06-17
17019	Pooja	3002	70	2022-06-18

<

- select*from employee where dep _ no<70;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05
17009	Arthi	3005	50	2022-06-08
17012	Suja	3002	50	2022-06-11

>=

- select*from employee where dep _ no>=50;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05
17007	Jaga	3003	70	2022-06-06

<=

- select*from employee where dep _ no<=80;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05
17007	Jaga	3003	70	2022-06-06

!=

QUERY: Select*from employee where dep _ no!=60;

OUTPUT:

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05
17007	Jaga	3003	70	2022-06-06
17009	Arthi	3005	50	2022-06-08

COUNT

- select count(emp _ id)as count from salary _ det ;

	COUNT
▶	18

COUNT WITH DIST

- select count(distinct designation _ id) as unique _ designation _ id _ count from salary _ det;

	unique_designation_id_count
▶	18

ORDER BY ASC

```
select *from department
order by dep _ name
ascending;
```

Dep_NO	Dep_name	Branch_ID	Branch_Name
80	Finance Department	244	KMC
80	Finance Department	244	KMC
60	HR Department	242	Velachery
60	HR Department	242	Velachery
50	Production Department	241	Anna Nagar
50	Production Department	241	Anna Nagar
70	Sales Department	243	Guindy
70	Sales Department	243	Guindy

ORDER BY DESC

```
select *from department
order      by dep _ name  desc;
```

Dep_NO	Dep_name	Branch_ID	Branch_Name
70	Sales Department	243	Guindy
70	Sales Department	243	Guindy
50	Production Department	241	Anna Nagar
50	Production Department	241	Anna Nagar
60	HR Department	242	Velachery
60	HR Department	242	Velachery
80	Finance Department	244	KMC
80	Finance Department	244	KMC

GROUP BY

- select dep _ no, count(emp _ id) from salary _ det group by dep _ no;

	dep_no	count(emp_id)
▶	241	8
	242	4
	243	5
	244	1

LIMIT

- select* from salary _ det limit 11,5;

	emp_id	designation_id	date_of_join	dep_no	amount
▶	18012	17012	2022-07-12	241	14000
	18013	17013	2022-07-13	242	28000
	18014	17014	2022-07-14	242	18000
	18015	17015	2022-07-15	244	30000
	18016	17016	2022-07-16	241	14000

LIKE

- select *from employee where employee_ name like 'a%';

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17006	Amutha	3006	50	2022-06-05
17009	Arthi	3005	50	2022-06-08
17013	Arun	3003	60	2022-06-12
NULL	NULL	NULL	NULL	NULL

NOT LIKE

- select *from employee where employee_ name not like 'v%';

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05

BETWEEN

- select*from salary _ det where emp _ id between 18005 and 18014;

emp_id	designation_id	date_of_join	dep_no	amount
18005	17005	2022-06-23	241	30000
18006	17006	2022-07-06	241	23000
18007	17007	2022-07-07	243	28000
18008	17008	2022-07-08	242	18000
18009	17009	2022-07-09	241	30000
18010	17010	2022-07-10	243	23000
18011	17011	2022-07-11	243	35000
18012	17012	2022-07-12	241	14000

MYSQL CACULATE FUNCTION

- SUM
- AVERAGE
- MIN
- MAX
- COUNT

SUM,AVG

- QUERY:

```
select sum(dep _ no)from  salary _ det;
select  avg (dep _ no)from  salary _ det;
```

	sum(dep_no)
▶	4355

	avg(dep_no)
▶	241.9444

MIN,MAX,COUNT

- QUERY: select min(dep _ no)from salary _ det;
- select max(dep _ no)from salary _ det;
- select count(dep _ no)from salary _ det;

