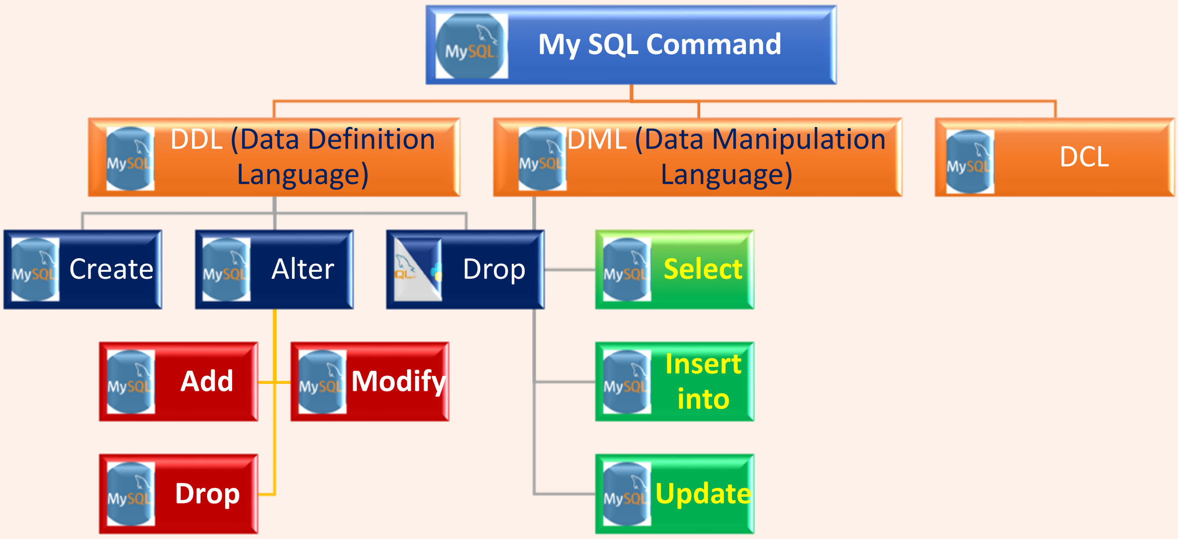
MYSQL (Important Notes)



Create Table Command is used to create the structure of a table. We can specify the various fields along with their data types, size and range of values which can be stored in those fields. We can also specify the constraints applied on fields.

Syntax:

Create table <table_name>

(Column1 <data_type>(size) constraint,

Column2 <data_type>(size) constraint,

……………………………………….

……………………………………….

);

Here **table_name** is valid name of table.

Column1,Column2,…. are the names of various field/attribute of table.

Data_Type is the name of valid data type. It is used to specify the type of value which can be stored in a particular column.

Size specifies the maximum number of digits or characters which can be stored in a particular field. It is optional in case of number data type.

constraint is the name of constraint that we want to apply on a particular field.

Example:

We have to create a table named **Student** with following structure:

Field name	Data Type	Size	Constraint
roll	Number	3	Primary Key
name	Varchar	20	
dateofbirth	Date		
fees	Number	7,2	
Class	Varchar	20	

SQL command for creating table with above structure is

```
Create table student(roll number(3) primary key,

name varchar(20),

dateofbirth date,

fees number(7,2),

Class varchar(20)

);
```

2. Alter Table

This statement is used to modify the structure of a table. We can add or modify fields of an existing table.

We can also specify the data type, size, constraints and domain of fields which are to be added or modified in an existing table.

We can increase the size of any field at any time but we should be careful while decreasing size of any field as it may result in data loss.

Alter table command can be used in two ways

Syntax 1:

```
Alter table <table_name>

add ( Column1  <data_type>(size) constraint,

Column2  <data_type>(size) constraint,

………………………………………

);
```

Here **table_name** is name of any existing table.

add specifies that new field is to be added to the table.

Column1,Column2,…. are the names of various field which are to be added to table.

data_type is the name of valid data type of Oracle. It is used to specify the type of value which can be stored in a particular column.

size specifies the maximum number of digits or characters which can be stored in a particular field. It is optional in case of number data type.

constraint is the name of constraint that we want to apply on a particular field.

Example 1:

```
Alter table student add (address varchar2(50));
```

Above command will add a new field address to table named student.

Example 2:

```
Alter table student add constraint u1 unique(address);
```

Above command will add a unique constraint on address field of student table.

Syntax 2:

```
Alter table <table_name>

modify( Column1  <data_type>(size) constraint,

Column2  <data_type>(size) constraint, …………………………………….. );
```

modify specifies that an existing field of a table is to be modified.

Example

```
Alter table student modify(name varchar2(25));
```

Above command will increase the width of field named “name” to 25.

Syntax 3:

```
Alter table <table_name>

drop column Column_name ,……………………………………..;
```

drop column specifies that an existing field of a table is to be deleted.

