

DBMS

Assignment-1

1. Write a query to create a table employee with empno, ename, designation and salary.

```
create table employeeega(empno number,ename varchar(10),designation varchar(10),salary number);
```

2. Write a query to display the column name and data type of the table employee

```
desc employeeega;
```

3. Write a query to create a table from an existing table with all the field

```
create table emp as select * from employeeega;
```

4. Write a query to create table from an existing table with selected fields

```
create table emp1 as select empno, salary from employeeega;
```

5. Write a query to create a new table from an existing table without any record

```
create table emp2 as select * from employeeega where 1=2;
```

6. Write a query to Alter the column empno number(4) to empno number(6).

```
alter table employeeega modify empno number(6);
```

7. Write a query to Alter the table employee with multiple columns (empno, ename).

```
alter table employeeega modify (empno number(10), ename varchar2(15));
```

8. Write a query to add a new column in employee table.

```
alter table employeeega add DOJ date;
```

9. Write a query to add multiple columns in employee table

```
alter table employeeega add (DOB date, age number(2));
```

10. Write a query to drop a column from an existing table employee.

```
alter table employeeega drop column age;
```

11. Write a query to drop multiple columns from the employee table.

```
alter table employeeega drop (ename, salary);
```

12. Write a query to rename table employee to emp.

```
alter table employee rename to emp;
```

Assignment-2

1. Create a table employee with attributes emp_id, f_name, l_name, job_type, salary, dept, commission, manager_id.

```
create table employeee(  
  emp_id number(5),  
  f_name varchar2(10),  
  l_name varchar2(10),  
  job_type varchar2(10),  
  salary number(10),  
  dept varchar2(10),  
  commission number(10),  
  manager_id number(10));
```

2. Make emp_id as the primary key of employee table.

```
alter table employeee  
add constraint employeee_pk primary key (emp_id);
```

3. Make f_name and salary NOT NULL type.

```
alter table employeee modify f_name varchar2(10) NOT NULL;  
alter table employeee modify salary number(10) NOT NULL;
```

4. Add a column date_of_joining in the employee table.

```
alter table employeee add date_of_joining date;
```

5. Create a table department with attribute d_name, d_loc and HOD_id where d_name is primary key.

```
create table department(  
  d_name varchar2(10) primary key,  
  d_loc varchar2(10),  
  HOD_id number(5));
```

6. Create a table location with attributes loc_id, city and contact_no.

```
create table location(  
  loc_id number(5),  
  city number(10),  
  contact_no number(10));
```

7. Enhance the size of the 'city' attribute by 5, in the location table.

```
alter table location modify city number(15);
```

8. Delete the contact_no attribute from the location table.

```
alter table location drop column contact_no;
```

9. Make the department attribute of the employee table its foreign key referencing the department table.

```
alter table employee add constraint employee_to_dept_fk foreign key (dept) references department;
```

10. Rename the city attribute to 'address' in the location table.

```
alter table location rename column city to address;
```

11. Rename the location table name to 'loc'.

```
alter table location rename to loc;
```

12. Insert the following rows in 'loc' table

loc_id	address
1	Kolkata
2	Mumbai

```
insert into loc (loc_id,address) values(1,'Kolkata');  
insert into loc (loc_id,address) values(2,'Mumbai');
```

13. Truncate the table 'loc'.

```
Truncate table loc;
```

14. Drop the table 'loc'

```
drop table loc;
```

15. Insert the following rows in the department table:

d_name	d_loc	Hod_id
Sales	Kol	4
Accounts	Delhi	6
Production	Kol	1
Marketing	Kol	2
R&D	Marketing	8

```
insert into department (d_name,d_loc,HOD_id) values('Sales','Kol',4);  
insert into department (d_name,d_loc,HOD_id) values('Accounts','Delhi',6);  
insert into department (d_name,d_loc,HOD_id) values('Production','Kol',1);  
insert into department (d_name,d_loc,HOD_id) values('Marketing','Kol',2);  
insert into department (d_name,d_loc,HOD_id) values('R&D','Marketing',8);
```

