ASSIGNMENT 1

Rule- connect sys/oracle as sysdba

1. Write a query to create a table employee with empno, ename, designation and salary.
2. Write a query to display the column name and data type of the table employee
3. Write a query to create a table from an existing table with all the fields.
4. Write a query to create table from an existing table with selected fields.
5. Write a query to create a new table from an existing table without any record.
6. Write a query to Alter the column empno number(4) to empno number(6).
7. Write a query to Alter the table employee with multiple columns (empno, ename).
8. Write a query to add a new column in employee table.
9. Write a query to add multiple columns in employee table.
10. Write a query to drop a column from an existing table employee.
11. Write a query to drop multiple columns from the employee table.
12. Write a query to rename table employee to emp

```
use assignment_1;
create table employee(empno binary(10),ename varchar(10),designation varchar(10),salary
binary(20));
desc employee;

select * from employee;
create table emp as select * from employee where 1>1;

select * from emp;
create table emp1 as select * from emp where 1>1;

create table emp2 as select * from employee where 1=0;

alter table employee modify empno binary(6);

alter table employee modify ename varchar(20);
alter table employee modify empno binary(12);

alter table employee add (doj date);

alter table employee add (dept varchar(10),age binary(10));

alter table employee drop column age;

alter table employee drop column doj;
alter table employee drop column dept;

alter table emp1 rename to employee1;
```

ASSIGNMENT 2

1. Create a table employee with attributes emp_id, f_name, l_name, job_type, salary, dept, commission, manager_id.
2. Make emp_id as the primary key of employee table.
3. Make f_name and salary NOT NULL type.
4. Add a column date_of_joining in the employee table.
5. Create a table department with attribute d_name, d_loc and HOD_id where d_name is primary key.
6. Create a table location with attributes loc_id, city and contact_no.
7. Enhance the size of the ‘city’ attribute by 5, in the location table.
8. Delete the contact_no attribute from the location table.
9. Make the department attribute of the employee table its foreign key referencing the department table.
10. Rename the city attribute to ‘address’ in the location table.
11. Rename the location table name to ‘loc’.
12. Insert the following rows in ‘loc’ table

loc_id	address
1	Kolkata
2	Mumbai

13. Truncate the table ‘loc’.
14. Drop the 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

16. Insert the following rows in the employee table:

Emp id	E_Name	L_name	Job_type	Salary	Comission	Name_id	DOJ
1	Arun	Khan	Manager	90000	500	5	04-Jan-1999

.

.

.

.

.

17. Show the values of departmental table.
18. Select the department names and their locations.
19. Show the employees f_name , l_name , salary and the salary after 1000rs. Bonus.
20. Show the employees annual salary with a 1000rs. Yearly bonus and the annual salary with a 100rs. Monthly bonus.
21. Show f_name as NAME and annual salary as ANNSAL from the employee table.
22. Show the l_name as LasT AND 100rs. Incremented salary as NewSal.
23. Show the emp_id, f_name, l_name, job_type of the employee getting highest salary.
24. Show the emp_id, f_name, l_name, job_type of the employee getting minimum salary.
25. Show the average salary of employees in the employee table.
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.

```
use assignment_2;
create table employee(emp_id binary(10),f_name varchar(10),l_name varchar(10),job_type
varchar(10),salary binary(20),dept varchar(10),commision binary(10),manager_id binary(10));
alter table employee add primary key(emp_id);
alter table employee modify f_name varchar(10) NOT NULL;
alter table employee modify salary binary(20) NOT NULL;
alter table employee add (doj date);
create table department (d_name varchar(10) primary key,d_loc varchar(10),HOD_id binary(10));
create table location (loc_id binary(10),city varchar(10),contact_no binary(20));
alter table location modify city varchar(5);
alter table location drop contact_no;
alter table employee add foreign key(dept) references department(d_name);
alter table location rename column city to address;
desc location;
alter table location rename to loc;
alter table loc modify address varchar(20);
insert into loc (loc_id,address) values(1,'Kolkata');
insert into loc (loc_id,address) values(2,'Mumbai');
select * from loc;
truncate loc;
desc loc;
```

```
drop table loc;
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);
select * from department;
alter table employee modify emp_id int(10);
alter table employee modify salary int(20);
alter table employee modify commision int(10);
alter table employee modify manager_id int(10);
desc employee;
alter table employee drop primary key;
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(1,'Arun','Khan','Manager',90000,'Production','1998-01-04');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(2,'Barun','Kumar','Manager',80000,'Marketing','1998-02-09');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(3,'Chitra','Kapoor','Engineer',60000,'Production',1,'1998-01-08');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(4,'Deeraj','Mishra','Manager',75000,'Sales',4,'2001-12-27');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(5,'Emma','Dutta','Engineer',55000,'Production',1,'2002-03-20');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(6,'Chinaswami','Iyer','Accounts',70000,'Accounts','2001-07-16');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(7,'Deeraj','Kumar','Clerk',40000,'Accounts',6,'2016-12-16');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(8,'Mr.','Paul','Engineer',60000,'R&D','2014-09-06');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(9,'Raj','Mishra','Clerk',30000,'Sales',4,'2018-03-27');
insert into employee(emp_id,f_name,l_name,job_type,salary,commision,dept,manager_id,doj)
values(10,'Sunny','Deol','Salesman',20000,10000,'Marketing',2,'2003-03-31');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(11,'Bobby','Deol','Engineer',35000,'R&D',8,'2017-10-07');
insert into employee(emp_id,f_name,l_name,job_type,salary,commision,dept,manager_id,doj)
values(10,'Choton','Khan','Salesman',15000,5000,'Marketing',2,'2013-05-11');
select d_name,d_loc from department;
select f_name,l_name,salary from employee where commision=1000;
select salary+1000,salary+12*100 from employee;
select f_name as name,salary as annsal from employee;
select l_name as "LAST", salary+100 as"Newsal" from employee;
select emp_id,f_name,l_name,job_type from employee where salary=(select max(salary) from
employee);
select emp_id,f_name,l_name,job_type from employee where salary=(select min(salary) from
employee);
select avg(salary) from employee;
```

ASSIGNMENT 3

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

Emp id	E_Name	L_name	Job_type	Salary	Comission	Name_id	DOJ
1	Arun	Khan	Manager	90000	500	5	04-Jan-1999
.							
.							
.							
.							
.							