Example:

Alter table student drop column address;

Above command will the delete the field address from the table student permanently.

3. Drop Table

This statement is used to delete the structure of a table as well as data stored in a particular table.

Syntax:

Drop table <table_name>

Here **table_name** is any valid name of Oracle.

Example

drop table student;

Above command will delete the table named student permanently.

DML Commands of SQL

There are four types of DML commands.

1. Insert

This command used to insert a new record in an existing table. We can insert a complete record or we can enter record values in specific fields also.

Syntax:

```
Insert into table_name(field1,field2,……….)

values (datavalue1, datavalue2, ………..);
```

Here **table_name** is the name of an existing table

field1,field2… represent the names of various fields of table

datavalue1,datavalue2, … represent the values which are to be inserted

into field1,field2,…….

Example 1:

```
Insert into student values (101 ‘amit’, ‘2000–01-05’,5000.50);
```

Above statement will insert a complete record in table named student.

101 will store in roll field, ‘amit’ in name field, ‘2000-01-05’ in dateofbirth field and 5000.50 in fees field.

Example 2:

```
Insert into student(roll,name ) values(102, ‘sumit’)
```

Above will only insert values in two fields named roll and name. 102 will be stored in roll field, ‘sumit’ in name field, dateofbirth and fees fields will get null.

2. update

This command is used to modify an existing record . We can modify more than one field at a time and we can also modify those records which satisfy a particular condition.

Syntax: –

```
Update     <table_name>
set column_name1= expression1 ,
column_name2= expression2 , ……..
where <condition >;
```

Here **table_name** is the name of an existing table

column_name1, column_name2 represent the names of various fields of table whose contents are to be modified.

expression1, expression2 represent the values or expressions which are to be stored into fields specified by columnname1, columnname2.

condition is any relational or logical expression which is used to specify the condition according to which records should be modified.

Example 1:

```
Update student set roll=106;
```

Above statement ill make all the values in roll field to be 106.

Example 2:

```
Update student set fees=fees+100;
```

Above statement will increase the fees by 100 in all records.

Example 3:

```
Update student set roll=roll+10 where fees>4500;
```

Above statement will increase the roll by 10 in those records where fees>4500.

3. Delete

This command is used to delete all the records or an existing record from a database table. Structure of table will not be deleted.

Syntax:

```
Delete  from<table_name>
where <condition >;
```

Here **table_name** is the name of an existing table

condition is any relational or logical expression which is used to specify the condition as per which records should be deleted.

Example 1

```
Delete from student;
```

Above command will delete all the records from table named student.

Example 2

```
Delete from student where fees>4000;
```

Above command will delete those records from table named student which is having fees more than 4000.

4. Select

This command is used to view all the records or specific records from a database table depending on a particular condition.

Syntax:

```
Select [*] / [column1,column2,………..]

from <table_name>

where <condition >;
```

Here **table_name** is the name of an existing table

***** represents that all the records from the table should be shown on the screen.

column1,column2 represent the names of fields whose contents are to be shown.

condition is any relational or logical expression which is used to specify the condition as per which records should be deleted.

Example 1

```
Select * from student;
```

Above command will show all the records from table named student.

Example 2

```
Select roll, name from student ;
```

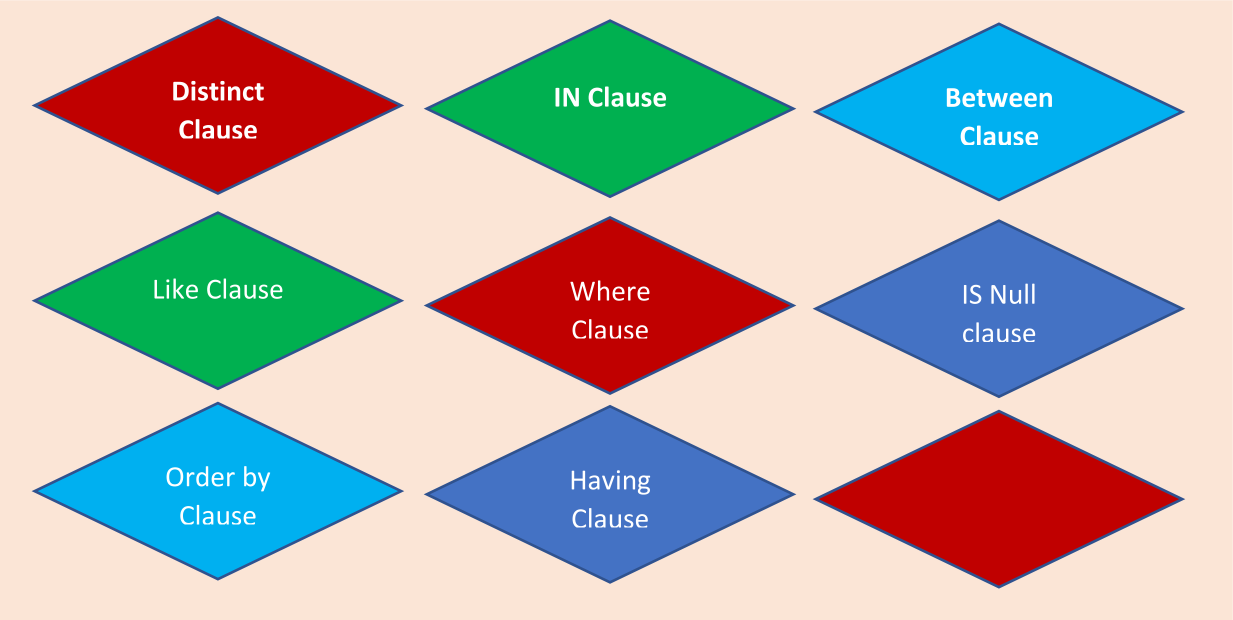
Above command will show contents of fields named roll and name for those records from table named student which have fees more than 4000.