- OUTPUT:

avg(dep_no)
241.9444

count(dep_no)
18

max(dep_no)
244

MYSQL STRING FUNCTIONS

- ✓ LOWER CASE
- ✓ UPPER CASE
- ✓ LEFT
- ✓ RIGHT
- ✓ CONCAT
- ✓ TRIM
- ✓ MID
- ✓ CHAR-LENGTH
- ✓ LENGTH

LCASE $ UCASE

- ✓ select*, u case (employee _ name) from employee;
- ✓ select*, l case (employee _ name) from employee;

employee_ID	employee_Name	designation_id	dep_no	date_of_join	ucase(employee_name)
17001	Geetha	3001	50	2022-05-10	GEETHA
17002	Guru	3002	50	2022-05-12	GURU
17003	Gokul	3003	50	2022-05-15	GOKUL
17004	Mani	3004	60	2022-05-20	MANI
17005	Moorthy	3005	50	2022-05-23	MOORTHY
17006	Amutha	3006	50	2022-06-05	AMUTHA

employee_ID	employee_Name	designation_id	dep_no	date_of_join	lcase(employee_name)
17001	Geetha	3001	50	2022-05-10	geetha
17002	Guru	3002	50	2022-05-12	guru
17003	Gokul	3003	50	2022-05-15	gokul
17004	Mani	3004	60	2022-05-20	mani
17005	Moorthy	3005	50	2022-05-23	moorthy
17006	Amutha	3006	50	2022-06-05	amutha

LEFT $ RIGHT

- ✓ select left(employee_name,5)from employee;
- ✓ select right(employee_name,5)from employee;

- ✓ OUTPUT:

left(employee_name,5)
Geeth
Guru
Gokul
Mani
Moort
Amuth
Jaga
Pavit
Arthi

right(employee_name,5)
eetha
Guru
Gokul
Mani
orthy
mutha
Jaga
ithra
Arthi

CONCAT $ TRIM

- ✓ select concatenate (employee _ name ," ", dep_ no) from employee;
- ✓ select trim employee_ name from employee;

	concat(employee_name," ",dep_no)
▶	Geetha 50
	Guru 50
	Gokul 50
	Mani 60
	Moorthy 50
	Amutha 50
	Jaga 70
	Pavithra 60

	trim(employee_name)
▶	Geetha
	Guru
	Gokul
	Mani
	Moorthy
	Amutha
	Jaga
	Pavithra
	Arthi

CHAR_LEN,LENGTH,MID

- ✓ select char _ length (employee _ name) from employee;
- ✓ select length(employee _name)as count _ length from employee;
- ✓ select employee_ name, mid(employee_name,3,5)as middle _name from employee;

- ✓ OUTPUT:

char_length(employee_name
6
4
5
4
7
6

char_length(employee_name)
6
4
5
4
7
6

employee_name	middle_name
Geetha	etha
Guru	ru
Gokul	kul
Mani	ni
Moorthy	orthy
Amutha	utha

MYSQL DATE FUNCTIONS

- ✓ DATE ADD
- ✓ DATEDIFF
- ✓ TIMESTAMP
- ✓ DATE FORMAT
- ✓ YEAR
- ✓ DAY
- ✓ MONTH
- ✓ NOW

DATEADD, TIMESTAMP, DATEDIFF

- ✓ select employee _ name, date _add (date _of _ join, interval 3 year)as add _ years from employee;
- ✓ select employee _ name, date diff (cur date(), date _ of _ join) as employee_ exp from employee;
- ✓ Select employee _ name ,time stamp diff (month, date _ of _join, cur date())from employee;

employee_name	add_years
Geetha	2025-05-10
Guru	2025-05-12
Gokul	2025-05-15
Mani	2025-05-20
Moorthy	2025-05-23
Amutha	2025-06-05

employee_name	employee_exp
Geetha	662
Guru	660
Gokul	657
Mani	652
Moorthy	649
Amutha	636

employee_name	timestampdiff(month,
Geetha	21
Guru	21
Gokul	21
Mani	21
Moorthy	21
Amutha	20

DATE FORMAT

%a	Abbreviated weekday name (Sun..Sat)
%b	Abbreviated month name (Jan..Dec)
%c	Month, numeric (0..12)
%D	Day of the month with English suffix (0th, 1st
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%f	Microseconds (000000..999999)
%H	Hour (00..23)

