Assignment 1

1) Select the employee in department 30.

```
SELECT * FROM employees WHERE department_id=30;
```

2) List the names, numbers and department of all clerks.

```
SELECT first_name,last_name,phone_number,job_id,department_id
FROM employees
WHERE job_id='PU_CLERK';
```

3) Find the depart numbers and the name of employee of all dept with Deptno greater or equal to 20.

```
SELECT department_id,first_name,last_name
FROM employees
WHERE department_id >=20
```

4) Find the employees whose commission is greater than their salary.

```
SELECT first_name,last_name
FROM employees
where commission_pct > salary;
```

5) Find the employees whose commission is greater than 60 percent of their salary.

```
SELECT first_name,last_name
FROM employees
where (commission_pct*salary)>(salary*0.6)
```

6) Find the employee whose commission is greater than 50 percent of their salary.

```
SELECT first_name,last_name
FROM employees
where (commission_pct*salary)>(salary*0.5)
```

7) List the name, job and salary of all employees in dept 20 who earn more than 2000.

```
SELECT first_name,last_name,job_id,salary
FROM employees
where department_id=20 and salary >2000
```

8) Find all salesmen in dept 30 whose salary is greater than or equal to Rs. 1500.

```
--NOTE; IF SALESMAN = PU_CLERK

SELECT first_name,last_name,job_id
FROM employees
WHERE job_id = 'SA_REP' and department_id = 30 and salary >= 1500
```

9) Find all the employees whose job is either a president or manager.

```
SELECT * FROM employees
WHERE job_id like '%PRES' or job_id like '%MGR' or job_id like '%MAN'
```

10) Find all managers who are not in dept 30.

```
SELECT * From Employees
WHERE job_id like '%MGR' and department_id=!30
```

11) Find the details of all managers and clerks in dept 10.

```
SELECT * FROM employees
where job_id LIKE '%MAN' or job_id like '%MGR' or job_id like '%CLERK' and
department_id=10
```

12) Find the details of all manager (in any dept) and all clerks in dept 10

```
SELECT * FROM employees
where job_id LIKE '%MAN' or job_id like '%MGR' or job_id like '%CLERK' and
department_id=10
```

13) Find the details of all managers in dept 10 and all clerks in dept 20.

```
SELECT * FROM employees
WHERE (job_id LIKE '%MGR' or job_id LIKE '%MAN' and department_id = 10) or (job_id
LIKE '%CLERK' and department_id=20)
```

14) Find the details of all the manager in dept 10, all clerk in dept 20

```
SELECT * FROM employees
WHERE (job_id LIKE '%MGR' or job_id LIKE '%MAN' and department_id = '10') or (job_id
LIKE '%CLERK' AND department_id='20');
```

15) And all employees who are neither clerks nor manager but whose salary is greater than or equal to Rs. 2000.

```
SELECT * FROM employees
WHERE NOT (job_id like '%MGR' or job_id like '%MAN' or job_id != '%CLERK') AND salary
>= 2000
```

16) Find the names of everyone in deptno 20 who is neither a clerk nor a Manager.

```
SELECT * FROM employees
WHERE NOT (job_id like '%MGR' or job_id like '%MAN' or job_id like '%CLERK') AND
department_id = 20
```

17) Find the employees who earns between Rs. 1200 and Rs.1400.

```
SELECT * FROM employees
WHERE salary between 1200 and 11400
```

18) Find the employees who are clerks, analysts or salesman.

```
SELECT * FROM
employees
```

WHERE (job_id like '%CLERK' or job_id like '%ANALYST' or job_id like 'SA%')

19) Find the employees who are not clerks, analyst or salesman.SELECT *
FROM employees
WHERE NOT (job_id like '%CLERK' or job_id like '%ANALYST' or job_id like 'SA%')

20) Find the employees who do not receive a commission.
SELECT * FROM employees
WHERE commission_pct is null

21) Find the employee whose commission is Rs. 0.
SELECT * FROM employees
WHERE commission_pct is 0

22) Find the different jobs of the employees receiving commission.SELECT *
FROM employees
WHERE commission_pct is not null

23) Find all employees who do not receive a commission or whose Commission is less than 0.1 .
If all employees not receiving commission are entailed to Rs. 250, Show the net earnings of all employees.
SELECT first_name,salary + (nvl2(commission_pct,(commission_pct*salary),(+250))) Net_Earning FROM employees

24) Find all employees whose total earnings are greater than Rs. 2000.
SELECT * FROM employees
WHERE (nvl(commission_pct,0)*salary)+salary >2000

25) Find all employees whose names begin with m.
SELECT * FROM employees
where first_name like 'M%'

26) Find all employees whose names end with m.
SELECT * FROM employees
where last_name like '%m'