16. Insert the following rows in the employee table

Emp_id	Fname	Lname	Job_Type	Salary	Commission	Dept	Manager_id	DOJ
1	Arun	Khan	Manager	90000		Production		04-Jan-1998
2	Barun	Kumar	Manager	80000		Marketing		09-Feb-1998 ^{Sunday 02}
3	Chitra	Kapoor	Engineer	60000		Production	1	08-Jan-1998
4	Dheeraj	Mishra	Manager	75000		Sales	4	27-Dec-2001
5	Emma	Dutt	Engineer	55000		Production	1	20-Mar-2002
6	Floki	Dutt	Accountant	70000		Accounts		16-Jul-2000
7	Deeraj	Kumar	Clerk	40000		Accounts	6	01-Jul-2016
8	Saul	Good	Engineer	60000		R&D		06-Sep-2014
9	Mou	Bhat	Clerk	30000		Sales	4	08-Mar-2018
10	Sunny	Deol	Salesman	20000	10000	Marketing	2	31-Mar-2001
11	Bobby	Deol	Engineer	35000		R&D	8	17-Oct-2017
12	Aamir	Khan	Salesman	15000	5000	Marketing	2	11-Jan-2013

```

alter table employee rename column date_of_joining to doj;
alter table employee drop constraint employee_pk drop constraint EMPLOYEE_TO_DEPT_FK;
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj) values(1,'Arun','Khan','Manager',90000,'Production','04-JAN-1998');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj) values(2,'Barun','Kumar','Manager',80000,'Marketing','09-FEB-1998');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj) values(3,'Chitra','Kapoor','Engineer',60000,'Production',1,08-JAN-1998);
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj) values(4,'Deeraj','Mishra','Manager',75000,'Sales',4,27-DEC-2001);
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj) values(5,'Emma','Dutta','Engineer',55000,'Production',1,20-MAR-2002);
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj) values(6,'Chinaswami','Iyer','Accounts',70000,'Accounts','16-JUL-2000');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj) values(7,'Deeraj','Kumar','Clerk',40000,'Accounts',6,01-JUL-2016);
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj) values(8,'Mr.','Paul','Engineer',60000,'RandD','06-SEP-2014');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj) values(9,'Raj','Mishra','Clerk',30000,'Sales',4,08-MAR-2018);
insert into employee(emp_id,f_name,l_name,job_type,salary,commission,dept,manager_id,doj) values(10,'Sunny','Deol','Salesman',20000,10000,'Marketing',2,31-MAR-2001);
insert into employee(emp_id,f_name,l_name,job_type,salary,commission,dept,manager_id,doj) values(11,'Bobby','Deol','Engineer',35000,5000,'R&D',8,17-OCT-2017);
insert into employee(emp_id,f_name,l_name,job_type,salary,commission,dept,manager_id,doj) values(12,'Choton','Khan','Salesman',15000,5000,'Marketing',2,11-JAN-2013);

```

17. Show the values of departmental table

```
select * from department;
```

18. Select the department names and their locations.

```
select d_name, d_loc from department;
```

19. Show the employees f_name, l_name, salary and the salary after 1000rs. Bonus.

```
select f_name,l_name,salary, salary+1000, commission from employee;
```

20. Show the employees annual salary with a 1000rs. Yearly bonus and the annual salary with a 100rs. Monthly bonus.

```
select salary*12+1000,commission*12, salary*12+100 from employee;
```

21. Show f_name as NAME and annual salary as ANNSAL from the employee table

```
select f_name as name,salary as annsal from employee;
```

22. Show the l_name as LasT AND 100rs. Incremented salary as NewSal.

```
select l_name as LasT, salary+100 as NewSal from employee;
```

23. Show the emp_id, f_name, l_name, job_type of the employee getting highest salary.

```
select emp_id, f_name, l_name, job_type from employee where salary=(select max(salary) from employee);
```

24. Show the emp_id, f_name, l_name, job_type of the employee getting minimum salary.

```
select emp_id, f_name, l_name, job_type from employee where salary=(select min(salary) from employee);
```

25. Show the average salary of employees in the employee table.

```
select avg(salary) from employee;
```

26. Consider the Insurance database given below. The primary keys are underlined and the data types are specified:

PERSON (driver-id: string, name: string, address: string)

CAR (Regno:string,model:string,year:int)

ACCIDENT (report-number:int,date:date,location:string)

OWNS (driver-id:string,regno:string)

PARTICIPATED (driver-id:string,regno:string,report-number:int,damage-amount:int)

i. Create the above tables by properly specifying the primary keys and the foreign keys

ii. Enter atleast five tuples for each relation

iii. Demonstrate how you a. Update the damage amount for the car with a specific regno in accident with report number 12 to 25000 b. Add a new accident to the database

iv. Find the total number of people who owned cars that were involved in accidents in 2006.

v. Find the number of accidents in which cars belonging to a specific model were involved.

```
SQL> create table person(driver_id varchar(10),name varchar(10),address varchar(10),primary key(driver_id));
```

```
SQL> create table car(regno varchar(10),model varchar(10),year int,primary key(regno));
```

```
SQL> create table accident(report_number int,accd_date date,location varchar(10),primary key(report_number));
```

```
SQL> create table owns(driver_id varchar(10),regno varchar(10),primary key(driver_id,regno),foreign key(driver_id) references person(driver_id));
```

```
SQL> create table participated(driver_id varchar(10),regno varchar(10),report_number int,damage_amount int,primary key(driver_id,regno,report_number));
```