1. Show f_name, l_name and job_type from employees.
2. Show employee details in the following fashion:

Employee details
Arun is a manager
3. Show the monthly salary details in the following fashion
Monthly Salary Details
Arun's monthly salary is Rs. 90000
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
5. Show the employee names who works in 'Sales'
6. Show the employee names who gets salary of more than 50000 per month
7. Show the details of the employee whose manager id is not 1
8. Show the employee details whose salary ranges between 40000 and 70000
9. Show the details of the employees who works under the manager having id 1, 6 and 8
10. Select the f_name and salary of those employees whose last name starts with 'K'
11. Select the f_name and salary of those employees whose last name starts with 'K' and ends with 'R'
12. Show the details of those employees where 3rd letter of l_name is 'o'
13. Select the details of those employees who works as an engineer with monthly salary more than 50000
14. Select the employees whose department is 'Production' or monthly salary is more than 60000 per month.
15. Find the minimum salary, maximum salary, total salary, average salary of the

employees who work in ‘Sales’ department
16. Find the employee l_name that is first and f_name that is last if they are arranged in an order
17. Find the number of employees working in each department
18. Find the number of departments from employee table
19. Find the average commission of the employees.
20. Find the average salaries of the employees department wise
21. Find the sum of salary of different job_type according to different departments
22. Find the department name and average salaries of those departments whose average salary is greater than 40000
23. Find the department name and maximum salaries of those departments whose maximum salary is greater than 55000
24. Display the job_type and total monthly salary for each job_type where total payroll is exceeding 100000
25. Display the name of the department having maximum average salary

```
use assignment_3;
create table employee(emp_id int(10),f_name varchar(20),l_name varchar(10),job_type
varchar(10),salary int(20),dept varchar(10),commision int(10),manager_id int(10));
alter table employee add (doj date);
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(1,'Arun','Khan','Manager',90000,'Production','1998-01-04');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(2,'Barun','Kumar','Manager',80000,'Marketing','1998-02-09');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(3,'Chitra','Kapoor','Engineer',60000,'Production',1,'1998-01-08');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(4,'Deeraj','Mishra','Manager',75000,'Sales',4,'2001-12-27');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(5,'Emma','Dutta','Engineer',55000,'Production',1,'2002-03-20');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(6,'Chinaswami','Iyer','Accounts',70000,'Accounts','2001-07-16');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(7,'Deeraj','Kumar','Clerk',40000,'Accounts',6,'2016-12-16');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,doj)
values(8,'Mr.','Paul','Engineer',60000,'R&D','2014-09-06');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(9,'Raj','Mishra','Clerk',30000,'Sales',4,'2018-03-27');
insert into employee(emp_id,f_name,l_name,job_type,salary,commision,dept,manager_id,doj)
values(10,'Sunny','Deol','Salesman',20000,10000,'Marketing',2,'2003-03-31');
insert into employee(emp_id,f_name,l_name,job_type,salary,dept,manager_id,doj)
values(11,'Bobby','Deol','Engineer',35000,'R&D',8,'2017-10-07');
insert into employee(emp_id,f_name,l_name,job_type,salary,commision,dept,manager_id,doj)
values(10,'Choton','Khan','Salesman',15000,5000,'Marketing',2,'2013-05-11');
select * from employee;
select f_name,l_name,job_type from employee;
create table department (d_name varchar(10) primary key,d_loc varchar(10),HOD_id int(10));
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);
select * from department;
select d_name from department;
select f_name,l_name from employee where dept='sales';
select f_name,l_name from employee where salary>50000;
select emp_id,f_name,l_name,salary,dept,job_type from employee where manager_id!=1;
select emp_id,f_name,l_name,dept,job_type from employee where salary>40000 and
salary<70000;
select emp_id,f_name,l_name,salary,dept,job_type from employee where manager_id>0 and
manager_id<9;
select f_name,salary from employee where l_name like 'k%';
select f_name,salary from employee where l_name like 'k%' and l_name like '%r';
select emp_id,f_name,l_name,dept,job_type from employee where l_name like '__o%';
select emp_id,f_name,l_name,dept,job_type from employee where job_type='engineer' and
salary>50000;
select emp_id,f_name,l_name,dept,job_type from employee where dept='production' and
salary>60000;
select min(salary),max(salary),avg(salary),sum(salary) from employee where dept='sales';
select f_name,l_name from employee order by f_name,l_name ASC;
select dept,count(dept) from employee group by dept;
select count(dept) from employee;
select avg(commision) from employee;
select avg(salary),dept from employee group by dept;
select job_type,sum(salary),dept from employee group by dept;
select dept,avg(salary) from employee where salary>40000 group by dept;
select dept,max(salary) from employee where salary>50000 group by dept;
select job_type,sum(salary) from employee where salary<100000 group by job_type;
select dept,avg(salary) from employee group by dept having avg(salary)=(select max(avg(salary))
from employee group by dept);
```

ASSIGNMENT 4

1. Show the use of upper and lower function.
2. Show the use of concat, instr and length function
3. Show the use of the following functions on numeric values:

a. Sqrt()
b. Power()
c. Ceil()
d. Substr()
e. Max()
f. min()

g. Round()
h. avg()
i. count()
j. Exp()
k. mod()
4. Solve the following queries
a. Find the ceiling and floor value of 14.887.
b. Find out the round-off 17.49989.
c. Calculate 87
.
5. Show the current date
6. Find the total experience of the employees in weeks who works in Sales department
7. Display the use of the following functions on date
a. Months_between
b. Add_months
c. Next_day
d. Last_day
e. Round
f. Trunc
g. To_char
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
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.
11. Show the employee names and the respective department location.
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

13.Write a query to find the addresses(location_id,street_address,city,state_province,country_name) of all departments
HINT : Use NATURAL JOIN.
Sample Table : locations

location_id	street_address	postal_code	city	state_province	country_id

1000 1297 via... 890 Rome Italy
IT
.
.
.
.
.
.

