Dbms assignment

Ans 1

```
create database student;
use student;
create table stud(rollno int primary key auto_increment, name varchar (30) , branch varchar (30));
insert into stud(name , branch)values("jay","computer branch"),("suhani","electronic and com"),("kriti","electronic and com");
select* from stud;

create table exam (rollno int, foreign key (rollno) references stud (rollno), s_code text ,marks int, p_code text );
```

Result Grid

rollno	name	branch
1	jay	computer branch
2	suhani	electronic and com
3	kriti	electronic and com
NULL	NULL	NULL

Ans 1.1

```
create table exam (rollno int, foreign key (rollno) references stud (rollno), s_code text ,marks int, p_code text );
insert into exam (rollno,s_code,marks,p_code)values(1,"cs11","50","cs"),(1,"cs12","60","cs"),(2,"ec101","66","ec"),(2,"ec102","70","ec"),(3,"ec101","45","ec"),(3,"ec102","50","
select* from exam;
```

Result Grid

rollno	s_code	marks	p_code
1	cs11	50	cs
1	cs12	60	cs
2	ec101	66	ec
2	ec102	70	ec
3	ec101	45	ec
3	ec102	50	ec

Ans 2

```
use student;
create table employee(employee_id int primary key auto_increment,first_name varchar(30),last_name varchar(30),salary int,joining_date datetime,department varchar(30) );
insert into employee(first_name , last_name , salary , joining_date , department)values
("john","abraham","1000000","2013-01-01 12:00:00","banking"),
("michel","clarke","800000","2013-01-01 12:00:00","insurence"),
("roy","thomas","700000","2013-02-01 12:00:00","banking"),
("tom","jose","600000","2013-02-01 12:00:00","insurence"),
```

Result Grid

employee_id	first_name	last_name	salary	joining_date	department
1	john	abraham	1000000	2013-01-01 12:00:00	banking
2	john	abraham	1000000	2013-01-01 12:00:00	banking
3	michel	clarke	800000	2013-01-01 12:00:00	insurence
4	roy	thomas	700000	2013-02-01 12:00:00	banking
5	tom	jose	600000	2013-02-01 12:00:00	insurence
6	jerry	pinto	650000	2013-02-01 12:00:00	insurence
7	phillip	mathew	750000	2013-01-01 12:00:00	services
8	testname1	kucch bhi	600000	2013-01-01 12:00:00	services
9	testname2	123	650000	2013-01-01 12:00:00	insurence
NULL	NULL	NULL	NULL	NULL	NULL

Ans 2.2

```
15 •  create table incentive(employee_ref_id int ,incentive_amount int,incentive_date datetime );
16 •  insert into incentive(employee_ref_id , incentive_date , incentive_amount)values
17    (1,'2013-02-01',5000),
18    (2,'2013-02-01',3000),
19    (3,'2013-01-01',4000),
20    (1,'2013-01-01',4500),
21    (2,'2013-01-01',3500);
```

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

employee_ref_id	incentive_amount	incentive_date
1	5000	2013-02-01 00:00:00
2	3000	2013-02-01 00:00:00
3	4000	2013-01-01 00:00:00
1	4500	2013-01-01 00:00:00
2	3500	2013-01-01 00:00:00

Ans 3

```
24 •  select first_name as employee from employee where first_name = 'tom';
```

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

employee
tom

Ans 4

```
26 •  select first_name, salary , joining_date from employee;
```

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

first_name	salary	joining_date
john	1000000	2013-01-01 12:00:00
michel	800000	2013-01-01 12:00:00
roy	700000	2013-02-01 12:00:00
tom	600000	2013-02-01 12:00:00
jerry	650000	2013-02-01 12:00:00
phillip	750000	2013-01-01 12:00:00
testname1	600000	2013-01-01 12:00:00
testname2	650000	2013-01-01 12:00:00

Ans 5

```
28 •  SELECT * FROM Employee ORDER BY First_Name ASC;
```

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

employee_id	first_name	last_name	salary	joining_date	department
5	jerry	pinto	650000	2013-02-01 12:00:00	insurence
1	john	abraham	1000000	2013-01-01 12:00:00	banking
2	michel	clarke	800000	2013-01-01 12:00:00	insurence
6	phillip	mathew	750000	2013-01-01 12:00:00	services
3	roy	thomas	700000	2013-02-01 12:00:00	banking
7	testname1	kucch bhi	600000	2013-01-01 12:00:00	services
8	testname2	123	650000	2013-01-01 12:00:00	insurence
4	tom	jose	600000	2013-02-01 12:00:00	insurence
NULL	NULL	NULL	NULL	NULL	NULL