Assignment- 3

1. Consider the following employee table and execute the queries based on it

Emp_id	Fname	Lname	Job_Type	Salary	Commission	Dept	Manager_id	DOJ
1	Arun	Khan	Manager	90000		Production		04-Jan-1998
2	Barun	Kumar	Manager	80000		Marketing		09-Feb-1998
3	Chitra	Kapoor	Engineer	60000		Production	1	08-Jan-1998
4	Dheeraj	Mishra	Manager	75000		Sales	4	27-Dec-2001
5	Emma	Dutt	Engineer	55000		Production	1	20-Mar-2002
6	Floki	Dutt	Accountant	70000		Accounts		16-Jul-2000
7	Dheeraj	Kumar	Clerk	40000		Accounts	6	01-Jul-2016
8	Saul	Good	Engineer	60000		R&D		06-Sep-2014
9	Mou	Bhat	Clerk	30000		Sales	4	08-Mar-2018
10	Sunny	Deol	Salesman	20000	10000	Marketing	2	31-Mar-2001
11	Bobby	Deol	Engineer	35000		R&D	8	17-Oct-2017
12	Amir	Khan	Salesman	15000	5000	Marketing	2	11-Jan-2013

1. Show f_name, l_name and job_type from employees.

```
select f_name, l_name, Job_type from employee;
```

2. Show employee details in the following fashion:

Employee details
Arun is a manager

```
select 'Employee details' || chr(10) || f_name || ' is a manager' from employee;
```

3. Show the monthly salary details in the following fashion

Monthly Salary Details
Arun's monthly salary is Rs. 90000

```
select 'Monthly Salary Details' || chr(10) || f_name || 's monthly salary is Rs. ' || salary from employee;
```

Consider the Department table to answer the queries

d_name	d_loc	Hod_id
Sales	Kol	4
Accounts	Delhi	6
Production	Kol	1
Marketing	Kol	2
R&D	Marketing	8

4. Show the different department names from department table

```
select d_name from Department;
```

5. Show the employee names who works in 'Sales'

```
select f_name, l_name from Employeee where dept='Sales';
```

6. Show the employee names who gets salary of more than 50000 per month

```
select f_name from employeee where salary>50000;
```

7. Show the details of the employee whose manager id is not 1

```
select * from employeee where manager_id!=1;
```

8. Show the employee details whose salary ranges between 40000 and 70000

```
select * from employeee where salary>40000 and salary<70000;
```

9. Show the details of the employees who works under the manager having id 1, 6 and 8

```
select * from employeee where manager_id in (1,6,8);
```

10. Select the f_name and salary of those employees whose last name starts with 'K'

```
select f_name, salary from employeee where l_name like 'K%';
```

11. Select the f_name and salary of those employees whose last name starts with 'K' and ends with 'R'

```
select f_name, salary from employeee where l_name like 'K%' and l_name like '%r';
```

12. Show the details of those employees where 3rd letter of l_name is 'o'

```
select * from employeee where l_name like '___o%';
```

13. Select the details of those employees who works as an engineer with monthly salary more than 50000;

```
select * from employeee where job_type='Engineer' and salary>50000;
```

14. Select the employees whose department is 'Production' or monthly salary is more than 60000 per month.

```
select * from employeee where dept='Production' or Salary>60000;
```

15. Find the minimum salary, maximum salary, total salary, average salary of the employees who work in 'Sales' department.

```
select min(salary),max(salary),sum(salary),avg(salary) from employeee where job_type='Engineer';
```

16. Find the employee l_name that is first and f_name that is last if they are arranged in an order

```
select f_name, l_name from employee order by f_name ASC;
```

17. Find the number of employees working in each department

```
select dept, count(dept) from employee group by dept;
```

18. Find the number of departments from employee table

```
select count(dept) from employee;
```

19. Find the average commission of the employees

```
select avg(commission) from employee;
```

20. Find the average salaries of the employees department wise

```
select avg(salary), dept from employee group by dept;
```

21. Find the sum of salary of different job_type according to different departments

```
select sum(salary), job_type from employee group by job_type;
```

22. Find the department name and average salaries of those departments whose average salary is greater than 40000.

```
select dept, avg(salary) from employee group by dept having avg(salary)>40000;
```

23. Find the department name and maximum salaries of those departments whose maximum salary is greater than 55000

```
select dept, max(salary) from employee group by dept having max(salary)>55000;
```

24. Display the job_type and total monthly salary for each job_type where total payroll is exceeding 100000

```
select job_type, sum(salary) from employee group by job_type having sum(salary)>100000;
```

25. Display the name of the department having maximum average salary

```
select f_name, l_name from employee group by department having max(avg(salary));
```

Assignment 4

1. Show the use of upper and lower function


```
select upper(f_name), lower(l_name) from employee;
```

2. Show the use of concat, instr and length function

```
select concat ('hi ',f_name) as Greet from employee;  
SELECT INSTR('THIS IS THE THING','TH') "Position Found" FROM DUAL;  
SELECT f_name, LENGTH(f_name) "Length in characters" from employee;
```

3.