DATE FORMAT

%h	Hour (01..12)
%I	Hour (01..12)
%i	Minutes, numeric (00..59)
%j	Day of year (001..366)
%k	Hour (0..23)
%l	Hour (1..12)
%M	Month name (January..December)
%m	Month, numeric (00..12)
%p	AM or PM
%r	Time, 12-hour (hh:mm:ss followed by AM or PM)
%S	Seconds (00..59)

DATE FORMAT

%T	Time, 24-hour (hh:mm:ss)
%U	Week (00..53), where Sunday is the first day of the week; WEEK() mode 0
%u	Week (00..53), where Monday is the first day of the week; WEEK() mode 1
%V	Week (01..53), where Sunday is the first day of the week; WEEK() mode 2; used with %X
%v	Week (01..53), where Monday is the first day of the week; WEEK() mode 3; used with %x
%W	Weekday name (Sunday..Saturday)
%w	Day of the week (0=Sunday..6=Saturday)
%X	Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
%x	Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
%Y	Year, numeric, four digits

DATE FORMAT,MONTH

- ✓ select date_ format (date _of _ join ,'%b')from employee;
- ✓ select*from employee where month(date _ of_ join)=05;

- ✓ OUTPUT:

	date_format(date_of_join,'%b')
▶	May
	May
	May
	May
	May
	Jun
	Jun
	Jun
	Jun

	employee_ID	employee_Name	designation_id	dep_no	date_of_join
▶	17001	Geetha	3001	50	2022-05-10
	17002	Guru	3002	50	2022-05-12
	17003	Gokul	3003	50	2022-05-15
	17004	Mani	3004	60	2022-05-20
	17005	Moorthy	3005	50	2022-05-23
	NULL	NULL	NULL	NULL	NULL

YEAR, DAY, NOW

- ✓ select*from employee where month(date _ of_ join)=05;
- ✓ select*from employee where day(date _of _join);
- ✓ select cur date()as now_ date from employee;

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05

now_date
2024-03-02
2024-03-02
2024-03-02
2024-03-02
2024-03-02
2024-03-02
2024-03-02

employee_ID	employee_Name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
17004	Mani	3004	60	2022-05-20
17005	Moorthy	3005	50	2022-05-23
17006	Amutha	3006	50	2022-06-05

LOGICAL FUNCTIONS

- IF
- IF WITH AND CONDITIONS
- IF WITH OR CONDITIONS
- COUNT IF

IF,COUNT IF

- ✓ select*,if(month (date _ of_ join)>5,'yes','no') as joining _ month from employee;
- ✓ Select amount, count(if(amount>=30000,'1','null')) as employee _ count from salary _ det group by amount;

- ✓ **OUTPUT:**

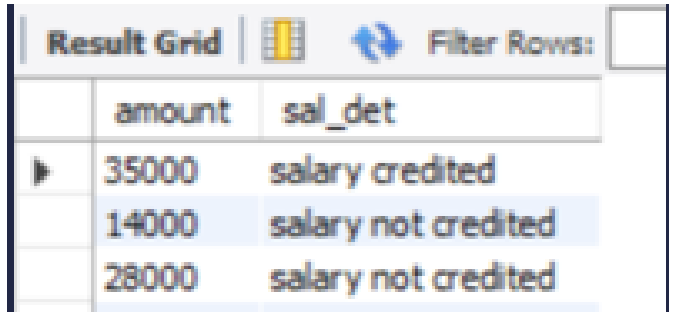
employee_ID	employee_Name	designation_id	dep_no	date_of_join	joining_mon
17001	Geetha	3001	50	2022-05-10	no
17002	Guru	3002	50	2022-05-12	no
17003	Gokul	3003	50	2022-05-15	no
17004	Mani	3004	60	2022-05-20	no
17005	Moorthy	3005	50	2022-05-23	no
17006	Amutha	3006	50	2022-06-05	yes
17007	Jaga	3003	70	2022-06-06	yes
17008	Pavithra	3007	60	2022-06-07	yes

