NAME : BHAVIK RANSUBHE
ROLL NO : 39055
CLASS : TE(B) COMP

ASSIGNMENT 3

CREATING DATABASE :

CREATE database assign3;

use assign3;

CREATING 2 TABLES AND INSERTING VALUES IN THOSE TABLES :

CREATE TABLE employee(

employee_id int(10) NOT NULL AUTO_INCREMENT,

name varchar(50) NOT NULL,

city varchar(50) NOT NULL,

phone varchar(50) DEFAULT NULL,

salary varchar(50) NOT NULL,

PRIMARY KEY (employee_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO employee(name,city,phone,salary) VALUES

('Bhavik','Pune','9090909090','10000'),

('Shahrukh','Kolkata','8080808080','12000'),

('Mukesh','Mumbai','7070707070','9000'),

('vijay','Banglore','6060606060','18000');

SELECT * FROM employee;

	employee_id	name	city	phone	salary
▶	1	Bhavik	Pune	9090909090	10000
	2	Shahrukh	Kolkata	8080808080	12000
	3	Mukesh	Mumbai	7070707070	9000
	4	vijay	Banglore	6060606060	18000
*	NULL	NULL	NULL	NULL	NULL

CREATE TABLE employee2(

employee2_id int(10) NOT NULL AUTO_INCREMENT,

name varchar(50) NOT NULL,

city varchar(50) NOT NULL,

phone varchar(50) DEFAULT NULL,

salary varchar(50) NOT NULL,

PRIMARY KEY (employee2_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO employee2(name,city,phone,salary) VALUES

('kim','LA','5050505050','13000'),

('Bhavik','Pune','9090909090','10000'),

('kylee','LV','4040404040','15000');

SELECT * FROM employee2;

	employee2_id	name	city	phone	salary
▶	1	kim	LA	5050505050	13000
	2	Bhavik	Pune	9090909090	10000
	3	kylee	LV	4040404040	15000
*	NULL	NULL	NULL	NULL	NULL

****

USE OF Update & Delete COMMANDS :

INSERT INTO employee2(name,city,phone,salary) VALUES

('Justin','New York','303030303030','17000');

update employee2

SET city = 'Califronia'

where employee2_id = 4;

select * from employee2;

employee2_id	name	city	phone	salary
1	kim	LA	5050505050	13000
2	Bhavik	Pune	9090909090	10000
3	kylee	LV	4040404040	15000
4	Justin	Califronia	303030303030	17000
NULL	NULL	NULL	NULL	NULL

DELETE FROM employee2 WHERE employee2_id=4;

employee2_id	name	city	phone	salary
1	kim	LA	5050505050	13000
2	Bhavik	Pune	9090909090	10000
3	kylee	LV	4040404040	15000
NULL	NULL	NULL	NULL	NULL

****

USE OF ALL Operators :

SELECT * FROM employee WHERE SALARY > 13000;

employee_id	name	city	phone	salary
4	vijay	Banglore	6060606060	18000
NULL	NULL	NULL	NULL	NULL

SELECT * FROM employee WHERE SALARY < 13000;

employee_id	name	city	phone	salary
1	Bhavik	Pune	9090909090	10000
2	Shahrukh	Kolkata	8080808080	12000
3	Mukesh	Mumbai	7070707070	9000
NULL	NULL	NULL	NULL	NULL

SELECT * FROM employee WHERE SALARY = 12000;

employee_id	name	city	phone	salary
2	Shahrukh	Kolkata	8080808080	12000
NULL	NULL	NULL	NULL	NULL

SELECT * FROM employee WHERE SALARY != 12000;

employee_id	name	city	phone	salary
1	Bhavik	Pune	9090909090	10000
3	Mukesh	Mumbai	7070707070	9000
4	vijay	Banglore	6060606060	18000
NULL	NULL	NULL	NULL	NULL

SELECT * FROM employee WHERE SALARY <> 10000;

employee_id	name	city	phone	salary
2	Shahrukh	Kolkata	8080808080	12000
3	Mukesh	Mumbai	7070707070	9000
4	vijay	Banglore	6060606060	18000
NULL	NULL	NULL	NULL	NULL

