### HR Schema Queries

1. Display details of jobs where the minimum salary is greater than 10000.

SELECT \* FROM JOBS WHERE MIN\_SALARY > 10000

1. Display the first name and join date of the employees who joined between 2002 and 2005.

SELECT FIRST\_NAME, HIRE\_DATE FROM EMPLOYEES

WHERE TO\_CHAR(HIRE\_DATE, 'YYYY') BETWEEN 2002 AND 2005 ORDER BY HIRE\_DATE

1. Display first name and join date of the employees who is either IT Programmer or Sales Man.

SELECT FIRST\_NAME, HIRE\_DATE

FROM EMPLOYEES WHERE JOB\_ID IN ('IT\_PROG', 'SA\_MAN')

1. Display employees who joined after 1st January 2008.

SELECT \* FROM EMPLOYEES  where hire\_date > '01-jan-2008'

1. Display details of employee with ID 150 or 160.

SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID in (150,160)

1. Display first name, salary, commission pct, and hire date for employees with salary less than 10000.

SELECT FIRST\_NAME, SALARY, COMMISSION\_PCT, HIRE\_DATE FROM EMPLOYEES WHERE SALARY < 10000

1. Display job Title, the difference between minimum and maximum salaries for jobs with max salary in the range 10000 to 20000.

SELECT JOB\_TITLE, MAX\_SALARY-MIN\_SALARY DIFFERENCE FROM JOBS WHERE MAX\_SALARY BETWEEN 10000 AND 20000

1. Display first name, salary, and round the salary to thousands.

SELECT FIRST\_NAME, SALARY, ROUND(SALARY, -3) FROM EMPLOYEES

1. Display details of jobs in the descending order of the title.

SELECT \* FROM JOBS ORDER BY JOB\_TITLE

1. Display employees where the first name or last name starts with S.

SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE  FIRST\_NAME  LIKE 'S%' OR LAST\_NAME LIKE 'S%'

1. Display employees who joined in the month of May.

SELECT \* FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE, 'MON')= 'MAY'

1. Display details of the employees where commission percentage is null and salary in the range 5000 to 10000 and department is 30.

SELECT \* FROM EMPLOYEES WHERE COMMISSION\_PCT IS NULL AND SALARY BETWEEN 5000 AND 10000 AND DEPARTMENT\_ID=30

1. Display first name and date of first salary of the employees.

SELECT FIRST\_NAME, HIRE\_DATE, LAST\_DAY(HIRE\_DATE)+1 FROM EMPLOYEES

1. Display first name and experience of the employees.

SELECT FIRST\_NAME, HIRE\_DATE, FLOOR((SYSDATE-HIRE\_DATE)/365)FROM EMPLOYEES

1. Display first name of employees who joined in 2001.

SELECT FIRST\_NAME, HIRE\_DATE FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE, 'YYYY')=2001

1. Display first name and last name after converting the first letter of each name to upper case and the rest to lower case.

SELECT INITCAP(FIRST\_NAME), INITCAP(LAST\_NAME) FROM EMPLOYEES

1. Display the first word in job title.

SELECT JOB\_TITLE,  SUBSTR(JOB\_TITLE,1, INSTR(JOB\_TITLE, ' ')-1) FROM JOBS

1. Display the length of first name for employees where last name contain character ‘b’ after 3rd position.

SELECT FIRST\_NAME, LAST\_NAME FROM EMPLOYEES WHERE INSTR(LAST\_NAME,'B') > 3

1. Display first name in upper case and email address in lower case for employees where the first name and email address are same irrespective of the case.