amount	employee_count
14000	5
18000	3
23000	2
28000	3
30000	3
35000	2

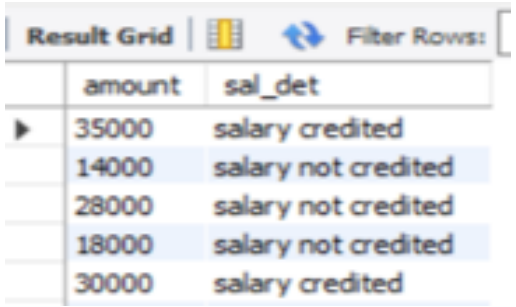
IF WITH AND ,IF WITH OR

```
select amount,  if((amount>=30000) and  (amount>=35000),'salary
credited', 'salary not credited' ) as salary _ det from table2;
```

```
select amount, if((amount>=30000)
or (amount>=35000),'salary credited', 'salary not credited' )
as salary_ det from table2;
```


Result Grid | Filter Rows:

amount	sal_det
35000	salary credited
14000	salary not credited
28000	salary not credited


Result Grid | Filter Rows:

amount	sal_det
35000	salary credited
14000	salary not credited
28000	salary not credited
18000	salary not credited
30000	salary credited

JOIN QUERIES

LEFT JOIN

- ✓ **select*from employee left join salary_ det on employee. employee _ id=salary_ det. emp _id;**

employee_ID	employee_Name	designation_id	dep_no	date_of_join	emp_id	designation_id	date_of_join	dep_no	amount
17001	Geetha	3001	50	2022-05-10	NULL	NULL	NULL	NULL	NULL
17002	Guru	3002	50	2022-05-12	NULL	NULL	NULL	NULL	NULL
17003	Gokul	3003	50	2022-05-15	NULL	NULL	NULL	NULL	NULL
17004	Mani	3004	60	2022-05-20	NULL	NULL	NULL	NULL	NULL
17005	Moorthy	3005	50	2022-05-23	NULL	NULL	NULL	NULL	NULL
17006	Amutha	3006	50	2022-06-05	NULL	NULL	NULL	NULL	NULL
17007	Jaga	3003	70	2022-06-06	NULL	NULL	NULL	NULL	NULL

JOIN QUERIES

RIGHT JOIN

- ✓ **select*from employee right join salary _ det on employee. employee _ id=salary _ det. Emp _ id;**

- ✓ **OUTPUT:**

	employee_ID	employee_Name	designation_id	dep_no	date_of_join	emp_id	designation_id	date_of_join	dep_no	amount
▶	NULL	NULL	NULL	NULL	NULL	18001	17001	2022-06-10	241	35000
	NULL	NULL	NULL	NULL	NULL	18002	17002	2022-06-12	241	14000
	NULL	NULL	NULL	NULL	NULL	18003	17003	2022-06-15	241	28000
	NULL	NULL	NULL	NULL	NULL	18004	17004	2022-06-20	242	18000
	NULL	NULL	NULL	NULL	NULL	18005	17005	2022-06-23	241	30000
	NULL	NULL	NULL	NULL	NULL	18006	17006	2022-07-06	241	23000
	NULL	NULL	NULL	NULL	NULL	18007	17007	2022-07-07	243	28000
	NULL	NULL	NULL	NULL	NULL	18008	17008	2022-07-08	242	18000