27) Find all employees whose names contain the letter m in any case.
SELECT * FROM employees
where Lower(first_name) like '%m%'

28) Find the employees whose names are 5 characters long and end with n.
SELECT * FROM employees
where first_name like '____%n;

29) Find the employees who have the letter r as the third letter in their name.

```
SELECT * FROM employees
where first_name like'__r%';
```

30) Find all employees hired in month of February (of any year).

```
SELECT * FROM employees
WHERE EXTRACT(Month from hire_date)=2;
```

31) Find all employees who were hired on the last day of the month.

```
SELECT * FROM employees
WHERE hire_date=last_day(hire_date);
```

32) Find the employees who were hired more than 12 years ago.

```
SELECT * FROM employees
WHERE EXTRACT(YEAR FROM hire_date)< EXTRACT(YEAR FROM
add_months(SYSDATE,144))
```

33) Find the managers hired in the year 1981.

```
SELECT * FROM employees
WHERE employee_id in ( select unique manager_id from employees) and
to_char(hire_date,'YYYY')=1981;
```

34) Display the names and the jobs of all employees, separated by a',

```
SELECT first_name||','||job_id from employees;
```

35) Display the names of all employees with the initial letter only in capitals.

```
SELECT initcap(first_name) from employees;
```

36) Display the length of the name of all employees.

```
SELECT  first_name, last_name, length(first_name)+length(last_name) from employees;
```

37) Show the first three characters of the names of all employees.

```
SELECT substr(first_name,1,3) from employees;
```

38) Show the last three characters of the names of all employees.

```
SELECT reverse(substr(reverse(first_name),1,3)) from employees;
```

39) Display the names of all employees with any 'a'.

```
SELECT first_name
FROM employees
WHERE first_name like '%a%;
```

40) Display the names of all employees and the position at which the string 'ar' occurs in the name.

```
SELECT first_name,instr(first_name,'ar',1) from employees;
```

41) Show the salary of all employees rounding it to the nearest Rs. 1000.

SELECT salary,ceil(salary/1000)*1000 from employees;

42) Show the salary of all employees ignoring fractions ,less than Rs.1000.

```
SELECT TRUNC(salary)
FROM employees
WHERE salary < 1000;
```

43) Display the details of all employees, sorted on the names.

```
SELECT * FROM employees order by first_name;
```

44) Display the name of all employees, based on their tenure, with the oldest employee coming first.

```
SELECT first_name, hire_date
FROM employees order by hire_date;
```

45) Display the names, job and salary of all employees sorted on jobs and Salary.

```
SELECT first_name,job_id,salary FROM employees order by salary,job_id;
```

46) Display the names, job and salary of all employees, sorted on jobs and within job, sorted on the descending order of salary.

```
SELECT first_name,job_id,salary
FROM employees order by salary desc;
```

Assignment 2

1. write a SQL query to find those employees who get higher salary than the employee whose ID is 163. Return first name, last name.

```
SELECT first_name, last_name,employee_id
FROM employees
WHERE salary >(SELECT salary FROM employees WHERE employee_id=163);
```

2. Display the name, salary, department id, job id for those employees who works in the same designation as the employee works whose id is 169

```
SELECT first_name,salary,department_id,job_id
FROM employees
WHERE job_id = (SELECT job_id FROM employees WHERE employee_id=169);
```

3.Display the name, salary, department id for those employees who earn such amount of salary which is the smallest salary of any of the departments

```
SELECT first_name,last_name,salary,department_id
FROM employees
WHERE salary
IN (SEIECT MIN(salary) FROM employees GROUP BY department_id);
```

4. Display the employee id, employee name for all employees who earn more than the average salary

```
SELECT employee_id,first_name,last_name
FROM employees
WHERE salary >
(SELECT AVG(salary) FROM employees);
```

5. Display the employee name, employee id and salary of all employees who report to John

```
SELECT first_name,last_name,employee_id,salary
FROM employees
WHERE manager_id = ANY (SELECT employee_id
FROM employees
WHERE first_name ='John');
```

6. SQL query to find all those employees who work in the HR department. Return department ID, name (first name), job ID and department name.

```
SELECT e.department_id,e.first_name,e.last_name,e.job_id,d.department_name
FROM employees e, departments d
WHERE e.department_id=d.department_id AND d.department_name ='Human Resources';
```

7. write a SQL query to find those employees whose ID matches any of the number 134, 159 and 183. Return all the fields.

```
SELECT *  FROM employees
WHERE employee_id IN (134,159,183);
```