SELECT name,salary FROM employee WHERE

salary > 12000 AND city = 'Banglore';

name	salary
vijay	18000

SELECT name,salary FROM employee WHERE

salary > 15000 OR city = 'Pune';

name	salary
Bhavik	10000
vijay	18000

-- like operator

SELECT * FROM employee WHERE name LIKE 'B%' ;

	employee_id	name	city	phone	salary
▶	1	Bhavik	Pune	9090909090	10000
＊	NULL	NULL	NULL	NULL	NULL

-- distinct operator

SELECT distinct(city) from employee ;

	city
▶	Pune
	Kolkata
	Mumbai
	Banglore

-- IN operator

SELECT * from employee WHERE city IN('Pune','Mumbai');

	employee_id	name	city	phone	salary
▶	1	Bhavik	Pune	9090909090	10000
	3	Mukesh	Mumbai	7070707070	9000
＊	NULL	NULL	NULL	NULL	NULL

-- NULL operator

```
91       -- NULL operator
92 ●     SELECT * from employee WHERE salary is null;
93 ●     SELECT * from employee WHERE salary is not null;
```

| Result Grid | Filter Rows: | | Edit: | Export/Im |

	employee_id	name	city	phone	salary
►	1	Bhavik	Pune	9090909090	10000
	2	Shahrukh	Kolkata	8080808080	12000
	3	Mukesh	Mumbai	7070707070	9000
	4	vijay	Banglore	6060606060	18000
*	NULL	NULL	NULL	NULL	NULL

****

SET OPERATIONS :

SELECT name,salary FROM employee

UNION

SELECT name,salary FROM employee2;

	name	salary
►	Bhavik	10000
	Shahrukh	12000
	Mukesh	9000
	vijay	18000
	kim	13000
	kylee	15000

SELECT name,salary FROM employee

UNION ALL

SELECT name,salary FROM employee2;

name	salary
▶ Bhavik	10000
Shahrukh	12000
Mukesh	9000
vijay	18000
kim	13000
Bhavik	10000
kylee	15000

// INTERSECT AND MINUS IS NOT SUPPORTED BY MySQL , Here is Syntax for both operations

SELECT name,salary FROM employee

INTERSECT

SELECT name,salary FROM employee2;

Output will look like this :

name	salary
▶ Bhavik	10000

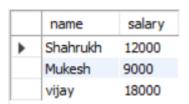--

SELECT name,salary FROM employee

MINUS

SELECT name,salary FROM employee2;

Output will look like this :

name	salary
▶ Shahrukh	12000
Mukesh	9000
vijay	18000

****

/* FUNCTIONS */

SELECT **MAX**(salary) FROM employee2;

	MAX(salary)
▶	15000

SELECT **MIN**(salary) FROM employee2;

	MIN(salary)
▶	10000

SELECT **COUNT**(employee_id) FROM employee;

	COUNT(employee_id)
▶	4

SELECT **AVG**(salary) FROM employee;

	AVG(salary)
▶	12250

SELECT **SUM**(salary)FROM employee;

	SUM(salary)
▶	49000

****

USE OF Clauses :

SELECT * FROM employee2 **ORDER BY** salary **DESC**;

employee2_id	name	city	phone	salary
3	kylee	LV	4040404040	15000
1	kim	LA	5050505050	13000
2	Bhavik	Pune	9090909090	10000
NULL	NULL	NULL	NULL	NULL

SELECT * FROM employee **ORDER BY** name **ASC**;

employee_id	name	city	phone	salary
1	Bhavik	Pune	9090909090	10000
3	Mukesh	Mumbai	7070707070	9000
2	Shahrukh	Kolkata	8080808080	12000
4	vijay	Banglore	6060606060	18000
NULL	NULL	NULL	NULL	NULL

-- distinct clause

SELECT **distinct**(city) from employee ;

city
Pune
Kolkata
Mumbai
Banglore

--HAVING Clause

select name,phone from employee **group by** employee_id **having** employee_id in (1,2);

name	phone
Bhavik	9090909090
Shahrukh	8080808080

****
****