Result 7 ×

JOIN QUERIES

INNER JOIN

```
select * from table1 inner join table2 on
table1.emp_id=table2.emp_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	designation_id	dep_no	date_of_join	salary_id	emp_id	salary_date	branch_id	amount
17001	Geetha	3001	50	2022-05-10	18001	17001	2022-06-10	241	35000
17002	Guru	3002	50	2022-05-12	18002	17002	2022-06-12	241	14000
17003	Gokul	3003	50	2022-05-15	18003	17003	2022-06-15	241	28000
17004	Mani	3004	60	2022-05-20	18004	17004	2022-06-20	242	18000
17005	Moorthy	3005	50	2022-05-23	18005	17005	2022-06-23	241	30000
17006	Amutha	3006	50	2022-06-05	18006	17006	2022-07-06	241	23000
17007	Jaga	3003	70	2022-06-06	18007	17007	2022-07-07	243	28000
17008	Pavithra	3007	60	2022-06-07	18008	17008	2022-07-08	242	18000
17009	Arthi	3005	50	2022-06-08	18009	17009	2022-07-09	241	30000
17010	Kabilan	3006	70	2022-06-09	18010	17010	2022-07-10	243	23000
17011	Manasi	3001	70	2022-06-10	18011	17011	2022-07-11	243	35000
17012	Suja	3002	50	2022-06-11	18012	17012	2022-07-12	241	14000
17013	Arun	3003	60	2022-06-12	18013	17013	2022-07-13	242	28000

JOIN QUERIES

FULL OUTER JOIN

(select*from employee left join on employee. dep_ no=department .dep_ no) union(select*from employee right join on employee. dep_ no= department. Dep _ no);

emp_id	emp_name	designation_id	dep_no	date_of_join	dep_no	dep_name	branch_id	branch_name
17001	Geetha	3001	50	2022-05-10	50	Production Department	241	Annan Nagar
17002	Guru	3002	50	2022-05-12	50	Production Department	241	Annan Nagar
17003	Gokul	3003	50	2022-05-15	50	Production Department	241	Annan Nagar
17005	Moorthy	3005	50	2022-05-23	50	Production Department	241	Annan Nagar
17006	Amutha	3006	50	2022-06-05	50	Production Department	241	Annan Nagar
17009	Arthi	3005	50	2022-06-08	50	Production Department	241	Annan Nagar
17012	Suja	3002	50	2022-06-11	50	Production Department	241	Annan Nagar
17016	Madhavi	3002	50	2022-06-15	50	Production Department	241	Annan Nagar
17004	Mani	3004	60	2022-05-20	60	HR Department	242	Velachery
17008	Pavithra	3007	60	2022-06-07	60	HR Department	242	Velachery
17013	Arun	3003	60	2022-06-12	60	HR Department	242	Velachery
17014	Deepa	3004	60	2022-06-13	60	HR Department	242	Velachery
17025	Devan	3006	60	2022-06-24	60	HR Department	242	Velachery

CASE END

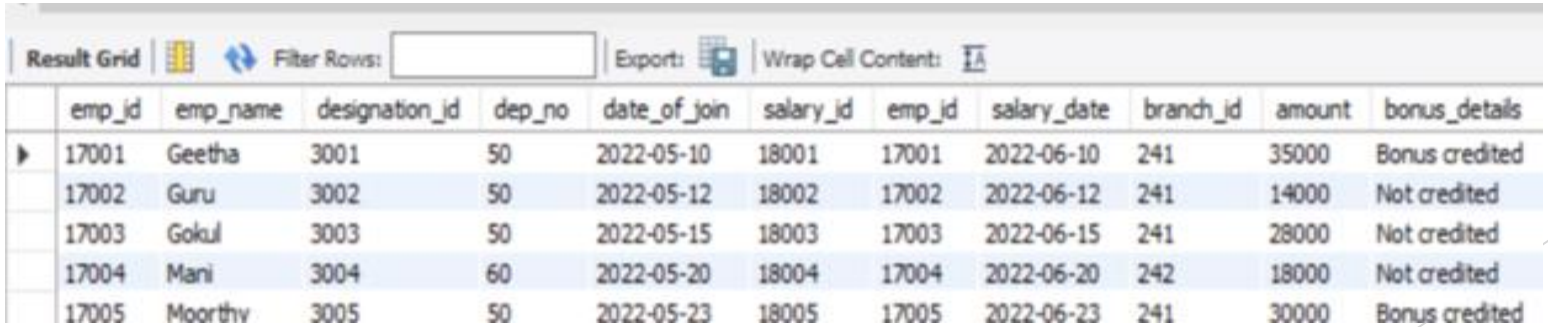
- ✓ select*,case when salary _ det. amount<=300000 then 'salary credited' else 'salary not credited' end as salary _ inf from salary _ det;

emp_id	designation_id	date_of_join	dep_no	amount	sal_inf
18001	17001	2022-06-10	241	35000	salary credited
18002	17002	2022-06-12	241	14000	salary credited
18003	17003	2022-06-15	241	28000	salary credited
18004	17004	2022-06-20	242	18000	salary credited
18005	17005	2022-06-23	241	30000	salary credited
18006	17006	2022-07-06	241	23000	salary credited
18007	17007	2022-07-07	243	28000	salary credited
18008	17008	2022-07-08	242	18000	salary credited

CASE WITH AND STATEMENTT

QUERY:

```
select*,case when salary _ det .amount>=35000 and
(employee .dep_ no)=50  then 'bonus credited' Else "not
credited" end as bonus  _ details from employee inner join
salary det  on employee. employee_ id=bonus_ details.
Emp _  id;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	designation_id	dep_no	date_of_join	salary_id	emp_id	salary_date	branch_id	amount	bonus_details
17001	Geetha	3001	50	2022-05-10	18001	17001	2022-06-10	241	35000	Bonus credited
17002	Guru	3002	50	2022-05-12	18002	17002	2022-06-12	241	14000	Not credited
17003	Gokul	3003	50	2022-05-15	18003	17003	2022-06-15	241	28000	Not credited
17004	Mani	3004	60	2022-05-20	18004	17004	2022-06-20	242	18000	Not credited
17005	Moorthy	3005	50	2022-05-23	18005	17005	2022-06-23	241	30000	Bonus credited

CASE WITH OR STATEMENT

QUERY:

```
select*,case when salary _ det .amount>=35000 or (employee
.dep_ no)=50  then 'bonus credited' Else "not credited" end as
bonus  _details from employee inner join salary _ det  on employee.
employee_ id=bonus_ det. Emp _  id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	designation_id	dep_no	date_of_join	salary_id	emp_id	salary_date	branch_id	amount	bonus_details
17001	Geetha	3001	50	2022-05-10	18001	17001	2022-06-10	241	35000	Bonus credited
17002	Guru	3002	50	2022-05-12	18002	17002	2022-06-12	241	14000	Bonus credited
17003	Gokul	3003	50	2022-05-15	18003	17003	2022-06-15	241	28000	Bonus credited
17004	Mani	3004	60	2022-05-20	18004	17004	2022-06-20	242	18000	Bonus credited
17005	Moorthy	3005	50	2022-05-23	18005	17005	2022-06-23	241	30000	Bonus credited
17006	Amutha	3006	50	2022-06-05	18006	17006	2022-07-06	241	23000	Bonus credited
17007	Jaga	3003	70	2022-06-06	18007	17007	2022-07-07	243	28000	Bonus credited
17008	Pavithra	3007	60	2022-06-07	18008	17008	2022-07-08	242	18000	Bonus credited
17009	Arthi	3005	50	2022-06-08	18009	17009	2022-07-09	241	30000	Bonus credited
17010	Kabilan	3006	70	2022-06-09	18010	17010	2022-07-10	243	23000	Bonus credited

HAVING

QUERY:

- Select*,round(date diff (cur date ()date _ of _ join)/365,0)as emp _ exp from employee having emp exp;

Result Grid | Filter Rows: | Export: | Wrap Cell Con

emp_id	emp_name	designation_id	dep_no	date_of_join	emp_exp
17001	Geetha	3001	50	2022-05-10	2
17002	Guru	3002	50	2022-05-12	2
17003	Gokul	3003	50	2022-05-15	2
17004	Mani	3004	60	2022-05-20	2
17005	Moorthy	3005	50	2022-05-23	2
17006	Amutha	3006	50	2022-06-05	2
17007	Jaga	3003	70	2022-06-06	2
17008	Pavithra	3007	60	2022-06-07	2
17009	Arthi	3005	50	2022-06-08	2
17010	Kabilan	3006	70	2022-06-09	2

DECLARING VARIABLES IN PROCEDURES