SELECT UPPER(FIRST\_NAME), LOWER(EMAIL) FROM EMPLOYEES WHERE UPPER(FIRST\_NAME)= UPPER(EMAIL)

1. Display employees who joined in the current year.

SELECT \* FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE,'YYYY')=TO\_CHAR(SYSDATE, 'YYYY')

1. Display the number of days between system date and 1st January 2011.

SELECT SYSDATE - to\_date('01-jan-2011') FROM DUAL

1. Display how many employees joined in each month of the current year.

SELECT TO\_CHAR(HIRE\_DATE,'MM'), COUNT (\*) FROM EMPLOYEES

WHERE TO\_CHAR(HIRE\_DATE,'YYYY')= TO\_CHAR(SYSDATE,'YYYY') GROUP BY TO\_CHAR(HIRE\_DATE,'MM')

1. Display manager ID and number of employees managed by the manager.

 SELECT MANAGER\_ID, COUNT(\*) FROM EMPLOYEES GROUP BY MANAGER\_ID

1. Display employee ID and the date on which he ended his previous job.

 SELECT EMPLOYEE\_ID, MAX(END\_DATE) FROM JOB\_HISTORY GROUP BY EMPLOYEE\_ID

1. Display number of employees joined after 15th of the month.

SELECT COUNT(\*) FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE,'DD') > 15

1. Display the country ID and number of cities we have in the country.

SELECT COUNTRY\_ID,  COUNT(\*)  FROM LOCATIONS GROUP BY COUNTRY\_ID

1. Display average salary of employees in each department who have commission percentage.

SELECT DEPARTMENT\_ID, AVG(SALARY) FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NOT NULL GROUP BY DEPARTMENT\_ID

1. Display job ID, number of employees, sum of salary, and difference between highest salary and lowest salary of the employees of the job.

SELECT JOB\_ID, COUNT(\*), SUM(SALARY), MAX(SALARY)-MIN(SALARY) SALARY FROM EMPLOYEES GROUP BY JOB\_ID

1. Display job ID for jobs with average salary more than 10000.

SELECT JOB\_ID, AVG(SALARY) FROM EMPLOYEES

GROUP BY JOB\_ID

HAVING AVG(SALARY)>10000

1. Display years in which more than 10 employees joined.

SELECT TO\_CHAR(HIRE\_DATE,'YYYY') FROM EMPLOYEES

GROUP BY TO\_CHAR(HIRE\_DATE,'YYYY')

HAVING COUNT(EMPLOYEE\_ID) > 10

1. Display departments in which more than five employees have commission percentage.

SELECT DEPARTMENT\_ID FROM EMPLOYEES

WHERE COMMISSION\_PCT IS NOT NULL

GROUP BY DEPARTMENT\_ID

HAVING COUNT(COMMISSION\_PCT)>5

1. Display employee ID for employees who did more than one job in the past.

SELECT EMPLOYEE\_ID FROM JOB\_HISTORY GROUP BY EMPLOYEE\_ID HAVING COUNT(\*) > 1

1. Display job ID of jobs that were done by more than 3 employees for more than 100 days.

SELECT JOB\_ID FROM JOB\_HISTORY

WHERE END\_DATE-START\_DATE > 100

GROUP BY JOB\_ID

HAVING COUNT(\*)>3

1. Display department ID, year, and Number of employees joined.

SELECT DEPARTMENT\_ID, TO\_CHAR(HIRE\_DATE,'YYYY'), COUNT(EMPLOYEE\_ID)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID, TO\_CHAR(HIRE\_DATE, 'YYYY')

ORDER BY DEPARTMENT\_ID

1. Display departments where any manager is managing more than 5 employees.

SELECT DISTINCT DEPARTMENT\_ID

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID, MANAGER\_ID

HAVING COUNT(EMPLOYEE\_ID) > 5

1. Change salary of employee 115 to 8000 if the existing salary is less than 6000.

UPDATE EMPLOYEES SET SALARY = 8000 WHERE EMPLOYEE\_ID = 115 AND SALARY < 6000

1. Insert a new employee into employees with all the required details.

INSERT INTO EMPLOYEES  (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE,JOB\_ID, SALARY, DEPARTMENT\_ID)

VALUES (207, 'ANGELA', 'SNYDER','ANGELA','215 253 4737', SYSDATE, 'SA\_MAN', 12000, 80)

1. Delete department 20.

DELETE FROM DEPARTMENTS WHERE DEPARTMENT\_ID=20

1. Change job ID of employee 110 to IT\_PROG if the employee belongs to department 10 and the existing job ID does not start with IT.