14.Wrte a query to find the name(firsr_name,last name),department ID and name of all the employees

employee_id firs_name last_name email phone_no hire_date
job_id
.
.
.
.
.
.

```
use assignment_4;
select upper('joy mohunbagan'), lower('JOY MOHUNBAGAN');
create table student(f_name varchar(20),l_name varchar(10));
insert into student(f_name,l_name) values('Chinnaswami','Iyer');
insert into student(f_name,l_name) values('Sribasana','Iyer');
select concat(f_name,l_name) from student;
select instr("sch","3") as matchposition;
select length('joymohunbagan');
select sqrt(amount) from table_name;
select power(4,2);
select ceil(25.75);
select substr("Joy Mohunbagan",5,3) as extractstring;
create table price(id int(10),price int(10));
insert into price(id,price) values(1,50);
insert into price(id,price) values(2,100);
insert into price(id,price) values(3,150);
select * from price;
select max(price) as largestprice from price;
select min(price) as smallestprice from price;
select round(25.51) as roundvalue;
select avg(price) from price;
select count(id) from price;
select exp(1);
select mod(18,4);
select ceiling(14.887) as ceilvalue, floor(14.887) as floorvalue;
select round(11.11111111) as roundvalue;
select power(8,7);
select getdate();
select * from employee where extract (year from age(current_date,hire-date)>1);
select months_between (to_date('02-02-2015','MM-DD-YYYY'), to_date('12-01-2014','MM-DD-YYYY'))"Months" from dual;
select to_char(add_months(hire_date,1),'DD-MM-YYYY')"next months" from dual;
select next_day('12-MAR-2015','MONDAY')"Next Day" from dual;
select last_day("2022-09-09");
select round(1.1,2);
select turncate(135.375,2);
select to_char(add_months(hire_date,1),'DD-MM-YYYY')"Next Month" from dual;
update employee set salary=salary+(salary*10/100);
update employee set marketing=marketing+(marketing*20/100);
```

```
use assignment_4;
create table employee (e_id varchar(10),e_name varchar(20), doj date, salary int(20), d_id varchar(10),d_name varchar(20));
create table department(d_id varchar(10),d_name varchar(20),name varchar(20));
insert into employee values('E203','Choton','2019-10-19',56000,'D420','Bekar');
insert into employee values('E049','Sumit','2019-01-08',65000,'D120','Marketing');
insert into employee values('E713','Paul','2016-10-06',86000,'D120','Marketing');
insert into employee values('E629','Prem','2017-12-19',49000,'D320','Sales');
insert into employee values('E191','Kabla','2011-03-10',66000,'D520','IT');
insert into employee values('E432','Vavla','2012-03-09',65000,'D425','Entry');
insert into employee values('E247','Kakarjibonfaka','2009-11-11',76000,'D247','Backofc');
insert into department values('D120','Marketing','Paglada');
insert into department values('D420','Bekar','Chaglada');
insert into department values('D320','Sales','Bokada');
insert into department values('D520','IT','Mondnada');
insert into department values('D425','Entry','Kestoda');
insert into department values('D247','Backofc','Mamatadidi');
select * from employee cross join department;
select * from employee E cross join department D where D.d_id=E.e_id;
select * from employee natural join department;
select * from employee E inner join department D on D.d_id=E.e_id;
select * from employee E left join department D on D.d_id=E.e_id;
select * from employee E right join department D on D.d_id=E.e_id;
select * from employee E full outer join department D on D.d_id=E.e_id;
select * from locations natural join countries;
select f_name,l_name,department_id from employee;
```

ASSIGNMENT 6

worker_id	first_name	last_name	salary	joining_date	department
.					
.					
.					
.					
.					
.					
.					

worker_ref_id	bonus_date	bonus_amount
.		
.		
.		
.		
.		

worker_ref_id	worker_title	affected_form
.		
.		
.		
.		
.		

1. Write An SQL Query To Fetch “FIRST_NAME” From Worker Table In Upper Case alias as WORKER_FIRSTNAME.
2. Write An SQL Query To Print The First Three Characters Of FIRST_NAME From Worker Table.
3. Write An SQL Query To Find The Position Of The Alphabet (‘A’) In The First Name Column ‘Amitabh’ From Worker Table.
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.
5. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending And DEPARTMENT Descending.
6. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Contains ‘A’.
7. Write An SQL Query To Print Details Of The Workers Whose FIRST_NAME Ends With ‘A’.
8. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000.
9. Write An SQL Query To Fetch The Count Of Employees Working In The Department ‘Admin’.
10. Write An SQL Query To Fetch The No. Of Workers For Each Department In The Descending Order.
11. Write An SQL Query To Print Details Of The Workers Who Are Also Managers.
12. Write An SQL Query To Show Only Odd Rows From A Table
13. Write An SQL Query To Show Records From One Table That Another Table Does Not Have.
14. Write An SQL Query To Show The Top N (Say 10) Records Of A Table.
15. Write An SQL Query To Fetch The List Of Employees With The Same Salary.