1. SQRT

```
select sqrt(100) SQUARE from dual;
```

2. POWER

```
select power(2,3) power from dual;
```

3. CEIL

```
select ceil(213.4214) from dual;
```

4. Substr

```
select substr('aforapple',2,5) from dual;
```

5. MAX

```
select max(salary) from employee;
```

6. MIN

```
select min(salary) from employee;
```

7. Round

```
select round(13.523) LOL from dual;
```

8. AVG

```
select avg(salary) from employee;
```

9. COUNT

```
select count(job_type) from employee;
```

10. Exp

```
select exp(100) from dual;
```

11. Mod

```
select mod(11,3) from dual;
```

4. Solve the following queries

- Find the ceiling and floor value of 14.887.
- Find out the round-off 17.49989.
- Calculate 8^7

```
select ceil(14.887), floor(14.887) from dual;  
select round(17.49989) from dual;  
select power(8,7) from dual;
```

5. Show the current date

```
select CURRENT_DATE from dual;
```

6. Find the total experience of the employees in weeks who works in Sales department

```
select f_name, abs((sysdate-doj))/7 wks_between from employee;
```

7. Display the use of the following functions on date

- Months_between
- Add_months
- Next_day
- Last_day
- Round
- Trunc
- To_char

```
SELECT MONTHS_BETWEEN (TO_DATE('02-02-1995', 'MM-DD-YYYY'), TO_DATE('01-01-1995', 'MM-DD-YYYY') ) "Months"  
FROM DUAL;
```

```
SELECT TO_CHAR(ADD_MONTHS(hire_date, 1), 'DD-MON-YYYY') "Next month"  
FROM employees
```

```
SELECT NEXT_DAY('12-MAR-2015', 'MONDAY') "NEXT DAY" FROM DUAL;
```

```
SELECT LAST_DAY('12-MAR-2015') "LAST DAY" FROM DUAL;
```

```
select ROUND(TO_DATE('22-NOV-18'), 'DAY') from dual ;
```

```
select TRUNC(TO_DATE('25-NOV-18'), 'YEAR') from dual;
```

```
SELECT TO_CHAR(12345.67, '99999.9') FROM DUAL;
```

8. Show the employee details with a revised salary. The salary is incremented in the following way:
- a. 10% for sales department
 - b. 20% for marketing department
 - c. No increment for others

```
update employee set salary=salary+(salary*10/100) where dept='Sales';
```

```
update employee set salary=salary+(salary*10/100) where dept='Marketing';
```

9. Determine the tax for each employee in production department based on the monthly salary. The tax rate are as per the following data:

Monthly Salary Range	Rate
0 – 19,999	0%
20,000 – 39,999	9%
40,000 – 59,999	20%
60,000 – 79,999	30%
80,000 or more	45%

10. Find the Cartesian product between Employee and Department table.

```
select * from employee cross join department;
```

11. Show the employee names and the respective department location

```
select f_name, d_loc from employee E cross join department D where E.DEPT=D.D_NAME;
```

12. Give an example of the following joins considering employee and department tables.
- k. Natural join
 - l. Inner join
 - m. Left outer join
 - n. Right outer join
 - o. Full outer join

```
select salary, d_loc from employee Natural Join department;
```

```
select job_type, d_loc from employee inner join department on employee.dept=department.d_name;
```

```
select job_type, doj, hod_id from employee e left outer join department d on e.dept=d.d_name;
```

```
select job_type, doj, hod_id from employee e right outer join department d on e.dept=d.d_name;
```

```
select job_type, doj, hod_id from employee e full outer join department d on e.dept=d.d_name;
```

1. Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments. [Go to the editor](#)
Hint : Use NATURAL JOIN.

Sample table: locations

location_id	street_address	postal_code	city	state_province	country_id
1000	1297 Via Cola di Rie	989	Roma		IT
1100	93091 Calle della Te	10934	Venice		IT
1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefectu	JP
1300	9450 Kamiya-cho	6823	Hiroshima		JP
1400	2014 Jabbawocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San	California	US
1600	2007 Zagora St	50090	South Brun	New Jersey	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
1800	147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA

Sample table: countries

country_id	country_name	region_id
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2
CH	Switzerland	1
CN	China	3
DE	Germany	1

```
select location_id, street_address, city, state_province, country_name from locations natural join countries;
```

14. 2. Write a query to find the name (first_name, last name), department ID and name of all the employees. [Go to the editor](#)

Sample table: employees

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1987-06-18	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	1987-06-19	AD_VP
103	Alexander	Hunold	AHUNOLD	590.423.4567	1987-06-20	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	1987-06-21	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	1987-06-22	IT_PROG
106	Valli	Pataballa	VPATABAL	590.423.4560	1987-06-23	IT_PROG

Sample table: departments

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500

```
select first_name, last_name, department_id from employees join departments;
```