UPDATE EMPLOYEES SET JOB\_ID= 'IT\_PROG'

WHERE EMPLOYEE\_ID=110 AND DEPARTMENT\_ID=10 AND NOT JOB\_ID LIKE 'IT%'

1. Insert a row into departments table with manager ID 120 and location ID in any location ID for city Tokyo.

INSERT INTO DEPARTMENTS VALUES(150,'SPORTS',120,1200)

1. Display department name and number of employees in the department.

SELECT DEPARTMENT\_NAME, COUNT(\*) FROM EMPLOYEES NATURAL JOIN DEPARTMENTS GROUP BY DEPARTMENT\_NAME

1. Display job title, employee ID, number of days between ending date and starting date for all jobs in department 30 from job history.

SELECT EMPLOYEE\_ID, JOB\_TITLE, END\_DATE-START\_DATE DAYS

FROM JOB\_HISTORY NATURAL JOIN JOBS

WHERE DEPARTMENT\_ID=30

1. Display department name and manager first name.

SELECT DEPARTMENT\_NAME, FIRST\_NAME FROM DEPARTMENTS D JOIN EMPLOYEES E ON (D.MANAGER\_ID=E.EMPLOYEE\_ID)

1. Display department name, manager name, and city.

SELECT DEPARTMENT\_NAME, FIRST\_NAME, CITY FROM DEPARTMENTS D JOIN EMPLOYEES E ON (D.MANAGER\_ID=E.EMPLOYEE\_ID) JOIN LOCATIONS L USING (LOCATION\_ID)

1. Display country name, city, and department name.

SELECT COUNTRY\_NAME, CITY, DEPARTMENT\_NAME

FROM COUNTRIES JOIN LOCATIONS USING (COUNTRY\_ID)

JOIN DEPARTMENTS USING (LOCATION\_ID)

1. Display job title, department name, employee last name, starting date for all jobs from 2000 to 2005.

SELECT JOB\_TITLE, DEPARTMENT\_NAME, LAST\_NAME, START\_DATE

FROM JOB\_HISTORY JOIN JOBS USING (JOB\_ID) JOIN DEPARTMENTS

USING (DEPARTMENT\_ID) JOIN  EMPLOYEES USING (EMPLOYEE\_ID)

WHERE TO\_CHAR(START\_DATE,'YYYY') BETWEEN 2000 AND 2005

1. Display job title and average salary of employees

SELECT JOB\_TITLE, AVG(SALARY) FROM EMPLOYEES

NATURAL JOIN JOBS GROUP BY JOB\_TITLE

1. Display job title, employee name, and the difference between maximum salary for the job and salary of the employee.

SELECT JOB\_TITLE, FIRST\_NAME, MAX\_SALARY-SALARY DIFFERENCE FROM EMPLOYEES NATURAL JOIN JOBS

1. Display last name, job title of employees who have commission percentage and belongs to department 30.

SELECT JOB\_TITLE, FIRST\_NAME, MAX\_SALARY-SALARY DIFFERENCE FROM EMPLOYEES NATURAL JOIN JOBS WHERE DEPARTMENT\_ID  = 30

1. Display details of jobs that were done by any employee who is currently drawing more than 15000 of salary.

SELECT JH.\*

FROM  JOB\_HISTORY JH JOIN EMPLOYEES E ON (JH.EMPLOYEE\_ID = E.EMPLOYEE\_ID)

WHERE SALARY > 15000

1. Display department name, manager name, and salary of the manager for all managers whose experience is more than 5 years.