16. Write An SQL Query To Show All Departments Along With The Number Of People Working There.
17. Write An SQL Query To Print The Name Of Employees Having The Highest Salary In Each Department.
18. Write An SQL Query To Fetch Departments Along With The Total Salaries Paid For Each Of Them.
19. Consider the following relations for an order processing database application in a company.
CUSTOMER (Cust #: int, Cname: string, City: string)
ORDER (Order #: int, Odate: date, Cust #: int, Ord-Amt: int) ORDER-ITEM (Order #: int, Item #: int, qty: int)
ITEM (Item #: int, Unit Price: int)
SHIPMENT (Order #: int, Warehouse #: int, Ship-Date: date) WAREHOUSE (Warehouse #: int, City: string)
i) Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.
iii) Produce a listing: CUSTNAME, NO_OF_ORDERS, AVG_ORDER_AMT, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer
iv) List the Order# for the orders that were shipped from all the warehouses that the company has in a specific city
v) Demonstrate how you delete Item# 10 from the ITEM table and make that field null in the ORDER- ITEM table.
20. Create a table Emp(e_no, e_name, e_phone, e_addr,e_salary) to store records of 10 employees:
i) Alter the data type of e_no from number to varchar
ii) Alter table by setting e_no as primary key
iii) Alter table by adding a column e_pin
iv) Update the phone number of an employee in the table
21. Create a table Dept(dept_no, dept_name,e_no, dept_loc,dept_hod) to store records of 10 departments:
i) Create the reference between Emp and Dept table with e_no attribute.
ii) Assign dept_no as primary key.
iii) Update the dept_hod for one department.
iv) Delete one department.
22. Solve the following queries
i) Write a query to find the employee name and dept_hod whose dept_hod is SAY, “John”.
ii) Write a query to find the average salary of the employee of CSE department.

```
use assignment_6;
create table worker (worker_id int(10),first_name varchar(20),last_name varchar(10),salary int(10),joining_date datetime,department varchar(10));
insert into worker values(001,'Momota','Didi',100000,'2014-02-20 09:00:00','HR');
insert into worker values(002,'Modi','Da',70000,'2014-06-11 09:00:00','Admin');
insert into worker values(003,'Kakajibonfaka','Khelahobe',300000,'2014-02-20 09:00:00','HR');
insert into worker values(004,'Murturswami','Iyer',500000,'2014-02-20 09:00:00','Admin');
insert into worker values(005,'Kesta','Da',500000,'2014-06-11 09:00:00','Admin');
insert into worker values(006,'Chagla','Da',80000,'2014-06-11 09:00:00','Account');
insert into worker values(007,'Pagla','Di',270000,'2014-01-20 09:00:00','Account');
insert into worker values(008,'Netaji','Bose',50000,'2014-06-09 09:00:00','Admin');
select * from worker;
create table bonus(Worker_Ref_Id int(10),Bonus_date datetime,Bonus_Amount int(20));
```

```
insert into bonus values(1,'2016-02-20 00:00:00',5000);
insert into bonus values(2,'2016-06-11 00:00:00',5000);
insert into bonus values(3,'2016-02-20 00:00:00',5000);
insert into bonus values(1,'2016-02-20 00:00:00',5000);
insert into bonus values(2,'2016-06-11 00:00:00',5000);
create table title(Worker_Ref_Id int(10),Worker_Title varchar(20),Affected_from datetime);
insert into title values(1,'Manager','2016-02-20 00:00:00');
insert into title values(2,'Executive','2016-06-11 00:00:00');
insert into title values(8,'Executive','2016-06-11 00:00:00');
insert into title values(5,'Manager','2016-06-11 00:00:00');
insert into title values(4,'Asst. Manager','2016-06-11 00:00:00');
insert into title values(7,'Executive','2016-06-11 00:00:00');
insert into title values(6,'Lead','2016-06-11 00:00:00');
insert into title values(3,'Lead','2016-06-11 00:00:00');
select first_name as Worker_Firstname from worker;
select substring(first_name,1,3) from worker;
select position("A" in first_name) from worker where first_name="Amitabh";
select instr(first_name,"A") from worker where first_name="Amitabh";
select concat(first_name,' ',last_name) as compile_name from worker;
select * from worker order by first_name ASC,department DESC;
select * from worker where first_name like "%a%";
select * from worker where first_name like "%a";
select * from worker where salary between 100000 and 500000;
select count(*) from worker where department="Admin";
select department,count(worker_id) as No_of_workers from worker group by department order by
No_of_workers DESC;
select distinct w.first_name,t.worker_title from worker w insert join title t on
w.worker_id=t.Worker_Ref_ID and t.worker_title in ('Manager');
select * from worker where MOD(worker_id,2)<>0;
select * from worker MINUS select * from title;
select * from worker order by salary DESC limit 10;
select distinct w.worker_id,w.first_name,w.salary from worker w, worker w1 where w.salary and
w.worker_id!=w1.worker_id;
select department,count(department) as 'number of workers' from worker group by department;
select t.department,t.first_name,t.salary from (select max(salary) as totalsalary,department from
worker group by department) as tempnew inner join worker t on
tempnew.department=t.department and tempnew.totalsalary=t.salary;
select department,sum(salary) from worker group by department;
```

ASSIGNMENT 8

1. The following tables are maintained by a book dealer
AUTHOR(author-id: int, name: string, city: string, country: string)
PUBLISHER(publisher-id: int name: string, city: string, country: string)