Example 3

```
Select * from student where fees>4000;
```

Above command will show all those records from table named student which have fees more than 4000.

Clauses used with DML Commands



Distinct Clause

IN Clause

Between Clause

Like Clause

Where Clause

IS Null clause

Order by Clause

Having Clause

1. DISTINCT Clause

This clause is used with select command to view Unique values in a particular field.

Example

select distinct class from emp;

This command will show different values in the class field of student table avoiding duplicate values.

2. IN Clause

This clause is used to specify a range of values. This clause is used with select, delete, update statements.

Example

Select * from student where class in (‘xi’,’xii’);

OR

Select * from student where city =‘xi’ or city=’xii’;

This command is used to view those records where class field of student table contains either ‘xi’ or ‘xii’

3. BETWEEN Clause

The BETWEEN clause allows to check if an expression is within a range of two values. The values can be text, date, or numbers. The BETWEEN clause used in a condition will return the records where value of expression is within the range of value1 and value2.

Example:

Select * from student where fees between 1000 and 2000;

OR

Select * from student where fees>=1000 and fees<=2000;

In the above examples, those records from table student will be shown where value of fees is between 1000 and 2000. Both values 1000 and 2000 are included in the output.

4. LIKE Clause

The LIKE clause is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards used with LIKE clause:

- % : The percent sign represents zero or more characters.
- _ : The underscore represents a single character.

Examples

LIKE Operator	Description
WHERE Name LIKE ‘A%’	**Finds values that start with “A”**
WHERE Name LIKE ‘%s’	**Finds values that end with “s”**
WHERE Name LIKE ‘%or%’	**Finds values that have “or” in any position**
WHERE Name LIKE ‘_a%’	**Finds values that have “a” in the second position**
WHERE Name LIKE ‘a_%’	**Finds values that start with “a” and are at least 2 characters in length**
WHERE Name LIKE ‘a__%’	**Finds values that start with “a” and are at least 3 characters in length**
WHERE Name LIKE ‘a%s’	**Finds any values that start with “a” and ends with “s”**

i. Select * from student where name like ‘A%’;

This command will show all those records from table named student where name field contains values starting with A for e.g. Arun, Aman, Anu , Amandeep etc.

ii. Select * from student where name like '%RU%';

This command will show all those records from table student where name field contains values containing RU in it for e.g. ARUN, VARUN, KARUNA.

iii. Select * from student where name like 'A_ _ _';

This command will show all those records from table student where name field contains values starting with A and having four characters for e.g. Arun, Aman etc.

iv. Select * from student where name like 'A_ _N';

This command will show all those records from table student where name field contains values starting from A and ending with N and four characters in it for e.g. ARUN, AMAN etc.

5. IS NULL Clause

NULL means empty. It is neither Zero nor blank but it means "not defined". This clause is used with select command to view those records which contain null values in a particular field i.e Field doesn't contain any value in various records.

Example

i.. select * from student where fees is null;

This command will show all the records from student table where fees field contains null value.

ii.. select * from student where fees is not null;

This command will show all the records from student table where fees field does not contain null value.

6. ORDER BY Clause

This clause is basically used to sort the records as per a particular field. It is used with select command to view records alphabetically or numerically in ascending order or descending order with respect to a particular field.

We can user word "asc" to specify that records should be in ascending order. It is the default order when when we user order by clause. We use "desc" with order by clause to show the records in descending order of a particular field.

Example

i. select * from student order by name;

OR

select * from student order by name asc;

This command will show all the records from student table in ascending order of field "name".

ii. select * from student order by name desc;

This command will show all the records from student table in descending order of field "name".