SELECT DEPARTMENT\_NAME, FIRST\_NAME, SALARY

FROM DEPARTMENTS D JOIN EMPLOYEES E ON (D.MANAGER\_ID=E.MANAGER\_ID)

WHERE  (SYSDATE-HIRE\_DATE) / 365 > 5

1. Display employee name if the employee joined before his manager.

SELECT FIRST\_NAME FROM  EMPLOYEES E1 JOIN EMPLOYEES E2 ON (E1.MANAGER\_ID=E2.EMPLOYEE\_ID)

WHERE E1.HIRE\_DATE < E2.HIRE\_DATE

1. Display employee name, job title for the jobs employee did in the past where the job was done less than six months.

SELECT FIRST\_NAME, JOB\_TITLE FROM EMPLOYEES E JOIN JOB\_HISTORY  JH ON (JH.EMPLOYEE\_ID = E.EMPLOYEE\_ID) JOIN JOBS J  ON( JH.JOB\_ID = J.JOB\_ID)

WHERE  MONTHS\_BETWEEN(END\_DATE,START\_DATE)  < 6

1. Display employee name and country in which he is working.

SELECT FIRST\_NAME, COUNTRY\_NAME FROM EMPLOYEES JOIN DEPARTMENTS USING(DEPARTMENT\_ID)

JOIN LOCATIONS USING( LOCATION\_ID)

JOIN COUNTRIES USING ( COUNTRY\_ID)

1. Display department name, average salary and number of employees with commission within the department.

SELECT DEPARTMENT\_NAME, AVG(SALARY), COUNT(COMMISSION\_PCT)

FROM DEPARTMENTS JOIN EMPLOYEES USING (DEPARTMENT\_ID)

GROUP BY DEPARTMENT\_NAME

1. Display the month in which more than 5 employees joined in any department located in Sydney.

SELECT TO\_CHAR(HIRE\_DATE,'MON-YY')

FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT\_ID) JOIN  LOCATIONS USING (LOCATION\_ID)

WHERE  CITY = 'Seattle'

GROUP BY TO\_CHAR(HIRE\_DATE,'MON-YY')

HAVING COUNT(\*) > 5

1. Display details of departments in which the maximum salary is more than 10000.

SELECT \* FROM DEPARTMENTS WHERE DEPARTMENT\_ID IN

( SELECT DEPARTMENT\_ID FROM EMPLOYEES

  GROUP BY DEPARTMENT\_ID

  HAVING MAX(SALARY)>10000)

1. Display details of departments managed by ‘Smith’.

SELECT \* FROM DEPARTMENTS WHERE MANAGER\_ID IN

  (SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE FIRST\_NAME='SMITH')

1. Display jobs into which employees joined in the current year.

SELECT \* FROM JOBS WHERE JOB\_ID IN

       (SELECT JOB\_ID FROM EMPLOYEES WHERE TO\_CHAR(HIRE\_DATE,'YYYY')=TO\_CHAR(SYSDATE,'YYYY'))

1. Display employees who did not do any job in the past.

SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID NOT IN

       (SELECT EMPLOYEE\_ID FROM JOB\_HISTORY)

1. Display job title and average salary for employees who did a job in the past.

SELECT JOB\_TITLE, AVG(SALARY) FROM JOBS NATURAL JOIN EMPLOYEES

GROUP BY JOB\_TITLE

WHERE EMPLOYEE\_ID IN

    (SELECT EMPLOYEE\_ID FROM JOB\_HISTORY)

1. Display country name, city, and number of departments where department has more than 5 employees.

SELECT COUNTRY\_NAME, CITY, COUNT(DEPARTMENT\_ID)

FROM COUNTRIES JOIN LOCATIONS USING (COUNTRY\_ID) JOIN DEPARTMENTS USING (LOCATION\_ID)

WHERE DEPARTMENT\_ID IN

    (SELECT DEPARTMENT\_ID FROM EMPLOYEES

   GROUP BY DEPARTMENT\_ID

   HAVING COUNT(DEPARTMENT\_ID)>5)

GROUP BY COUNTRY\_NAME, CITY;

1. Display details of manager who manages more than 5 employees.

SELECT FIRST\_NAME FROM EMPLOYEES

WHERE EMPLOYEE\_ID IN

(SELECT MANAGER\_ID FROM EMPLOYEES

 GROUP BY MANAGER\_ID

 HAVING COUNT(\*)>5)

1. Display employee name, job title, start date, and end date of past jobs of all employees with commission percentage null.