CATLOG(book-id: int, title : string, author-id: int, publisher-id: int, category: int, year: int, price: int)
CATEGORY(category-id: int, description: string)
ORDER-DETAILS(order-no: int, book-id: int, quantity: int)
i) Create above tables by properly specifying the primary keys and the foreign keys.
ii) Enter atleast five tuples for each relation.
iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books is
greater than the average price of the books in the catalog and the year of publication is after 2010.
iv) Find the author of the book which has maximum sales.
v) Demonstrate how to increase price of books published by specific publisher by 10%
2. Consider the following database for BANK.
BRANCH(branch-name: string, branch-city: string, assets: real)
ACCOUNT(accno: int, branch-name: string, balance: real)
DEPOSITOR(customer-name: string, accno: int)
CUSTOMER(customer-name: string, customer-street: string, customer-city: string)
LOAN(loan-no: int, branch-name: string, amount: real)
BORROWER(customer-name: string, loan-no: int)
i) Create the above tables by properly specifying the primary keys and foreign keys.
ii) Enter atleast five tuples for each relation.
iii) Find all the customers who have atleast two accounts at the main branch.
iv) Find all customer who have an account at all the branches located in a specific city.
v) Demonstrate how t0 delete all account tuples at every branch located in specific city.
3. Consider the following database for ORDER PROCESSING.
CUSTOMER(cust-no: int, cname: string, city: string)
ORDER(orderno: int, odate: date, ord-amt: real)
ORDER_ITEM(orderno: int, itemno:int, qty: int)
ITEM(itemno: int, unitprice: real)
SHIPMENT(orderno: int, warehouseno: int, ship-date: date)
WAREHOUSE(warehouseno: int, city: string)
i) Create the above tables by properly specifying the primary keys and the foreign keys
ii) Enter atleast five tuples for each relation.
iii) List the order number and ship date for all orders shipped from particular warehouse
iv) Produce a listing: customer name, no of orders, average order amount
v) List the orders that were not shipped within 30 days of ordering
University of Engineering & Managemen

```
use assignment_8;
```

1)iii)

```
select * from author1 where author1_id in (select author_id from catalogue1 where year>2000
and price>(select avg(price) from catalogue1) group by author1_id having count(*)>1;
```

iv)

```
select author1_name from author1 a.catalogue1 e where a.author1_id=e.author1_id and book_id
in (select book_id from order details 1 where quantity=(select max(quantity) from order details1);
```

v)

```
update catalogue1 set price=1.1* price where publisher1_id in (select publisher1_id from
publisher1 where publisher1_name='person');
```

2)iii)

```
select customer_name from depositor where acc_no from account branch_name='main') group
by acc_no having count(acceno)>2;
```

iv)

```
select customer_name from customer where acc_no in (select acc_no from account where
branch_name in (select branch_name from branch where branch_city like 'Bangalore'));
```

v)

```
Delete acc_no from account where branch_name in (select branch_name,branch_city from branch where branch_city like 'Bangalore' group by branch_city);
```

3)iii)

```
select c.custno, count(*) as no-of_orders,avg(o,ord_amt) as avg_order_amount from customer c, order1 o where o.customer=c.custno group by c.custno;
```

iv)

```
select s.orderno from shipment s, warehouse w where s.warehouse=w.warehouse and w.city='Bangalore' group by orderno having count(*) and not count(*)=0;
```

v)

```
delete from item where itemno=3;
```

ASSIGNMENT 9

Rules: set serveroutput on

```
edit test1
  To run : @test1
```

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

```
declare
  a number;
  b number;
  c number;
begin
  a:=&a;
  b:=&b;
  c:=&c;
if (a>b and a>c) then
  dbms_output.put_line(‘a is largest’ || a);
elsif (b>a and b>c) then
dbms_output.put_line(‘b is largest’ || b);
else
dbms_output.put_line(‘c is the largest’||c);
endif;
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;
```

Assignment - 2 , Assignment - 5 (another pdf)