7. GROUP BY Clause

This clause can be used with select statement. This clause is used to group records based on distinct values that exist for specified columns i.e. it creates a data set containing several sets of records grouped together based on a condition.

We can specify the condition for which records should be grouped by using "having" clause.

Example:

i. Select class, sum(fees) from student group by job ;

This command will show the classes along with sum of fees corresponding to various classes in the table student.

ii. Select class, sum(fees) from student group by job having count(class)>3;

This command will show the classes along with sum of fees corresponding to various classes where number of records corresponding to a particular class are more than 3.

SQL Functions

i. SUM

This function is used to find the sum of values stored in a numeric field of a table.

Syntax:

SUM(Field_Name)

Field_Name is the name of any numeric field on which we want to apply sum Function.

Example:

select sum(fees) from student;

This command will compute the sum of values stored in fees field of student table.

ii AVG:

This function is used to find the average of values stored in a numeric field of a table.

Syntax:

AVG(Field_Name)

Field_Name is the name of any numeric field on which we want to apply avg Function.

Example:

select avg(fees) from student;

This command will compute the average of values stored in fees field of student table.

iii. MIN

This function is used to find the minimum value among the values stored in a numeric field of a table.

Syntax:

MIN(Field_Name)

Field_Name is the name of any numeric field on which we want to apply MIN Function.

Example:

select MIN(fees) from student;

This command will find minimum fees from student table.

iv. MAX

This function is used to find the maximum value among the values stored in a numeric field of a table.

Syntax:

MAX(Field_Name)

Field_Name is the name of any numeric field on which we want to apply MIN Function.

Example:

select MAX(fees) from student;

This command will find maximum fees from student table.

v. COUNT

This function is used to count the number of records in a particular field or in complete table of Oracle.

Syntax

count(Expression/*)

Expression may be the name of any field of table or any expression based on a field of a table.

***** is used when we want to know the number of records in a table.

Example

i. select count(fees) from student;

This command will show the number of non null values stored in fees field of student table.

ii. select count(*) from student;

This command will show the total number of records stored in student table.

SQL Joins

It is the way to combine records of more than one table in which there must exist one common field on which we create a relation between two tables using JOIN.

I. EQUI JOIN

This type of join is used to combine records from tables where the common field of both tables have same value.

Equi join is a special type of join in which we use only an equality (=) operator to select only those records having same value in common field.

Example:

Table: Foods

```
+---------+--------------+-----------+
| ITEM_ID | ITEM_NAME    | COMPANY_ID|
+---------+--------------+-----------+
| 1       | Chex Mix     | 16        |
| 6       | Cheez-It     | 15        |
| 2       | BN Biscuit   | 15        |
| 4       | Pot Rice     | 15        |
+---------+--------------+-----------+
```

Table: Company

```
+------------+---------------+
| COMPANY_ID | COMPANY_NAME  |
+------------+---------------+
| 15         | Jack Hill     |
| 16         | Akas Foods    |
| 17         | Foodies.      |
| 19         | sip-n-Bite.   |
+------------+---------------+
```

SELECT FOODS.ITEM_ID,FOOD.ITEM_NAME, COMPANY.COMPANY_ID,COMPANY.COMPANY_NAME

```
FROM FOODS,COMPANY
WHEREFOODS.COMPANY_ID=COMPANY.COMPANY_ID;
```

OUTPUT

```
+---------+--------------+-----------+--------------+
| ITEM_ID | ITEM_NAME    | COMPANY_ID| COMPANY_NAME |
+---------+--------------+-----------+--------------+
| 1       | Chex Mix     | 16        |  Akas Foods  |
| 6       | Cheez-It     | 15        |  Jack Hill   |
| 2       | BN Biscuit   | 15        |  Jack Hill   |
| 4       | Pot Rice     | 15        |  Jack Hill   |
+---------+--------------+-----------+--------------+
```

Output will show only those records from both the tables where company_id is same.

ii. NATURAL JOIN

A natural join is a type of equi join which occurs implicitly by comparing all the same names columns in both tables. The join result has only one column for each pair of equally named columns.

Natural Join has following conditions:

- The tables to be joined must have one or more columns having same name.
- The columns must be the same data type.