SELECT FIRST\_NAME, JOB\_TITLE, START\_DATE, END\_DATE

FROM JOB\_HISTORY JH JOIN JOBS J USING (JOB\_ID) JOIN EMPLOYEES E  ON ( JH.EMPLOYEE\_ID = E.EMPLOYEE\_ID)

WHERE COMMISSION\_PCT IS NULL

1. Display the departments into which no employee joined in last two years.

SELECT  \* FROM DEPARTMENTS

WHERE DEPARTMENT\_ID NOT IN

( SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE FLOOR((SYSDATE-HIRE\_DATE)/365) < 2)

1. Display the details of departments in which the max salary is greater than 10000 for employees who did a job in the past.

SELECT \* FROM DEPARTMENTS

WHERE DEPARTMENT\_ID IN

(SELECT DEPARTMENT\_ID FROM EMPLOYEES

 WHERE EMPLOYEE\_ID IN (SELECT EMPLOYEE\_ID FROM JOB\_HISTORY)

 GROUP BY DEPARTMENT\_ID

 HAVING MAX(SALARY) >10000)

1. Display details of current job for employees who worked as IT Programmers in the past.

SELECT \* FROM JOBS

WHERE JOB\_ID IN

 (SELECT JOB\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID IN

        (SELECT EMPLOYEE\_ID FROM JOB\_HISTORY WHERE JOB\_ID='IT\_PROG'))

1. Display the details of employees drawing the highest salary in the department.

SELECT DEPARTMENT\_ID,FIRST\_NAME, SALARY FROM EMPLOYEES OUTER WHERE SALARY =

    (SELECT MAX(SALARY) FROM EMPLOYEES WHERE DEPARTMENT\_ID = OUTER.DEPARTMENT\_ID)

1. Display the city of employee whose employee ID is 105.

SELECT CITY FROM LOCATIONS WHERE LOCATION\_ID =

    (SELECT LOCATION\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_ID =

                 (SELECT DEPARTMENT\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID=105)

  )

1. Display third highest salary of all employees

select salary

from employees main

where  2 = (select count( distinct salary )

            from employees

            where  salary > main.salary)

    71. Display nth highest salary (replace number 5 with required number)

select max(salary) from hr.employees  
where salary < (select min(salary) from (select salary from (select distinct salary from hr.employees order by salary desc) where rownum<5));

**PL/SQL Programs**

1. Write a program to interchange the salaries of employee 120 and 122.

Declare

   V\_salary\_120   employees.salary%type;

Begin

  Select  salary into v\_salary\_120

  From employees where  employee\_id = 120;

  Update employees set salary  = ( select salary from employees where employee\_id = 122)

  Where employee\_id = 120;

  Update employees set salary  =  v\_salary\_120  Where employee\_id = 122;

  Commit;

End;

1. Increase the salary of employee 115 based on the following conditions: If experience is more than 10 years, increase salary by 20% If experience is greater than 5 years, increase salary by 10% Otherwise 5% Case by Expression:

declare

    v\_exp  number(2);

    v\_hike number(5,2);

begin

    select  floor((sysdate-hire\_date) / 365 ) into v\_exp

    from employees

    where employee\_id = 115;

    v\_hike := 1.05;

    case

      when  v\_exp > 10 then

            v\_hike := 1.20;

      when  v\_exp > 5  then

            v\_hike := 1.10;

    end case;

    update employees set salary = salary \* v\_hike

    where employee\_id = 115;

end;

1. Change commission percentage as follows for employee with ID = 150. If salary is more than 10000 then commission is 0.4%, if Salary is less than 10000 but experience is more than 10 years then 0.35%, if salary is less than 3000 then commission is 0.25%. In the remaining cases commission is 0.15%.