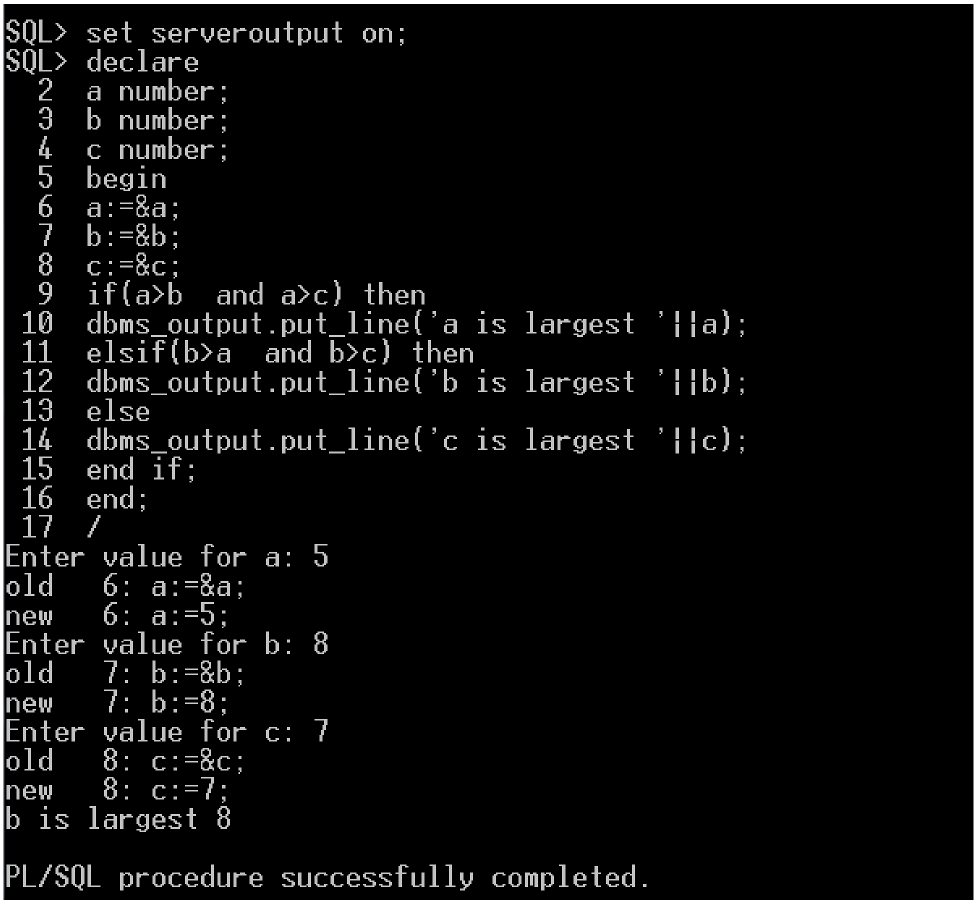
Assignment - 7

Assignment - 9

Added here

Assignment 9

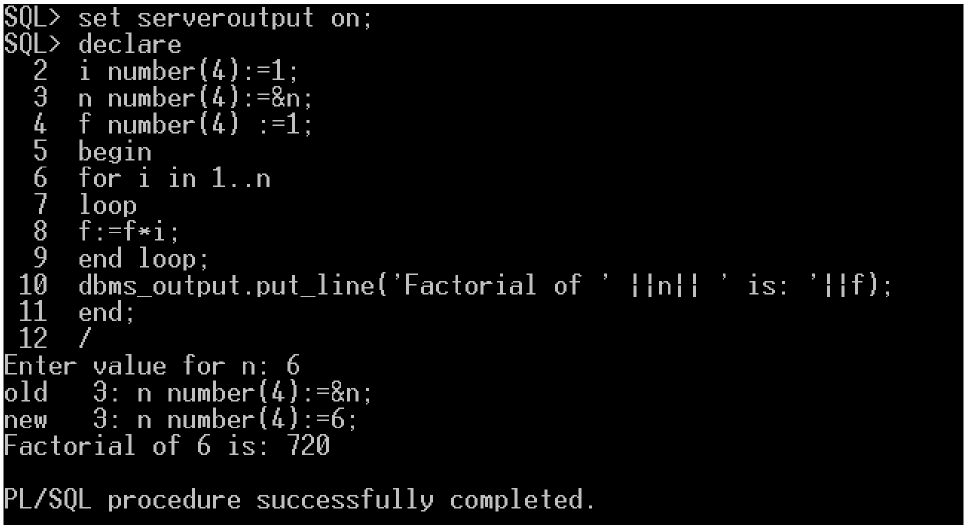
1



```
SQL> set serveroutput on;
SQL> declare
  2  a number;
  3  b number;
  4  c number;
  5  begin
  6  a:=&a;
  7  b:=&b;
  8  c:=&c;
  9  if(a>b  and a>c) then
 10  dbms_output.put_line('a is largest '||a);
 11  elsif(b>a  and b>c) then
 12  dbms_output.put_line('b is largest '||b);
 13  else
 14  dbms_output.put_line('c is largest '||c);
 15  end if;
 16  end;
 17  /
Enter value for a: 5
old   6: a:=&a;
new   6: a:=5;
Enter value for b: 8
old   7: b:=&b;
new   7: b:=8;
Enter value for c: 7
old   8: c:=&c;
new   8: c:=7;
b is largest 8

PL/SQL procedure successfully completed.
```

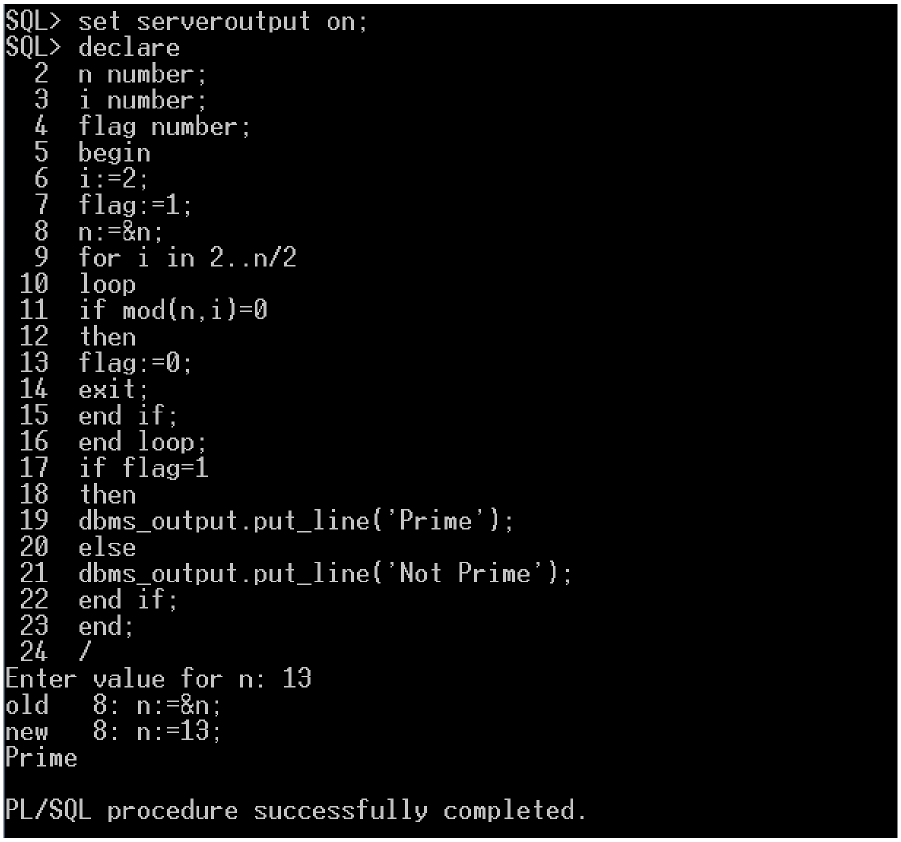
3



```
SQL> set serveroutput on;
SQL> declare
  2  i number(4):=1;
  3  n number(4):=&n;
  4  f number(4) :=1;
  5  begin
  6  for i in 1..n
  7  loop
  8  f:=f*i;
  9  end loop;
 10  dbms_output.put_line('Factorial of ' ||n|| ' is: '||f);
 11  end;
 12  /
Enter value for n: 6
old   3: n number(4):=&n;
new   3: n number(4):=6;
Factorial of 6 is: 720

PL/SQL procedure successfully completed.
```