```
Create procedure store _ data 102()
Begin
Declare n int;
Set n=1;
Select n;
End//
Delimiter ;
Call store_data102;
```

DECLARING AND STORING VARIABLES IN PROCEDURE

```
Delimiter //
Create procedure store _ data103()
Begin
Declare emp _ count int;
Select count(emp _ id)info emp _ count from employee;
Select emp _ count;
end //
Delimiter ;
Call store _ data103();
```

TRIGGERS

- ✓ BEFORE INSERT
- ✓ AFTER INSERT
- ✓ BEFORE UPDATE
- ✓ AFTER UPDATE
- ✓ BEFORE DELETE
- ✓ AFTER DELETE

BEFORE INSERT

```
create table employee(emp _id int, employee_ name varchar(30),
Designation _ int, dep _ no int, date _of_ join date, primary
    key(emp_ id));
delimiter //
create trigger name_ check before insert
on employee for each row
begin
if new. Emp _ name is null
then set new. emp _ name='kindly update your name';
end if;
end //
delimiter ;
```

AFTER INSERT

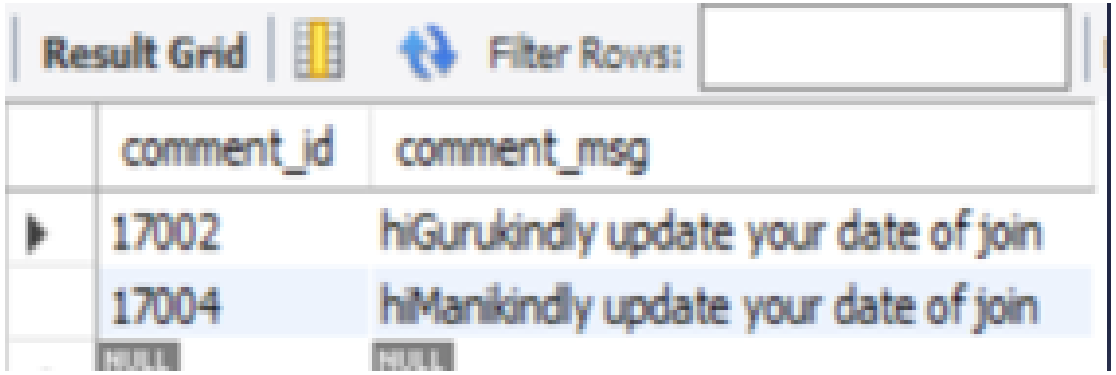
QUERY:

```
create table employee _det(emp_ id int, emp_ name
varchar(30),designation_ id int, dep_ no int, date_ of_ join date, primary
key(emp _ id));

create table msg_info1 (comment _ id int, comment_ msg
varchar(50),primary key(comment _ id));