declare

    v\_salary  employees.salary%type;

    v\_exp     number(2);

    v\_cp      number(5,2);

begin

    select  v\_salary,  floor ( (sysdate-hire\_date)/365) into v\_salary, v\_exp

    from  employees

    where employee\_id = 150;

    if v\_salary > 10000 then

           v\_cp := 0.4;

    elsif  v\_exp > 10 then

           v\_cp := 0.35;

    elsif  v\_salary < 3000 then

           v\_cp := 0.25;

    else

           v\_cp := 0.15;

    end if;

    update employees set commission\_pct = v\_cp

    where employee\_id = 150;

end;

1. Find out the name of the employee and name of the department for the employee who is managing for employee 103.

declare

    v\_name     employees.first\_name%type;

    v\_deptname departments.department\_name%type;

begin

    select  first\_name , department\_name into v\_name, v\_deptname

    from  employees join departments using (department\_id)

    where employee\_id = ( select manager\_id from employees    where employee\_id = 103);

    dbms\_output.put\_line(v\_name);

    dbms\_output.put\_line(v\_deptname);

end;

1. Display missing employee IDs.

declare

     v\_min  number(3);

     v\_max  number(3);

     v\_c    number(1);

begin

     select min(employee\_id), max(employee\_id) into v\_min, v\_max

     from employees;

     for i in  v\_min + 1 .. v\_max - 1

     loop

           select count(\*) into v\_c

           from employees

           where employee\_id = i;

           if  v\_c = 0 then

                dbms\_output.put\_line(i);

           end if;

    end loop;

end;

1. Display the year in which maximum number of employees joined along with how many joined in each month in that year.

declare

      v\_year  number(4);

      v\_c     number(2);

begin

      select  to\_char(hire\_date,'yyyy') into v\_year

      from  employees

      group by to\_char(hire\_date,'yyyy')

      having count(\*) =

             ( select  max( count(\*))

               from  employees

               group by to\_char(hire\_date,'yyyy'));

      dbms\_output.put\_line('Year : ' || v\_year);

      for month in 1 .. 12

      loop

          select  count(\*) into v\_c

          from employees

          where  to\_char(hire\_date,'mm') = month and to\_char(hire\_date,'yyyy') = v\_year;

          dbms\_output.put\_line('Month : ' || to\_char(month) || ' Employees : ' || to\_char(v\_c));

     end loop;

end;

1. Change salary of employee 130 to the salary of the employee with first name ‘Joe’. If Joe is not found then take average salary of all employees. If more than one employee with first name ‘Joe’ is found then take the least salary of the employees with first name Joe.

declare

    v\_salary  employees.salary%type;

begin

     select salary into v\_salary

     from employees where first\_name = 'Joe';

     update employees set salary = v\_salary

     where employee\_id = 130;

exception

     when no\_data\_found then

       update employees set salary = (select avg(salary) from employees)

     where employee\_id = 130;

end;

1. Display Job Title and Name of the Employee who joined the job first day.

declare

       cursor  jobscur is select  job\_id, job\_title from jobs;

     v\_name  employees.first\_name%type;

begin

       for jobrec in jobscur

     loop

           select first\_name into v\_name

             from employees

             where hire\_date = ( select min(hire\_date) from employees where job\_id = jobrec.job\_id)

                  and  job\_id = jobrec.job\_id;

             dbms\_output.put\_line( jobrec.job\_title || '-' || v\_name);

     end loop;

end;

1. Display 5th and 10th employees in Employees table.

declare

     cursor empcur is

        select employee\_id, first\_name

        from employees;

begin

     for emprec  in empcur

     loop

         if empcur%rowcount > 4 then

              dbms\_output.put\_line( emprec.first\_name);

              exit  when   empcur%rowcount > 10;

         end if;

     end loop;

end;

1. Update salary of an employee based on department and commission percentage. If department is 40 increase salary by 10%. If department is 70 then 15%, if commission is more than .3% then 5% otherwise 10%.