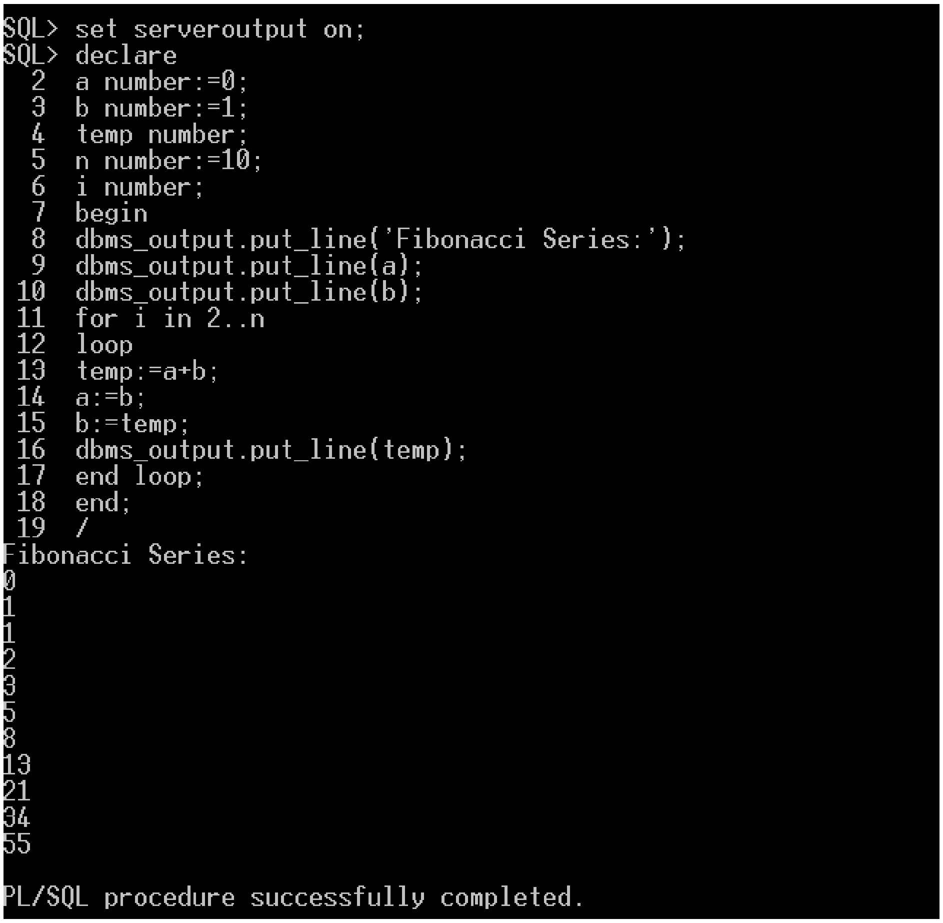
4



```
SQL> set serveroutput on;
SQL> declare
  2  n number;
  3  i number;
  4  flag number;
  5  begin
  6  i:=2;
  7  flag:=1;
  8  n:=&n;
  9  for i in 2..n/2
 10  loop
 11  if mod(n,i)=0
 12  then
 13  flag:=0;
 14  exit;
 15  end if;
 16  end loop;
 17  if flag=1
 18  then
 19  dbms_output.put_line('Prime');
 20  else
 21  dbms_output.put_line('Not Prime');
 22  end if;
 23  end;
 24  /
Enter value for n: 13
old   8: n:=&n;
new   8: n:=13;
Prime

PL/SQL procedure successfully completed.
```

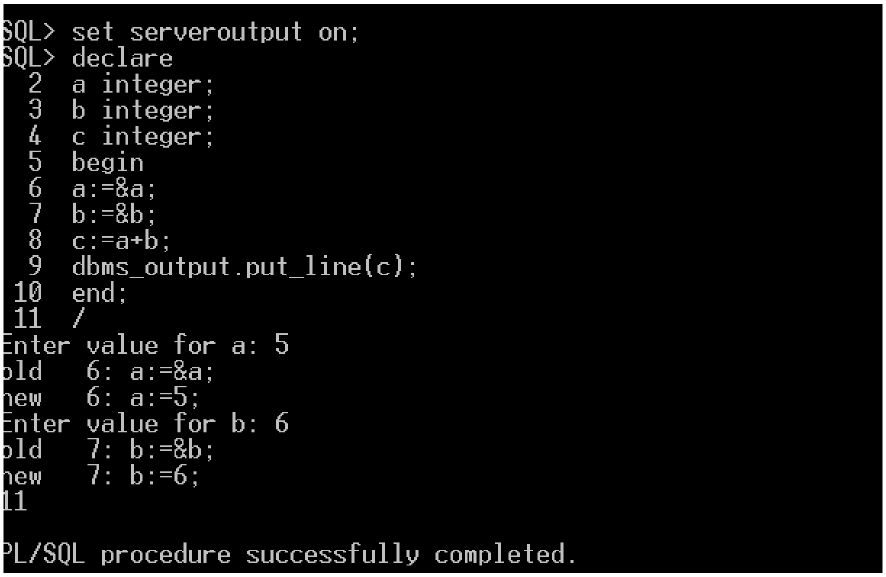
5



```
SQL> set serveroutput on;
SQL> declare
  2  a number:=0;
  3  b number:=1;
  4  temp number;
  5  n number:=10;
  6  i number;
  7  begin
  8  dbms_output.put_line('Fibonacci Series:');
  9  dbms_output.put_line(a);
 10  dbms_output.put_line(b);
 11  for i in 2..n
 12  loop
 13  temp:=a+b;
 14  a:=b;
 15  b:=temp;
 16  dbms_output.put_line(temp);
 17  end loop;
 18  end;
 19  /
Fibonacci Series:
0
1
1
2
3
5
8
13
21
34
55

PL/SQL procedure successfully completed.
```