Assignment 3

a. **Table Name :**salesman_master

Description: Use to store information about salesman working in the company

Column Name	Data Type	Size	Attributes
salesman_no	Varchar	6	Primary key/first letter must start with 'S'
salesman_name	Varchar	20	not null
Address1	Varchar	30	not null
Address2	Varchar	30	
city	Varchar	20	
pincode	Varchar	6	
state	Varchar	20	
sal_amt	number	8,2	not null, cannot be 0
tgt_to_get	number	6,2	not null, cannot be 0
ytd_sales	number	6,2	not null
remarks	Varchar	60	

```
CREATE TABLE salesman_master (
salesman_no VARCHAR(6) PRIMARY KEY CONSTRAINT START_S
            CHECK  (salesman_no LIKE 'S%'),
salesman_name VARCHAR(20) NOT NULL,
address1 VARCHAR(30) NOT NULL,
address2 VARCHAR(30),
city VARCHAR(20),
pincode VARCHAR(6),
```

```
state VARCHAR(20),
sal_amt NUMBER(8,2)NOT NULL CONSTRAINT
        SALCHECKNOTZERO  CHECK(SAL_AMT != 0),
tgt_to_get NUMBER(6,2)NOT NULL CONSTRAINT
        TGTCHECKNOTZERO  CHECK(TGT_TO_GET != 0),
ytd_sales NUMBER(6,2)NOT NULL,
remarks VARCHAR(60));
```

b. Table Name :sales_order

Description: Use to store information about order

Column Name	Data Type	Size	Attributes
s_order_no	Varchar	6	Primary key/first letter must start with 'O'
s_order_date	Datetime		
client_no	Varchar	6	Foreign key references client_no of client_master table
dely_addr	Varchar	25	
salesman_no	Varchar	6	Foreign key references salesman_no of salesman_master table
dely_type	Char	1	Default 'F',delivery :part (P) / full (F)
billed_yn	Char	1	delivery :part (Y) / full (N), Default 'N'
dely_date	Date		cannot be less than s_order_date
order_status	Varchar	10	values ('in process', 'Fulfilled', 'BackOrder', 'Canceled')

```
Create table SALES_ORDER (
S_ORDER_NO VARCHAR(6) PRIMARY KEY CONSTRAINT
START_IN_O  CHECK(S_ORDER_NO LIKE 'O%'),
S_ORDER_DATE DATE DEFAULT '10-MAY-96',
CLIENT_NO VARCHAR(6) REFERENCES
CLIENT_MASTER(CLIENT_NO), DELY_ADDR VARCHAR(25),
BILLED_YN CHAR(1) DEFAULT 'N',
SALESMAN_NO VARCHAR(6) REFERENCES
SALESMAN_MASTER(SALESMAN_NO), DELY_DATE DATE ,
ORDER_STATUS VARCHAR(10),
CONSTRAINT CHECK_DATE
CHECK(DELY_DATE>S_ORDER_DATE), CONSTRAINT
CHECK_Y_N CHECK (BILLED_YN = ANY('Y','N')),
CONSTRAINT ORDER_STAT_CHECK
CHECK(ORDER_STATUS = ANY('in  process','FullFilled',
'BackOrder', 'Canceled')));
```

c.**Table Name:**sales_order_details

Description: Use to store information about products ordered.

Column Name	Data Type	Size	Attributes
s_order_no	Varchar	6	Foreign key references s_order_no of sales_order table
product_no	Varchar	6	Foreign key references product_no of product_master table
qty_ordered	Numeric	8	
qty_disp	Numeric	8	
product_rate	Numeric	10,2	

```
create table sales_order_details(
s_order_no varchar(6) references sales_order,
product_no varchar(6) references product_master,
qty_ordered number(8),
```

qty_disp number(8),
product_rate number(10,2));

3. Data for salesman_ master table:

Salesman_ no	Salesman_ name	Address1	Address2	City	Pin code	State	sal_amt	Tgt_to Get	Ytd sales	Remarks
S00001	Kiran	A/14	Worli	Bombay	400002	MAH	3000	100	50	Good
S00002	Manish	65	Nariman	Bombay	400001	MAH	3000	200	100	Good
S00003	Ravi	P-7	Bandra	Bombay	400032	MAH	3000	200	100	Good
S00004	Ashish	A/5	Juhu	Bombay	400044	MAH	3500	200	150	Good

```
INSERT ALL
INTO salesman_master
(salesman_no, salesman_name, address1, address2,
city,pincode,state,sal_amt,tgt_to_get,ytd_sales, remarks)
VALUES('S00001', 'Kiran', 'A/14', 'Worli', 'Bombay', '400002', 'MAH', '3000',
      '100', '50', 'Good') INTO salesman_master
(salesman_no, salesman_name, address1, address2,
city,pincode,state,sal_amt,tgt_to_get,ytd_sales, remarks)
VALUES('S00002', 'Manish', '65', 'Nariman', 'Bombay', '400001', 'MAH',
      '3000', '200', '50',  'Good')
INTO salesman_master
(salesman_no, salesman_name, address1, address2,
city,pincode,state,sal_amt,tgt_to_get,ytd_sales, remarks)
VALUES('S00003', 'Ravi', 'P-7', 'Bandra', 'Bombay', '400032', 'MAH', '3000',
      '200', '50', 'Good') INTO salesman_master
(salesman_no, salesman_name, address1, address2,
city,pincode,state,sal_amt,tgt_to_get,ytd_sales, remarks)
VALUES('S00004', 'Ashish', 'A/5', 'Juhu', 'Bombay', '400004', 'MAH', '3500',
      '200', '50', 'Good') SELECT * FROM dual;
```