Assignment-5

Emp_id	Fname	Lname	Job_Type	Salary	Commission	Dept	Manager_id	DOJ
1	Asuri	Khan	Manager	90000		Production		04-Jan-1998
2	Esurun	Kumar	Manager	80000		Marketing		09-Feb-1998
3	Chitra	Kapoor	Engineer	60000		Production	1	08-Jan-1998
4	Dheeraj	Mishra	Manager	75000		Sales	4	27-Dec-2001
5	Emma	Dutt	Engineer	55000		Production	1	20-Mar-2002
6	Floki	Dutt	Accounts	70000		Accounts		16-Jul-2000
7	Dheeraj	Kumar	Clerk	40000		Accounts	6	01-Jul-2016
8	Saul	Good	Engineer	60000		R&D		06-Sep-2014
9	Mou	Bhat	Clerk	30000		Sales	4	08-Mar-2018
10	Sunny	Deol	Salesman	20000	10.000	Marketing	2	31-Mar-2001
11	Bobby	Deol	Engineer	35000		R&D	8	17-Oct-2017
12	Amir	Khan	Salesman	15000	5.000	Marketing	2	11-Jan-2013

D_Name	D_Loc	HOD_ID
Sales	Kol	4
Accounts	Delhi	6
Production	Kol	1
Marketing	Kol	2
R&D	Delhi	8

- Find the Cartesian product between Employee and Department table.

```
select * from employee cross join department;
```

- Show the employee names and the respective department location

```
select concat(f_name,l_name) as name from employee natural join department;
```

- Find the employee name and date of joining who are working in Delhi

```
select concat(f_name,l_name) as name,doj from employee natural join department where d_loc='Delhi';
```

- Create a table 'Emp_Address' for storing the permanent address of the employees and insert the following values:

Emp-id	City	District	State
1	Suri	Birbhum	WB
2	Kolkata	Kolkata	WB
3	Bhubaneswar	Khurda	Odisha
4	Durgapur	Burdwan	WB
5	Noida	G.B Nagar	UP
6	Secunderabad	Hyderabad	Telangana
7	Dehradun	Dehradun	Uttarakhand
8	Asansol	Bardwan	WB
9	Siliguri	Darjeeling	WB
10	Kolkata	Kolkata	WB
11	Jalpaiguri	Jalpaiguri	WB
12	New Delhi	New Delhi	Delhi

```
create table Emp_Address(
Emp_id number(2),
City varchar2(10),
District varchar2(10),
State varchar2(10));
insert into Emp_Address (Emp_id, city, District, state) values (1,'Suri', 'Birbhum', 'WB');
insert into Emp_Address (Emp_id, city, District, state) values (2,'Kolkata', 'Kolkata', 'WB');
insert into Emp_Address (Emp_id, city, District, state) values (3,'Bhubaneswar', 'Khurda', 'Odisha');
insert into Emp_Address (Emp_id, city, District, state) values (4,'Durgapur', 'Burdwan', 'WB');
```

```

insert into Emp_Address (Emp_id, city, District, state) values (5,'Noida', 'GB Nagar', 'UP');
insert into Emp_Address (Emp_id, city, District, state) values (6,'Secunderabad', 'Hyderabad', 'Telengana');
insert into Emp_Address (Emp_id, city, District, state) values (7,'Dehra Dun', 'Dehradun', 'Uttarakhand');
insert into Emp_Address (Emp_id, city, District, state) values (8,'Asansol', 'Burdwan', 'WB');
insert into Emp_Address (Emp_id, city, District, state) values (9,'Siliguri', 'Darjeeling', 'WB');
insert into Emp_Address (Emp_id, city, District, state) values (10,'Kolkata', 'Kolkata', 'WB');
insert into Emp_Address (Emp_id, city, District, state) values (11,'Jalpaiguri', 'Jalpaiguri', 'WB');
insert into Emp_Address (Emp_id, city, District, state) values (12,'New Delhi', 'New Delhi', 'Delhi');

```

5. Display the name of employees, department location and the city name the employee belongs to, from the Employee, Department and Emp_Address tables.

```

select concat(f_name, l_name) as Name, d_loc, city from Employee E join Department D on E.dept=D.d_name join Emp_Address A on E.emp_id=A

```

6. Find the name of each department's manager.

```

select dept,concat(f_name, l_name) as name from employee E join Department D on E.dept=D.d_name where job_type='Manager' order by dept;

```

7. Create 'Job_Grades' table and insert the following values:

GRADE	LOWEST_SAL	HIGHEST_SAL
A	10000	24999
B	25000	49999
C	50000	100000

```

create table Job_Grades(
Grade varchar2(3),
Lowest_Sal number(10),
Highest_Sal number(10));
insert into Job_Grades(Grade, Lowest_sal, Highest_Sal) values('A', 10000, 24999);
insert into Job_Grades(Grade, Lowest_sal, Highest_Sal) values('A', 25000, 49999);
insert into Job_Grades(Grade, Lowest_sal, Highest_Sal) values('A', 50000, 100000);