Syntax:

```
SELECT *
FROM table1
NATURAL JOIN table2;
```

Example:

Table: Foods

```
+---------+--------------+-----------+
| ITEM_ID | ITEM_NAME    | COMPANY_ID|
+---------+--------------+-----------+
| 1       | Chex Mix     | 16        |
| 6       | Cheez-It     | 15        |
| 2       | BN Biscuit   | 15        |
| 4       | Pot Rice     | 15        |
+---------+--------------+-----------+
```

Table: Company

```
+------------+---------------+
| COMPANY_ID | COMPANY_NAME  |
+------------+---------------+
| 15         | Jack Hill     |
| 16         | Akas Foods    |
| 17         | Foodies.      |
| 19         | sip-n-Bite.   |
+------------+---------------+
```

```
SELECT *
FROM foods  NATURAL JOIN company;
```

OUTPUT

```
+---------+--------------+-----------+--------------+
| ITEM_ID | ITEM_NAME    | COMPANY_ID| COMPANY_NAME |
+---------+--------------+-----------+--------------+
| 1       | Chex Mix     | 16        |  Akas Foods  |
| 6       | Cheez-It     | 15        |  Jack Hill   |
| 2       | BN Biscuit   | 15        |  Jack Hill   |
| 4       | Pot Rice     | 15        |  Jack Hill   |
+---------+--------------+-----------+--------------+
```

Output will show only those records from both the tables where company_id is same.

Q 1. Write SQL queries on the basis of following table.

Relation : Student				
Name	**Class**	**Fee**	**Gender**	**DOB**
Rahul	XII	1200	M	2005-02-01
Mehul	XII	1200	M	2004-12-11
Manisha	XI	1050	F	2006-10-12
Sujoy	XI	1050	M	NULL
Sonakshi	XII	1200	F	2005-09-19
Suman	X	950	F	2008-06-16

i. **Display all records from table student.**

Select * from student;

ii. **List names of all students from table student.**

Select name from student;

iii. **List name and class of all students from table student.**
Select name,class from student;

iv. List all students studying in class XII.

Select * from student where class=’XII’;

iv. List names of male students .

Select name from student where gender=’M’;

v. **Show information of students having fee more than 1000.**
Select * from student where fee>1000;

vi. **Show name, fee and class of students having fee below or equal to 1100.**
Select name,fee,class from student where fee<=1100;

vii. **View all records in ascending order of student's name.**

Select * from student order by name;

viii. **View name and fee of all students in descending order of fee.**

Select name,fee from student order by fee desc;

ix. **List male students in descending order of their class.**

Select * from student where gender='M' order by class;

Q2. Write SQL queries on the basis of following table.

Relation : Teacher

TName	Incharge	Salary	Gender	Dateofjoin
Rahul	XII	12000	M	2015-02-11
Mehul	XII	12500	M	2014-12-21
Manisha	XI	13500	F	2016-10-22
Sujoy	XI	21050	M	2016-10-22
Sonakshi	XII	21000	F	2015-08-19
Suman	X	16950	F	2018-06-26

i. Show information of teachers having salary more than 20000.

Select * from teacher where salary>20000;

ii. List teachers who are incharge of classes X or XII.

Select * from teacher where incharge='X' or incharge='XII';

OR

Select * from teacher where incharge in ('X','XII');

iii List teachers teaching class X or having salary more than 10000.

Select * from teacher where incharge='X' or salary>10000;

iv. View those records where salary is between 10000 and 20000 .

Select * from teacher where salary between 10000 and 20000;

OR

```
Select * from teacher where salary>=10000 and  salary<=20000;
```

v. List female teachers having salary 21000 .

```
Select * from teacher where gender=’F’ and salary=21000;
```

vi. List different classes available in above table (no duplicates) .

```
Select distinct incharge from teacher;
```

vii. List salary and names of teachers having salary less than 15000 or more than 20000 .

```
Select salary,tname from teacher where salary<15000 or salary>20000;
```

viii. List name, salary and annual salary(salary*12) of all teachers.

```
Select tname, salary, salary*12  from teacher;
```

ix. List name of all teachers teaching class XII.

```
Select tname from teacher where incharge=’XII’;
```

x. List name and salary of all teachers who joined before ‘2016-10-01’.

```
Select tname, salary from teacher where dateofjoin<‘2016-10-01’;
```

Q3. Write SQL queries on the basis of following table.

Relation: Hospital

PName	Fee	Gender	Dateofvisit
Ramesh	200	M	2020-02-11
Mohnish	250	M	2019-12-22
Muskan	350	F	2019-11-22
Sunil	250	M	2018-12-02
Sonam	null	F	2019-01-19
Sahil	16950	F	2019-02-26

A. **Show information of patients who visited after ‘2020-01-01’.**

```
Select * from hospital where dateofvisit>’2020-01-01’;
```

B. **Show information of patients who visited between ‘2018-12-01’ and ‘2019-12-01’.**
Select * from hospital where dateofvisit between ‘2018-12-01’ and ‘2019-12-01’;
OR
Select * from hospital where dateofvisit>= ‘2018-12-01’ and dateofvisit<= ‘2019-12-01’;

C. **Show information of patients who visited before ‘2018-12-01’ or after 2019-12-01’.**
Select * from hospital where dateofvisit< ‘2018-12-01’ or dateofvisit> ‘2019-12-01’;

D. **List those patients whose name starts with alphabet ‘M’.**
Select * from hospital where pname like ‘M%’;

E. **List those patients whose name ends with alphabet ‘m’.**
Select * from hospital where pname like ‘%m’;

F. **List those patients whose name contains alphabet ‘a’.**
Select * from hospital where pname like ‘%a%’;

G. **List those patients whose name starts with alphabet ‘M’ and ends with ‘n’.**
Select * from hospital where pname like ‘M%n’;

H. **List those patients whose name contains 4 characters.**
Select * from hospital where pname like ‘_ _ _ _ ‘;

I. **List those patients whose fee is null.**
OR
List those patients who have not been charged any fee.
Select * from hospital where fee is null;

J. **List those patients where fee is not null.**
Select * from hospital where fee is not null;

K.