delimiter //

create trigger detail_ check after insert

on employee_ det for each row

begin

if new. date _ of_ join is null

then  insert into msg _ info(comment _ id, comment_ msg) values (new. emp_
id, concatenate('hi', new. emp_  name, 'kindly update your date of join'));
```

OUTPUT

Result Grid | Filter Rows: | Edit:

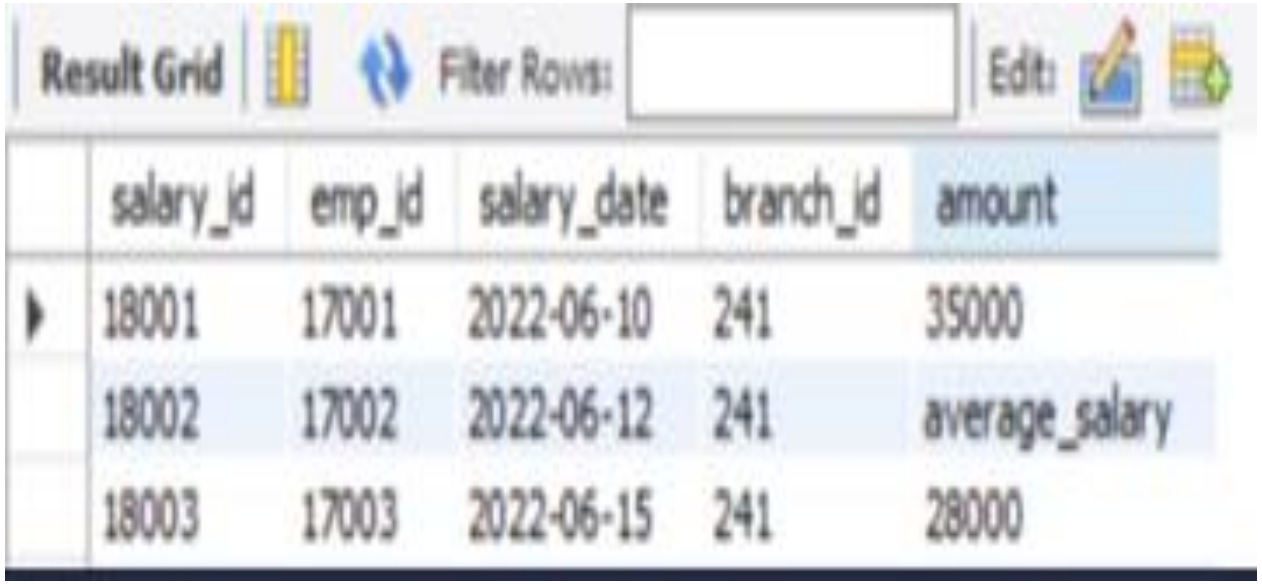
emp_id	emp_name	designation_id	dep_no	date_of_join
17001	harini	3001	50	2022-05-10
17002	Guru	3002	50	NULL
17003	priya	3003	50	2022-05-15
17004	Mani	3004	60	NULL
17005	Moorthy	3005	50	2022-05-23
NULL	NULL	NULL	NULL	NULL



BEFORE UPDATE

```
create table salary_ info(salary _id
    int, emp_ id int, salary_ date  ,
    branch_ id int, amount int,
    primary key(emp_ id));

insert into salary_ info values

(18001,17001,'2022-10',241,35000),

(18002,17002,'2022-12',241,14000);

delimiter //

create trigger update_ salary before
    update on salary _info for each
    row begin

if new. amount>=40000 then

set new. amount='high _salary';

elseif new. amount>=35000 then

set new. amount='Good _salary';

elseif new. amount>=15000 then
set new. amount='average_ salary';
elseif new. amount>=0

then set new. amount='Low_
salary';

 end if;

end //

delimiter ;

select * from salary _info; alter
table salary_ info modify amount
varchar(40); update salary _info set
amount=25000 where emp_
id=17002; select * from salary_
info;
```

OUTPUT



salary_id	emp_id	salary_date	branch_id	amount
18001	17001	2022-06-10	241	35000
18002	17002	2022-06-12	241	average_salary
18003	17003	2022-06-15	241	28000

BEFORE DELETE

```
create table emp_ info (emp id int,
emp_ name varchar(30),designation_
id int, dep_ no int ,date_ of_ join date,
primary key(emp _id));

insert into emp _info values

(17001,'Geetha',3001,50,'2022-5-10'),

(17002,'Guru',3002,50,'2022-5-12'),

create table emp _ info _backup (emp_
id int, emp  _name
varchar(30),designation _id int, dep_
no int, date  _ of _ join date, primary
key(emp _id));

delimiter //

create trigger backup_update1 before
delete on emp _info for each row

begin

insert into emp_ info_ backup
values(old.emp_id,old.emp_name,old
.designation_id,old.dep_no,old.date_
of_join);

end //

delimiter ;

select * from emp _info;

select * from emp_ info_ backup;

delete from emp_ info where emp
_id=17001;
```

OUTPUT

Result Grid | Filter Rows: | Edit:

emp_id	emp_name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
17002	Guru	3002	50	2022-05-12
17003	Gokul	3003	50	2022-05-15
NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit:

emp_id	emp_name	designation_id	dep_no	date_of_join
17001	Geetha	3001	50	2022-05-10
NULL	NULL	NULL	NULL	NULL

THANK YOU