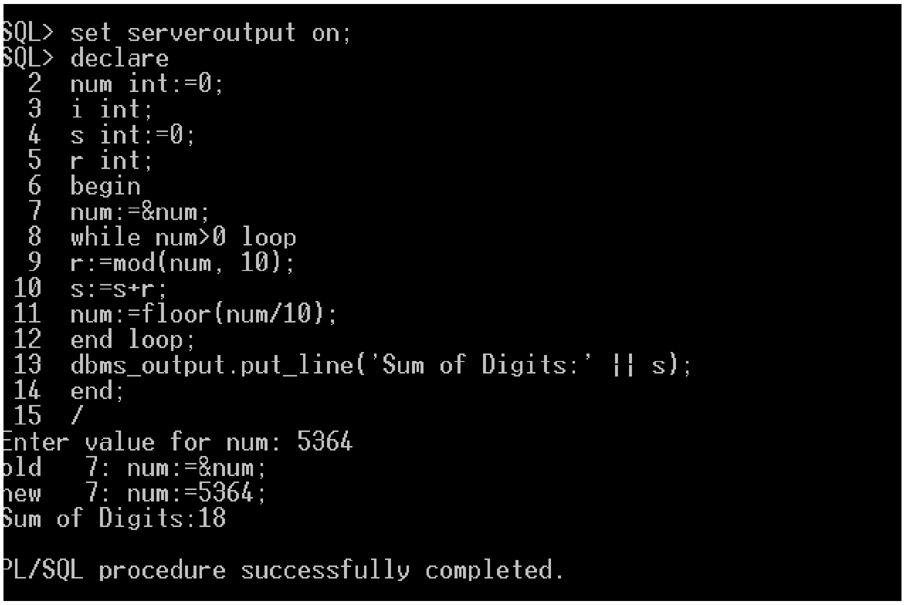
6



```
SQL> set serveroutput on;
SQL> declare
  2  a integer;
  3  b integer;
  4  c integer;
  5  begin
  6  a:=&a;
  7  b:=&b;
  8  c:=a+b;
  9  dbms_output.put_line(c);
 10  end;
 11  /
Enter value for a: 5
old   6: a:=&a;
new   6: a:=5;
Enter value for b: 6
old   7: b:=&b;
new   7: b:=6;
11

PL/SQL procedure successfully completed.
```

8



```
SQL> set serveroutput on;
SQL> declare
  2  num int:=0;
  3  i int;
  4  s int:=0;
  5  r int;
  6  begin
  7  num:=&num;
  8  while num>0 loop
  9  r:=mod(num, 10);
 10  s:=s+r;
 11  num:=floor(num/10);
 12  end loop;
 13  dbms_output.put_line('Sum of Digits:' || s);
 14  end;
 15  /
nter value for num: 5364
old   7: num:=&num;
new   7: num:=5364;
Sum of Digits:18

PL/SQL procedure successfully completed.
```

Assignment-7

1) SQL) create table Job_History (Emp-id number(2), start_date date, end-date date, job-type varchar(20), D-name varchar(20));

2) SQL) select job-types from employees;

3) SQL) select ~~job~~ department, job-type from employees;

4) SQL) select emp_id, job-type from employees where previous i= current-id;

5) SQL) select emp-name ~~frome~~ from employeeswhere change = 'new job';

6) SQL) select employee * from employees as employee join salary as salary an salary . empna = employee. MGR where employee .sal > chitrasal;

7) SQL) select emp_id, ename from employee where emp_id = 7 join job-History as H where job_type = "same";

8) SQL) select emp-id, ename from employeeswhere salary ~~Q~~> 10000 and salary < 70000;

9) SQL) select ~~emp~~ lname, job-type, salary ~~of~~ from employeeswhere salary =35000;

10) SQL) select job-type from employees where salary = 10,000;

11) SQL) select department from employees where salary > sales;

12) SQL) select * from employees where salary =(select min(salary) from employees);

13) SQL) select * from employees where salary > all (select avg(salar from employeesgroup by department-id = "Engineer";

Assignment No -6

1) SQL) select First_Name as upper(worker_Firstname) from Worker;

2) SQL) select substring (first_name, 1,3) from worker;

3) SQL) select position ("A" in first_name) from worker where first_name
= "Amitabh";

SQL) select instr (first_name, "A") from worker where first_name = "Amitabh";

4) SQL) select concat (first_name, " ", last_name) as complete_name from
worker;

5) SQL) select * from worker order by first_name ASC, department DESC;

6) SQL) select * from worker where first_name like "%a%";

7) SQL) select * from worker where first_name like "%a";

8) SQL) select * from worker where salary between 100000 AND
500000;

9) SQL) select count (*) from worker where department = "Admin";

10) SQL) select department, count (worker_id) No_of_workers from worker
group by department, order by no_of_workers DESC;

11) SQL) select distinct w.first_name, t.worker_title from worker w
insert join title t on w.worker_id = t.worker_ref_id ~~and~~
AND t.worker_title in ('Manager');

12) SQL) select * from worker where MOD (worker_id, 2)<>0;

13) SQL) select * from worker ~~minus~~ MINUS select * from title;

14) SQL) select * from worker order by salary DESC limit 10;

15) SQL) select distinct w.worker_id, w.first_name, w.salary from
worker w, worker w1 where w.salary = w1.salary and w.worker_
id != w1.worker_id;

16) SQL) select department, count (department) as 'number of workers'
from worker group by department;

```
17) SQL> select t.department, t.first_name, t.salary from (~~sel~~ select max
    (salary) as totalsalary, department from worker group by departme
    as tempnew inner jain worker t on tempnew.department = t.
    department and tempnew.totalsalary = t.salary;

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