Q 4. Write SQL queries on the basis of following table.

Relation : Emp				
Empid	**EmpName**	**Department**	**Salary**	**Gender**
101	Mahesh	Finance	32000	M
303	Vijay	HR	42500	M
401	Mansha	Finance	31500	F

603	Kamal	Computer	32150	M
604	Vandana	HR	42000	F
631	Sujata	Finance	39500	F

a) **Increase salary of all employees by 1000.**

```
update emp set salary=salary+1000;
```

b) **Decrease value of EmpID of male employees by 10.**

```
update emp set empid=empid-10 where gender=’M’;
```

c) **Department of ‘Mahesh’ should be updated as ‘HR’.**

```
update emp set department=’HR’ where empname=’Mahesh’;
```

d) **Name should be updated as ‘Manish Saini’ and salary as 50000 for employee id 603.**

```
update emp set empname=’Manish Saini’, salary=50000 where empid=603;
```

e) **‘Finance’ department should be updated as ‘Fin’.**

```
update emp set department=’Fin’ where department=’Finance’;
```

f) **Delete records of female employees.**

```
delete from emp where gender=’F’;
```

g) **Delete all records of Emp table.**

```
delete from emp;
```

h) **Insert a new record with data as 700,’Raman’,’HR’,40000,’M’**

```
insert into emp values(700,’Raman’,’HR’,40000,’M’);
```

i) **Add a new field Phoneno of integer type to table.**

```
Alter table emp add phoneno int;
```

j) **Set the size of EmpName field to 40.**

```
Alter table emp modify empname varchar(40);
```

Q 5. Write SQL queries on the basis of following table.

Relation : Employee			
Id	**Name**	**Designation**	**Sal**
101	Naresh	Clerk	32000
102	Ajay	Manager	42500
103	Manisha	Clerk	31500
104	Komal	Advisor	32150
105	Varun	Manager	42000
106	NULL	Clerk	32500

a) **Count number of records in the table.**

select count(*) from employee;

b) **Count number of names in the table.**

select count(name) from employee;

c) **Count number of designations.**

select count(distinct designation) from employee;

d) **Count number of clerks.**

select count(*) from employee where designation=’Clerk’;

e) **Find sum of salaries of all employees.**

select sum(sal) from employee;

f) **Find maximum salary in the table.**

select max(sal) from employee;

g) **Find minimum salary in the table.**

select min(sal) from employee;

h) **Find average salary in the table.**

select avg(sal) from employee;

i) **Find minimum and maximum salary of Managers.**

select max(sal),min(sal) from employee where designation='Manager';

j) **Display number of records for each individual designation.**

select designation, count(*) from employee group by designation;

k) **Display number of records along with sum of salaries for each individual designation where number of records are more than 1.**

select designation, count(*), sum(sal) from employee group by designation having count(*)>1;

l) **Display sum of salaries of clerks and managers**

select designation, sum(sal) from employee group by designation having designation in ('Clerk','Manager');

Q 6. Write SQL queries on the basis of following tables.

Relation : Employee		
id	**Name**	**Designation**
101	Naresh	Clerk
102	Ajay	Manager
103	Manisha	Clerk
104	Komal	Advisor

Relation : Details		
id	Salary	**PhoneNo**

101	32000	9898090909
102	45000	9888000909
103	33000	8008078787
104	40000	7009876543

a) **View Name and salary of all employees.**

b) select name,salary from employee, details where employee.id=details.id;

OR

select name,salary from employee E, details D where E.id=D.id;

c) **List Name, designation and salary of all Clerks.**

select name,designation salary from employee, details where employee.id=details.id and designation='Clerk';

OR

select name,designation salary from employee E1, details D1 where E1.id=D1.id and designation='Clerk';

d) **Display id, Name and salary of all employees.**

select employee.id, name,salary from employee, details where employee.id=details.id;
OR
select E1.id, name,salary from employee E1, details D1 where E1.id=D1.id;

e) **Display Name and salary of all employees having salary more than 40000**

select name, salary from employee, details where employee.id=details.id and salary>40000;
OR

select name, salary from employee E, details D where E.id=D.id and salary>40000;

Q 1. Write output of the SQL queries on the basis of following table.