```

Assignment -6

Sample Table – Worker

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
001	Monika	Arora	100000	2014-02-20 09:00:00	HR
002	Niharika	Verma	80000	2014-06-11 09:00:00	Admin
003	Vishal	Singhal	300000	2014-02-20 09:00:00	HR
004	Amitabh	Singh	500000	2014-02-20 09:00:00	Admin
005	Vivek	Bhati	500000	2014-06-11 09:00:00	Admin
006	Vipul	Diwan	200000	2014-06-11 09:00:00	Account
007	Satish	Kumar	75000	2014-01-20 09:00:00	Account
008	Geetika	Chauhan	90000	2014-04-11 09:00:00	Admin

WORKER_REF_ID	BONUS_DATE	BONUS_AMOUNT
1	2016-02-20 00:00:00	5000
2	2016-06-11 00:00:00	3000
3	2016-02-20 00:00:00	4000
1	2016-02-20 00:00:00	4500
2	2016-06-11 00:00:00	3500

Sample Table – Title

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
1	Manager	2016-02-20 00:00:00
2	Executive	2016-06-11 00:00:00
8	Executive	2016-06-11 00:00:00
5	Manager	2016-06-11 00:00:00
4	Asst. Manager	2016-06-11 00:00:00
7	Executive	2016-06-11 00:00:00
6	Lead	2016-06-11 00:00:00
3	Lead	2016-06-11 00:00:00

```
create table worker (worker_id number(10),first_name varchar(20),last_name varchar(10),salary number(10),joining_date date,department v
insert into worker values(001,'Momota','Didi',100000,'02-Jun-2002','HR');
insert into worker values(002,'Modi','Da',70000,'02-Jun-2002','Admin');
insert into worker values(003,'Kakajibonfaka','Khelahobe',300000,'02-Jun-2002','HR');
insert into worker values(004,'Murturwami','Iyer',500000,'02-Jun-2002','Admin');
insert into worker values(005,'Kesta','Da',500000,'02-Jun-2002','Admin');
insert into worker values(006,'Chagla','Da',80000,'02-Jun-2002','Account');
insert into worker values(007,'Pagla','Di',270000,'02-Jun-2002','Account');
insert into worker values(008,'Netaji','Bose',50000,'02-Jun-2002','Admin');
```

1. Write An SQL Query To Fetch "FIRST_NAME" From Worker Table In Upper Case alias as WORKER_FIRSTNAME.

```
select UPPER(first_name) "WORKER_FIRSTNAME" from worker;
```

2. Write An SQL Query To Print The First Three Characters Of FIRST_NAME From Worker Table

```
select substr(first_name,1,3) from worker;
```

3. Write An SQL Query To Find The Position Of The Alphabet ('A') In The First Name Column 'Amitabh' From Worker Table.

```
Select INSTR(FIRST_NAME, 'a') from Worker where FIRST_NAME = 'Amitabh';
```

4. Write An SQL Query To Print The FIRST_NAME And LAST_NAME From Worker Table Into A Single Column COMPLETE_NAME. A Space Char Should Separate Them.

```
select concat(first_name,concat(' ', last_name)) as COMPLETE_NAME from worker;
```

5. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending And DEPARTMENT Descending.

```
select * from worker order by first_name ASC,department DESC;
```

6. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Contains 'A'.

```
select * from worker where first_name like '%A%';
```

7. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With 'A'.

```
select * from worker where first_name like '%A';
```

8. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000.

```
select * from worker where salary>100000 and salary<500000;  
select * from worker where salary between 100000 and 500000;
```

9. Write An SQL Query To Fetch The Count Of Employees Working In The Department 'Admin'.

```
select count(first_name) from Worker where department='Admin';
```

10. Write An SQL Query To Fetch The No. Of Workers For Each Department In The Descending Order

```
select department,count(worker_id) as No_of_workers from worker group by department order by No_of_workers DESC;
```

11. Write An SQL Query To Print Details Of The Workers Who Are Also Managers

```
create table title(Worker_Ref_Id number(10),Worker_Title varchar(20),Affected_from date);  
insert into title values(1,'Manager','02-Jun-2002');  
insert into title values(2,'Executive','02-Jun-2002');  
insert into title values(8,'Executive','02-Jun-2002');  
insert into title values(5,'Manager','02-Jun-2002');  
insert into title values(4,'Asst. Manager','02-Jun-2002');  
insert into title values(7,'Executive','02-Jun-2002');  
insert into title values(6,'Lead','02-Jun-2002');  
insert into title values(3,'Lead','02-Jun-2002');
```

```
select distinct w.first_name,t.worker_title from worker w join title t on t.worker_title in ('Manager');
```

12. Write An SQL Query To Show Only Odd Rows From A Table

```
select * from worker where mod(worker_id,2)<>0;
```

13. Write An SQL Query To Show Records From One Table That Another Table Does Not Have.

```
select * from worker MINUS select * from title;
```