Ans 6

```
30 •  select * from employee where first_name like 'j%'
```

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

employee_id	first_name	last_name	salary	joining_date	department
1	john	abraham	1000000	2013-01-01 12:00:00	banking
5	jerry	pinto	650000	2013-02-01 12:00:00	insurence
NULL	NULL	NULL	NULL	NULL	NULL

Ans 7 & 8

```
34 •  select Department, max(Salary) as Max_salary from employee
35    group by Department
36    order by Max_salary asc;
```

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

Department	Max_salary
services	750000
insurence	800000
banking	1000000

1000000

Ans 9

```
38 •  SELECT e.First_Name, i.Incentive_amount
39    FROM Employee e
40    JOIN Incentive i ON e.Employee_id = i.Employee_ref_id
41    WHERE i.Incentive_amount > 3000;
```

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

First_Name	Incentive_amount
john	5000
roy	4000
john	4500
michel	3500

Ans 10

```
43    create trigger demotable
44    after insert
45    on employee for each row
46    begin
47    INSERT INTO Employee (First_Name, Last_Name, Salary, Joining_date, Department, record)
48    VALUES (NEW.First_Name, NEW.Last_Name, NEW.Salary, NEW.Joining_date, NEW.Department)
49    end//
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	salary	joining_date	department
1	john	abraham	1000000	2013-01-01 12:00:00	banking
2	michel	clarke	800000	2013-01-01 12:00:00	insurence
3	roy	thomas	700000	2013-02-01 12:00:00	banking
4	tom	jose	600000	2013-02-01 12:00:00	insurence
5	jerry	pinto	650000	2013-02-01 12:00:00	insurence
6	phillip	mathew	750000	2013-01-01 12:00:00	services
7	testname1	kucch bhi	600000	2013-01-01 12:00:00	services
8	testname2	123	650000	2013-01-01 12:00:00	insurence
NULL	NULL	NULL	NULL	NULL	NULL

Ans 11

```
create database pro3;
use pro3;
create table salesperson(
SNo int primary key,
SNAME varchar(30) not null,
CITY varchar(30) not null,
COMM float not null);
```

Result Grid | Filter Rows:

SNo	SNAME	CITY	COMM
1001	PEEL	LONDON	0.12
1002	SERRES	SAN JOSE	0.13
1003	AXELROD	NEW YORK	0.1
1004	MOTIKA	LONDON	0.11
1005	RAFKIN	BARCELONA	0.15
NULL	NULL	NULL	NULL

Ans 12

```
CREATE TABLE CUSTOMER (
COMM int primary key auto_increment,
CNAME varchar(30) NOT NULL,
CITY varchar(30) NOT NULL,
RATING INT,
SNo int);
insert into CUSTOMER (COMM,CNAME,CITY,RATING,SNo)
```

Result Grid | Filter Rows: | Edit:

COMM	CNAME	CITY	RATING	SNo
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	san jose	300	1002
204	grass	Barselona	100	1002
206	Clemens	London	300	1007
207	Pereira	Roe	100	1004
NULL	NULL	NULL	NULL	NULL

Ans `13

```
select max(RATING) from CUSTOMER where RATING > 1000;
```

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

max(RATING)
NULL

Ans 14

```
33 • select SNAME, city from Salesperson where city = 'London' AND comm > 0.12;
```

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

SNAME	city

Ans 15

```
35 • select SNAME, city from Salesperson where city = 'London' OR city = 'Barcelona';
```

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

SNAME	city
PEEL	LONDON
MOTIKA	LONDON
RAFKIN	BARCELONA

Ans 16

```
37 • select SNAME, COMM from Salesperson where COMM > 0.10 AND COMM < 0.12;
```

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

SNAME	COMM
PEEL	0.12
AXELROD	0.1
MOTIKA	0.11

Ans 17

```
39 • select CNAME, RATING, city from CUSTOMER where city = 'Roe' or RATING <= 100;
```

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

CNAME	RATING	city
Hoffman	100	London
Giovanne	200	Roe
grass	100	Barselona
Pereira	100	Roe

Ans 18