Relation : Student				
Name	**Class**	**Fee**	**Gender**	**DOB**
Rahul	XII	1200	M	2005-02-01
Mehul	XII	1200	M	2004-12-11
Manisha	XI	1050	F	2006-10-12
Sujoy	XI	1050	M	NULL
Sonakshi	X	1200	F	2005-09-19

Rahul	XII	1200	M	2005-02-01
Mehul	XII	1200	M	2004-12-11
Manisha	XI	1050	F	2006-10-12
Sujoy	XI	1050	M	NULL
Sonakshi	X	1200	F	2005-09-19

i. Select * from student;

ii. Select name from student;

Rahul
Mehul
Manisha
Sujoy
Sonakshi

iii. Select name,class from student;

Rahul	XII
Mehul	XII
Manisha	XI
Sujoy	XI
Sonakshi	X

iv. Select * from student where class=’XII’;

Rahul	XII	1200	M	2005-02-01
Mehul	XII	1200	M	2004-12-11

v. Select name from student where gender=’M’;

Rahul
Mehul
Sujoy

vi. Select * from student where fee>1100;

Rahul	XII	1200	M	2005-02-01
Mehul	XII	1200	M	2004-12-11
Sonakshi	X	1200	F	2005-09-19

vii. Select name,fee,class from student where fee<=1100;

Manisha	1050	XI

Sujoy	1050	XI

viii. Select * from student order by name;

Mehul	XII	1200	M	2004-12-11
Manisha	XI	1050	F	2006-10-12
Rahul	XII	1200	M	2005-02-01
Sonakshi	X	1200	F	2005-09-19
Sujoy	XI	1050	M	NULL

ix. Select name,fee from student order by fee desc;

Rahul	1200
Mehul	1200
Manisha	1050
Sujoy	1050
Sonakshi	1200

x. Select * from student where gender='M' order by class;

Mehul	XII	1200	M	2004-12-11
Rahul	XII	1200	M	2005-02-01
Sujoy	XI	1050	M	NULL

Q 2. Write output of SQL queries on the basis of following table.

Relation : Teacher				
TName	**Incharge**	**Salary**	**Gender**	**Dateofjoin**
Suresh	XII	12000	M	2015-02-11
Mohnish	XII	12500	M	2014-12-21
Manisha	XI	13500	F	2016-10-22
Munish	XI	21050	M	2016-10-22
Suman	X	16950	F	2018-06-26

i. Select * from teacher where salary>15000;

Munish	XI	21050	M	2016-10-22
Suman	X	16950	F	2018-06-26

ii. Select * from teacher where incharge=’X’ or incharge=’XII’;
OR
Select * from teacher where incharge in (‘X’,’XII’);

Suresh	XII	12000	M	2015-02-11
Mohnish	XII	12500	M	2014-12-21
Suman	X	16950	F	2018-06-26

iii Select * from teacher where incharge=’X’ or salary<13000;

Suresh	XII	12000	M	2015-02-11

Mohnish	XII	12500	M	2014-12-21
Suman	X	16950	F	2018-06-26

iv. Select * from teacher where salary between 13000 and 20000;
OR
Select * from teacher where salary>=13000 and salary<=20000;

Manisha	XI	13500	F	2016-10-22
Suman	X	16950	F	2018-06-26

v. Select * from teacher where gender=’F’ and salary=16950;

Suman	X	16950	F	2018-06-26

vi. Select distinct incharge from teacher;

XII
XI
X

vii. Select salary,tname from teacher where salary<15000 or salary>20000;

12000	Suresh
12500	Mohnish
13500	Manisha
21050	Munish

viii. Select tname, salary, salary*12 from teacher;

Suresh	12000	144000

Mohnish	12500	150000
Manisha	13500	162000
Munish	21050	252600
Suman	16950	203400

ix. Select tname from teacher where incharge=’XII’;

Suresh	XII	12000	M	2015-02-11
Mohnish	XII	12500	M	2014-12-21

x. Select tname, salary from teacher where dateofjoin<‘2016-10-01’;

Suresh	12000
Mohnish	12500

Q3. Write SQL queries on the basis of following table.