4. Data for sales_ordertable :

S_order_no	S_order_date	Client No	Dely Type	Bill Yn	salesman no	Dely Date	Order Status
O19001	12-Jan-2021	C00001	F	N	S00001	20-Jan-2021	IP
O19002	25-Jan-2021	C00002	P	N	S00002	27-Jan-2021	C
O46865	18-Feb-2021	C00003	F	Y	S00003	20-Feb-2021	F
O19003	03-Apr-2021	C00001	F	Y	S00001	07-Apr-2021	F
O46866	20-May-2021	C00004	P	N	S00002	22-May-2021	C
O10008	24-May-2021	C00005	F	N	S00004	26-May-2021	IP

```
INSERT INTO SALES_ORDER
VALUES
('O19001','12-JAN-2021','C00001','F','N','S00001','20-JAN-2021', 'in
process'); INSERT INTO SALES_ORDER
VALUES
('O19002','25-JAN-2021','C00002','P','N','S00002','27-JAN-2021',
'Canceled'); INSERT INTO SALES_ORDER
VALUES
('O46865','18-FEB-2021','C00003','F','Y','S00003','20-FEB-2021',
'FullFilled'); INSERT INTO SALES_ORDER
```

VALUES
('O19003','03-APR-2021','C00001','F','Y','S00001','07-APR-2021',
'FullFilled'); INSERT INTO SALES_ORDER
VALUES
('O46866','20-MAY-2021','C00004','P','N','S00002','22-MAY-2021',
'Canceled'); INSERT INTO SALES_ORDER
VALUES ('O10008','24-MAY-2021','C00005','F','N','S00004','26-MAY-2021', 'in process');

5. Data for sales_order_details table:

s_order_no	product_no	Qty ordered	qty_Disp	Product rate
O19001	P00001	4	4	525
O19001	P07965	2	1	8400
O19001	P07885	2	1	5250
O19002	P00001	10	0	525
O46865	P07868	3	3	3150
O46865	P07885	3	1	5250
O46865	P00001	10	10	525
O46865	P03453	4	4	1050
O19003	P03453	2	2	1050
O19003	P06734	1	1	12000
O46866	P07965	1	0	8400
O46866	P07975	1	0	1050
O10008	P00001	10	5	525
O10008	P07975	5	3	1050

INSERT INTO
SALES_ORDER_DETAILS VALUES
('O19001','P00001',4,4,525); INSERT
INTO SALES_ORDER_DETAILS
VALUES ('O19001','P07965',2,1,8400);
INSERT INTO
SALES_ORDER_DETAILS VALUES
('O19001','P07885',2,1,5250); INSERT
INTO SALES_ORDER_DETAILS
VALUES ('O19002','P00001',10,0,525);
INSERT INTO
SALES_ORDER_DETAILS VALUES
('O46865','P07868',3,3,3150); INSERT
INTO SALES_ORDER_DETAILS
VALUES ('O46865','P07885',3,1,5250);
INSERT INTO
SALES_ORDER_DETAILS VALUES
('O46865','P00001',10,10,525); INSERT
INTO SALES_ORDER_DETAILS
VALUES ('O46865','P03453',4,4,1050);
INSERT INTO
SALES_ORDER_DETAILS VALUES
('O19003','P03453',2,2,1050); INSERT
INTO SALES_ORDER_DETAILS
VALUES ('O19003','P06734',1,1,12000);
INSERT INTO
SALES_ORDER_DETAILS VALUES
('O46866','P07965',1,0,8400); INSERT

```
INTO SALES_ORDER_DETAILS
VALUES ('O46866','P07975',1,0,1050);
INSERT INTO
SALES_ORDER_DETAILS VALUES
('O10008','P00001',10,5,525); INSERT
INTO SALES_ORDER_DETAILS
VALUES ('O10008','P07975',5,3,1050);
```