```
1  create database pro4;
2  use pro4;
3  create table salesman (
4  salesman_id int primary key,
5  name varchar(30) not null,
6  CITY varchar(30) not null,
7  commission float not null);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Co

salesman_id	name	CITY	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5003	Lauson Hen	San Jose	0.12
5005	Pit Alex	LONDON	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
NULL	NULL	NULL	NULL

Ans 19

```
17 CREATE TABLE orders (
18 ord_no int primary key not null,
19 purch_amt float NOT NULL,
20 ord_date date NOT NULL,
21 customer_id INT,
22 salesman_id int ,
23 foreign key (salesman_id) references salesman (salesman_id));
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70002	65.26	2012-10-05	3002	5001
70003	2480.4	2012-10-10	3009	5003
70004	110.5	2012-08-17	3009	5003
70005	2400.6	2012-07-27	3007	5001
70007	948.5	2012-09-10	3005	5002
70008	5760	2012-09-10	3002	5001
70009	270.65	2012-09-10	3001	5005
70010	1983.43	2012-10-10	3004	5006
70011	75.29	2012-08-17	3003	5007
70012	250.45	2012-06-27	3008	5002
70013	3045.6	2012-04-25	3002	5001
NULL	NULL	NULL	NULL	NULL

Ans 19.1

```
39 • select salesman_id, ord_no, purch_amt from orders where salesman_id = 5001;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content

salesman_id	ord_no	purch_amt
5001	70002	65.26
5001	70005	2400.6
5001	70008	5760
5001	70013	3045.6
NULL	NULL	NULL

Ans 20

```
1  create database pro5;
2 • use pro5;
3 • create table item_mast (
4  PRO_ID int primary key auto_increment,
5  PRO_NAME varchar(30) not null,
6  PRO_PRICE float not null,
7  PRO_COM int not null);
```

Result Grid | Filter Rows: | Edit:

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
1	Mother Board	3200	15
2	Key Board	450	16
3	ZIP drive	250	14
4	Speaker	550	16
5	Monitor	5000	11
6	DVD drive	900	12
7	CD drive	800	12
8	Printer	2600	13
9	Refill cartridge	350	13
10	Mouse	250	12
NULL	NULL	NULL	NULL

Ans 20.1

```
22 • select PRO_NAME, PRO_PRICE from item_mast where PRO_PRICE > 200 and PRO_PRICE < 600;
```

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

PRO_NAME	PRO_PRICE
Key Board	450
ZIP drive	250
Speaker	550
Refill cartridge	350
Mouse	250

Ans 21

```
24 •  select avg (PRO_PRICE) as avg_price from item_mast where PRO_COM = 16;
```

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

avg_price
500

Ans 22

```
26 •  select PRO_NAME as 'Item Name', PRO_PRICE as 'Price in Rs.' from item_mast;
```

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

Item Name	Price in Rs.
Mother Board	3200
Key Board	450
ZIP drive	250
Speaker	550
Monitor	5000
DVD drive	900
CD drive	800
Printer	2600
Refill cartridge	350
Mouse	250

Ans 23

```
28 •  select PRO_NAME, PRO_PRICE from item_mast where PRO_PRICE >= 250;
```

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

PRO_NAME	PRO_PRICE
Mother Board	3200
Key Board	450
ZIP drive	250
Speaker	550
Monitor	5000
DVD drive	900
CD drive	800
Printer	2600
Refill cartridge	350
Mouse	250

Ans 23.1

```
30 •  select PRO_NAME, PRO_PRICE from item_mast order by PRO_PRICE desc;
```

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

PRO_NAME	PRO_PRICE
Monitor	5000
Mother Board	3200
Printer	2600
DVD drive	900
CD drive	800
Speaker	550
Key Board	450
Refill cartridge	350
ZIP drive	250
Mouse	250

Ans 23.2

```
32 •  select PRO_NAME, PRO_PRICE from item_mast order by PRO_NAME asc;
```

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

PRO_NAME	PRO_PRICE
CD drive	800
DVD drive	900
Key Board	450
Monitor	5000
Mother Board	3200
Mouse	250
Printer	2600
Refill cartridge	350
Speaker	550
ZIP drive	250

Ans 24

```
34 •  select PRO_COM as compney_cod, avg(PRO_PRICE) as average_price
35    from item_mast
36    group by PRO_COM;
```

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

compney_cod	average_price
15	3200
16	500
14	250
11	5000
12	650
13	1475