declare

    cursor empcur is

     select employee\_id, department\_id, commission\_pct

     from employees;

    v\_hike  number(2);

begin

    for emprec in empcur

    loop

         if  emprec.department\_id = 40 then

              v\_hike := 10;

         elsif emprec.department\_id = 70 then

              v\_hike := 15;

         elsif emprec.commission\_pct  > 0.30 then

              v\_hike := 5;

         else

              v\_hike := 10;

         end if;

         update employees set salary = salary + salary \* v\_hike/100

         where employee\_id = emprec.employee\_id;

    end loop;

end;

1. Create a function that takes department ID and returns the name of the manager of the department.

create or replace function get\_dept\_manager\_name(deptid number)

return varchar is

   v\_name  employees.first\_name%type;

begin

   select first\_name into v\_name

   from employees

   where  employee\_id = ( select manager\_id from departments where department\_id = deptid);

   return v\_name;

end;

1. Create a function that takes employee ID and return the number of jobs done by the employee in the past.

create or replace function get\_no\_of\_jobs\_done(empid number)

return number is

   v\_count  number(2);

begin

   select count(\*) into v\_count

   from job\_history

   where  employee\_id = empid;

   return v\_count;

end;

1. Create a procedure that takes department ID and changes the manager ID for the department to the employee in the department with highest salary. (Use Exceptions).

create or replace procedure change\_dept\_manager(deptid number)

is

   v\_empid  employees.employee\_id%type;

begin

   select employee\_id  into v\_empid

   from employees

   where  salary = ( select max(salary) from employees where department\_id = deptid)

     and department\_id = deptid;

   update departments set manager\_id = v\_empid

   where  department\_id = deptid;

end;

1. Create a function that takes a manager ID and return the names of employees who report to this manager. The names must be returned as a string with comma separating names.

create or replace function get\_employees\_for\_manager(manager number)

return varchar2

is

   v\_employees varchar2(1000) := '';

   cursor empcur is

      select  first\_name from employees

      where   manager\_id = manager;

begin

     for emprec in empcur

     loop

         v\_employees :=  v\_employees ||  ',' || emprec.first\_name;

     end loop;

     -- remove extra  , at the beginning

     return  ltrim(v\_employees,',');

end;

1. Ensure no changes can be made to EMPLOYEES table before 6am and after 10pm in a day.

create or replace trigger  trg\_employees\_time\_check

before update or insert or delete

on employees

for each row

begin

   if  to\_char(sysdate,'hh24') < 6 or to\_char(sysdate,'hh24') > 10 then

         raise\_application\_error(-20111,'Sorry! No change can be made before 6 AM and after 10 PM');

   end if;

end;

1. Create a Trigger to ensure the salary of the employee is not decreased.

create or replace trigger  trg\_employees\_salary\_check

before update

on employees

for each row

begin

   if  :old.salary > :new.salary then

         raise\_application\_error(-20111,'Sorry! Salary can not be decreased!');

   end if;

end;

1. Create a trigger to ensure the employee and manager belongs to the same department.

**Note:** This trigger need to read the row that is being modified, which causes mutating problem.  The solution to mutating problem is

explained at: [Trigger with mutating problem](http://mukeshonplsql.blogspot.in/2015/10/trigger-with-mutating-problem-if-you.html) Please check it out.

1. Whenever the job is changed for an employee write the following details into job history. Employee ID, old job ID, old department ID, hire date of the employee for start date, system date for end date. But if a row is already present for employee job history then the start date should be the end date of that row +1.

 create or replace trigger trg\_log\_job\_change

after update of job\_id

on employees

for each row

declare

    v\_enddate   date;

    v\_startdate date;

begin

   -- find out whether the employee has any row in job\_history table

   select max(end\_date) into v\_enddate

   from job\_history

   where employee\_id = :old.employee\_id;

   if v\_enddate is null then

      v\_startdate := :old.hire\_date;

   else

      v\_startdate := v\_enddate + 1;

   end if;

   insert into  job\_history values (:old.employee\_id, v\_startdate, sysdate, :old.job\_id, :old.department\_id);

end;

**Note:** Before testing the above trigger, you need to disable UPDATE\_JOB\_HISTORY trigger, which is already present in HR account, as it does the same.