<table>
<tr><th colspan="4">Relation : Hospital</th></tr>
<tr><th>PName</th><th>Fee</th><th>Gender</th><th>Dateofvisit</th></tr>
<tr><td>Ramesh</td><td>200</td><td>M</td><td>2020-02-11</td></tr>
<tr><td>Mohnish</td><td>250</td><td>M</td><td>2019-12-22</td></tr>
<tr><td>Muskan</td><td>350</td><td>F</td><td>2019-11-22</td></tr>
<tr><td>Sunil</td><td>250</td><td>M</td><td>2018-12-02</td></tr>
<tr><td>Sonam</td><td>null</td><td>F</td><td>2019-01-19</td></tr>
</table>

Sahil	16950	F	2019-02-26

i. Select * from hospital where dateofvisit>’2020-01-01’;

Ramesh	200	M	2020-02-11

ii. Select * from hospital where dateofvisit between ‘2018-12-01’ and ‘2019-12-01’;
OR
Select * from hospital where dateofvisit>= ‘2018-12-01’ and dateofvisit<= ‘2019-12-01’;

Muskan	350	F	2019-11-22
Sunil	250	M	2018-12-02
Sonam	Null	F	2019-01-19
Sahil	16950	F	2019-02-26

iii. Select * from hospital where dateofvisit< ‘2018-12-01’ or dateofvisit> ‘2019-12-01’;

Ramesh	200	M	2020-02-11
Mohnish	250	M	2019-12-22
Muskan	350	F	2019-11-22

iv.Select * from hospital where pname like ‘M%’;

Mohnish	250	M	2019-12-22
Muskan	350	F	2019-11-22

v. Select * from hospital where pname like ‘%m’;

Sonam	Null	F	2019-01-19

vi. Select * from hospital where pname like ‘%a%’;

Ramesh	200	M	2020-02-11
Muskan	350	F	2019-11-22
Sonam	Null	F	2019-01-19

vii. Select * from hospital where pname like ‘M%n’;

Muskan	350	F	2019-11-22

viii. Select * from hospital where pname like ‘_ _ _ _ _ ‘;

Sunil	250	M	2018-12-02
Sonam	Null	F	2019-01-19
Sahil	16950	F	2019-02-26

ix. Select * from hospital where fee is null;

Sonam	Null	F	2019-01-19

x. Select * from hospital where fee is not null;

Ramesh	200	M	2020-02-11
Mohnish	250	M	2019-12-22
Muskan	350	F	2019-11-22
Sunil	250	M	2018-12-02
Sahil	16950	F	2019-02-26

Q 4. Write output of following SQL queries on the basis of following table.

<table>
<tr><th colspan="4">Relation : Employee</th></tr>
<tr><th>id</th><th>Name</th><th>Designation</th><th>Sal</th></tr>
<tr><td>101</td><td>Naresh</td><td>Clerk</td><td>30000</td></tr>
<tr><td>102</td><td>Ajay</td><td>Manager</td><td>40000</td></tr>
<tr><td>103</td><td>Manisha</td><td>Clerk</td><td>30000</td></tr>
<tr><td>104</td><td>Komal</td><td>Advisor</td><td>32000</td></tr>
<tr><td>105</td><td>Varun</td><td>Manager</td><td>42000</td></tr>
<tr><td>106</td><td>NULL</td><td>Clerk</td><td>33000</td></tr>
</table>

i. select count(*) from employee;

6

ii select count(name) from employee;

5

iii. select count(distinct designation) from employee;

3

iv. select count(*) from employee where designation='Clerk';

3

v. select sum(sal) from employee;

207000

vi. select max(sal) from employee;

42000

vii. select min(sal) from employee;

30000

viii. select avg(sal) from employee;

34500

ix. select max(sal),min(sal) from employee where designation='Manager';

42000   40000

x. select designation, count(*) from employee group by designation;

Clerk	3
Manager	2
Advisor	1

xi. select designation, count(*), sum(sal) from employee group by designation having count(*)>1;

Clerk	3	93000
Manager	2	82000
Advisor	1	32000

xii. select designation, sum(sal) from employee group by designation having designation in ('Clerk','Manager');

Clerk	93000
Manager	82000

Q5. Write output of following SQL queries on the basis of tables as given.

Relation : Employee		
id	**Name**	**Designation**
101	Naresh	Clerk

102	Ajay	Manager
103	Manisha	Clerk
104	Komal	Advisor

Relation : Details		
id	Salary	**PhoneNo**
101	32000	9898090909
103	33000	8008078787
104	40000	7009876543

i. select name,salary from employee, details where employee.id=details.id;
OR
select name,salary from employee E, details D where E.id=D.id;

Naresh	32000
Manisha	33000
Komal	40000

ii. select name,designation salary from employee, details where employee.id=details.id and designation=’Clerk’;
OR
select name,designation salary from employee E1, details D1 where E1.id=D1.id and designation=’Clerk’;

Naresh	Clerk	32000

Manisha	Clerk	33000

iii. Select employee.id, name,salary from employee, details where employee.id=details.id;
OR
select E1.id, name,salary from employee E1, details D1 where E1.id=D1.id;

101	Naresh	32000
103	Manisha	33000
104	Komal	40000

iv. select name, salary from employee, details where employee.id=details.id and salary<40000;

OR
select name, salary from employee E, details D where E.id=D.id and salary<40000;

Naresh	Clerk	32000
Manisha	Clerk	33000