14. Write An SQL Query To Show The Top N (Say 10) Records Of A Table

```
select * from worker where ROWNUM <= 10;
```

15. . Write An SQL Query To Fetch The List Of Employees With The Same Salary.


```
select e1.f_name from employee e1, employee e2 where e1.salary=e2.salary and e1.f_name<>e2.f_name;
```

16. Write An SQL Query To Show All Departments Along With The Number Of People Working There

```
select department,count(department) "number of workers" from worker group by department;
```

17. Write An SQL Query To Print The Name Of Employees Having The Highest Salary In Each Department

```
select e.f_name, e.dept from employee e where e.salary in(select max(salary) from employee group by dept);
```

18. Write An SQL Query To Fetch Departments Along With The Total Salaries Paid For Each Of Them

```
select department,sum(salary) from worker group by department;
```

Assignment-7

Emp_id	Start_date	End_date	Job_type	D_name
1	04-Jan-1998	30-Jun-2001	Engineer	Production
2	09-Feb-1998	28-Feb-2002	Sales man	Sales
1	01-Jul-2001	31-Dec-2010	Manager	R & D
4	27-Dec-2001	19-Sep-2016	Sales executive	Marketing
2	01-Mar-2002	30-Mar-2015	Sales Executive	Marketing
2	01-Apr-2016	15-Dec-2017	Manager	Sales
4	20-Sep-2016	16-Dec-2017	Asst. Asst. Manager	Sales
6	16-Jul-2000	30-Nov-2006	Clerk	Accounts
5	20-Mar-2002	12-Aug-2011	Engineer	R & D
1	01-Jan-2011	31-Jan-2012	Engineer	Production

```
CREATE TABLE job_history (
EMP_ID number(3),
START_DATE date,
END_DATE date,
JOB_Type varchar(10) ,
D_Name varchar(12)
);
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(1, '04-Jan-1998', '30-Jun-2001', 'Engineer', 'Production'
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(1, '09-Feb-1998', '28-Feb-2002', 'Engineer', 'Production'
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(1, '01-Jul-2001', '31-Dec-2010', 'Engineer', 'Production'
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(1, '27-Dec-2001', '19-Sep-2016', 'Engineer', 'Production'
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(1, '01-Mar-2002', '30-Mar-2015', 'Engineer', 'Production'
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(1, '01-Apr-2016', '15-Dec-2017', 'Engineer', 'Production'
insert into job_history(emp_id, start_date, end_date, job_type, d_name) values(2, '04-Apr-2016', '18-Dec-2017', 'Sales', 'Accounts');
```

2. Display the previous and current job_types of all the employees.

```
select emp_id, job_type from job_history order by job_type;
```

3. Display the previous and current department and job_types of all the employees.

```
select emp_id, d_name, job_type from job_history order by job_type;
```

4. Display the employee id and job_types of the employees who currently have a job title that they held previously.

5. Find the name of those employees who have not changed their jobs once

6. Find the names of the employees who earn more than Chitra

```
select f_name from employee where salary>(select salary from employee where f_name='Chitra');
```

7. Find the details of those employees who have the same job_type as of emp_id 7.

```
select * from employee where job_type=(select job_type from employee where emp_id=7) and emp_id<>7;
```

8. Find the details of the employees whose job_type is same as that of emp_id 3 and whose salary is greater than that of emp_id 7

```
select * from employee where job_type=(select job_type from employee where emp_id=3) and emp_id<>3 and salary>(select salary from emp
```

9. Display l_name, job_type and the salary of the employees whose salary is equal to the minimum salary.

```
select l_name, job_type, salary from employee where salary=(select min(salary) from employee);
```

10. Find the job_type with lowest average salary.

11. Display all the departments that have a minimum salary greater than that of 'Sales' department.

```
select * from employee where salary>(select min(salary) from employee where dept not in('Sales')) and salary>(select max(salary) from
```

12. Find the employees who earn the same salary for each department

```
select e1.emp_id from employee e1, employee e2 where e1.salary=e2.salary and e1.emp_id=e2.emp_id and e1.dept<>e2.dept;
```

13. Display the employees who are not engineers and whose salary is less than that of any engineer.

```
select * from employee where job_type not in('Engineer') and salary<(select min(salary) from employee where job_type='Engineer');
```

14. Display the employees whose salary is less than the salary of all employees with a job_type 'Clerk' and whose job_type is not 'Clerk'.

```
select * from employee where job_type not in('Clerk') and salary<(select min(salary) from employee where job_type='Clerk');
```

15.

Assignment - 9

1. Write a PL/SQL program to find the largest of three numbers

```
Declare
  a number;
  b number;
  c number;
Begin
  dbms_output.put_line('Enter a:');
  a:=&a;
  dbms_output.put_line('Enter b:');
  b:=&b;
  dbms_output.put_line('Enter c:');
  c:=&c;
  if (a>b) and (a>c)
  then
    dbms_output.put_line('A is GREATEST'||A);
  elsif (b>a) and (b>c)
  then
    dbms_output.put_line('B is GREATEST'||B);
  else
    dbms_output.put_line('C is GREATEST'||C);
  end if;
End;
```

2. Write a PL/SQL program to generate reverse for given number

```
declare
n number(4) := &n;
s number(4) := 0;
r number(4);
begin
while n > 0
loop
r:= mod(n,10);
s:=(s*10)+r;
n:=trunc(n/10);
end loop;
dbms_output.put_line('the reverse number is');
dbms_output.put_line(s);
end;
```

3. Write a PL/SQL program to find the factorial of a given number

```
declare
i number(4) :=1;
n number(4) := &n;
f number(4) :=1;
begin
for i in 1..n
loop
f:=f*i;
end loop;
dbms_output.put_line('factorial of a number is '|| f);
end;
```

4. Write a PL/SQL program to check whether given number is prime or not

```
declare
n number;
i number;
flag number;

begin
i:=2;
flag:=1;
n:=&n;

for i in 2..n/2
loop
if mod(n,i)=0
then
flag:=0;
exit;
end if;
end loop;

if flag=1
then
dbms_output.put_line('prime');
else
dbms_output.put_line('not prime');
end if;
end;
/
```

5. Write a PL/SQL program to generate Fibonacci series upto N

```
declare
a number := 0;
b number := 1;
temp number;
n number := 10;
i number;
begin
dbms_output.put_line('fibonacci series is ');
dbms_output.put_line(a);
dbms_output.put_line(b);
for i in 2..n
loop
temp:= a + b;
a := b;
b := temp;
dbms_output.put_line(temp);
end loop;
end;
```

6. Write a PL/SQL program for calculating sum of two numbers.

```
Declare
a number(5);
b number(5);
c number(5);
Begin
a:=100;
b:=110;
c:=a+b;
dbms_output.put_line(c);
End;
/
```

7. Write a PL/SQL program to check the given year is leap year or not

```
DECLARE
year NUMBER := 2012;
BEGIN
IF MOD(year, 4)=0
```

```

        AND
        MOD(year, 100) != 0
        OR
        MOD(year, 400) = 0 THEN
            dbms_output.Put_line(year || ' is leap year ');
    ELSE
        dbms_output.Put_line(year || ' is not leap year. ');
    END IF;
END;

```

8. Find the sum of the digits of a given number

```

declare
    num int := 0;
    i int;
    s int := 0;
    r int;

begin
    num := &num;
    while num > 0 loop
        r := MOD(num, 10);
        s := s + r;
        num := floor(num/10);
    end loop;

    dbms_output.put_line(' the sum of digits is ' || s );

end;

```

9. Check the number of vowels and consonants in a given string

```

DECLARE
    v          VARCHAR2(400) := 'Ramesh is a Geek';
    noofvowels NUMBER := 0;
    noofconsonants NUMBER := 0;
    c          CHAR;
BEGIN
    FOR i IN 1..Length(v) LOOP
        c := Substr(v, i, 1);

        -- Check if the current character is vowel
        IF c IN ( 'A', 'E', 'I', 'O', 'U' )
           OR c IN ( 'a', 'e', 'i', 'o', 'u' ) THEN
            noofvowels := noofvowels + 1;

        -- Else current character is a consonant except space
        ELSE
            IF c NOT IN ( ' ' ) THEN
                noofconsonants := noofconsonants + 1;
            END IF;
        END IF;
    END LOOP;

    dbms_output.Put_line('No. of Vowels: '
                        || noofvowels);

    dbms_output.Put_line('No. of Consonants: '
                        || noofconsonants);

END;

```

10. Count odd and even digits in a number

```

DECLARE
    digits NUMBER := 23146579;
    length2 VARCHAR2(50);
    count_odd NUMBER(10) := 0;
    count_even NUMBER(10) := 0;
BEGIN
    FOR i IN 1..Length(digits)
    LOOP
        length2 := Substr(digits, i, 1);
    END LOOP;

```

```

IF mod(length2, 2) != 0 THEN
    count_odd := count_odd + 1;
ELSE
    count_even := count_even + 1;
END IF;
END LOOP;
dbms_output.Put_line('count of odd digits in a number are : ' || count_odd);
dbms_output.Put_line('count of even digits in a number are : ' || count_even);
END;

```

11. Explain the concepts of stored procedure and triggers in a database management system.

Sr. No.	Key	Triggers	Stored procedures
1	Basic	trigger is a stored procedure that runs automatically when various events happen (eg update, insert, delete)	Stored procedures are a pieces of the code in written in PL/SQL to do some specific task
2	Running Methodology	It can execute automatically based on the events	It can be invoked explicitly by the user
3	Parameter	It can not take input as parameter	It can take input as a parameter
4	Transaction statements	we can't use transaction statements inside a trigger	We can use transaction statements like begin transaction, commit transaction, and rollback inside a stored procedure
5	Return	Triggers can not return values	Stored procedures can return values