The following is the structure of the tables provided by Oracle in Human Resource Schema (HR).

Task 1

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

```
select first_name||' '|| last_name full_name, salary, commission_pct, hire_date
from employees
where salary<10000;
```

2. Display city names (without repeated names) in ascending order from locations table.

```
a)  select unique city from locations order by city;
b)  select distinct city from locations order by city;
```

3. Display the first name, hire date and job ID of the employees who is either IT Programmer or Sales Manager, and hired between 2002 and 2005.

```
a)  select first_name, hire_date, job_id
    from employees
    where (job_id like 'SA_MAN' or job_id like 'IT_PROG') and
    (extract(year from hire_date) between 2002 and 2005);

b)  select first_name, hire_date, job_ID from employees
    where (job_ID='IT_PROG' or job_ID='SA_MAN')
    and (hire_date between '01-jan-02' and '31-dec-05') ;
```

4. Display details from jobs table in the descending order of the job title.

```
select * from jobs order by job_title desc;
```

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

```
select * from employees
where commission_pct is null and
salary between 5000 and 10000
and department_id=30;
```

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

```
select * from employees where hire_date>'01-JAN-08'
```

7. Display details of employees with ID 150, 160 or 170.

```
select * from employees where employee_id in (150,160,170);
```

8. 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%';
```

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

```
select length(first_name) from employees where last_name like '___b%'
```

Task 2

1. Write a query in SQL to display job Title, the difference between minimum and maximum salaries for those jobs which max salary within the range 12000 to 18000.

```
select job_title, max_salary-min_salary dif_salary
from jobs where max_salary between 12000 and 18000;
```

2. Display the details of the employees who have no commission percentage and whose salary is within the range 7000 to 12000 for those employees who are not working in the departments 50,30 and 80.

```
select * from employees
where commission_pct is null and
salary between 7000 and 12000 and
department_id not in (30,50,80);
```

3. Write a query in SQL to display the full name (first name and last name), hire date, commission percentage, email and telephone separated by '-', and salary for those employees whose salary is above 11000 and make the result set in a descending order by the full name.

```
select first_name||' '||last_name full_name, hire_date, commission_pct, email,
replace (phone_number,'.','-') phone, salary
from employees
where salary>11000
order by full_name desc;
```

4. Write a query in SQL to display the first and last name, and salary for those employees whose first name is ending with the letter “m” and they have been hired before June 5th, 2010.

```
select first_name, last_name, salary
from employees
where first_name like '%m' and
hire_date<'05-JUN-10';
```

5. Display the full name (first and last), the phone number and email separated by hyphen, and salary, for those employees whose salary is not within the range of 9000 and 17000 and commission is not null. The column headings assign with Full_Name, Contact _Details and Remuneration respectively.

```
select first_name||' '||last_name "Full_Name" ,
phone_number||'-'||email "Contact Details" , salary "Remuneration"
from employees where salary not between 9000 and 17000 and commission_pct is not null;
```

6. Write a query in SQL to display all the information about the department Marketing.

```
select * from departments
where department_name ='Marketing';
```

7. Write a query to display data from job_history and make the result set in descending order by the epmloyee_id and ascending order by start date.

```
select * from job_history order by employee_id desc, start_date;
```

8. Write a query to display job_id and salary of employees whose phone number starts with 515 or 590 and was hired after 2003 by sorting hire_date and salary in ascending way.

```
select job_id, salary, extract(year from hire_date) from employees
where (phone_number like '515%' or phone_number like '590%')
and extract(year from hire_date)>2003
order by hire_date, 2;
```

9. Write a query to display employees who were hired in 2001.

```
a) select first_name, last_name
   from employees where extract (year from hire_date)=2001;

b) select first_name, last_name from employees where hire_date between '01-jan-01'
   and '31-dec-01';
```

10. Write a query to display employees’ first and last name who were not hired in 2006 and 2007.

```
select first_name, last_name
from employees
where extract (year from hire_date) not in (2006,2007);
```

11. Write a query to display email, job_id and first name of employees whose hired year was 2007 or hired month was 1.

```
select first_name, email, job_id
from employees where extract (year from hire_date) = 2007
```

```
or extract (month from hire_date) = 1;
```

12. Write a query to display details of employees who was hired after 2007 or salary is less than 10000.

```
select * from employees
where extract (year from hire_date)> 2007
or salary<10000;
```

Task 3

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

a)
```
select first_name, last_name, hire_date from employees where extract(month from hire_date) = 5;
```

b)
```
select first_name, last_name, hire_date from employees where to_char(hire_date, 'MON')='MAY';
```

2. Display employees who joined in the current year.

```
select first_name, last_name, hire_date from employees where extract(year from hire_date) = extract(year from sysdate);
```

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

```
select round(sysdate-to_date('01/01/2001','dd/mm/yyyy') ) days from dual;
```

4. Display maximum salary of employees.

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

5. Display number of employees in each department.

```
select count(*), department_id from employees group by department_id;
```

6. Display number of employees who joined after 15th of month.

```
select count(*) from employees where extract(day from hire_date)>15;
```

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

```
select round(avg(salary), 2), department_id
from employees
where commission_pct is not null
group by department_id;
```

8. 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
```

9. Display job ID, number of employees, sum of salary, and difference between the highest salary and the lowest salary of the employees for all jobs.

```
select job_id, count(*), sum(salary), max(salary)-min(salary)
from employees
group by job_id;
```

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

```
select manager_id, count(*) from employees
where manager_id is not null
group by manager_id
```

Task 4

1. Show minimum, average and maximum salary in last 15 years according to job id.

```
select min(salary), max(salary), round(avg(salary), 2), job_id
from employees
where extract (year from hire_date)>extract (year from sysdate)-15
group by job_id;
```

2. How many employees hired after 2005 for each department?

```
select count(*), department_id from employees
where extract(year from hire_date)>2005
group by department_id;
```

3. Write a query to show departments in which the difference between maximum and minimum salary is greater than 5000.

```
select department_id, max(salary)-min(salary)
from employees
group by department_id
having max(salary)-min(salary)>5000;
```

4. Display sum salaries of employees who has not commission pct according to departments.

```
select sum(salary), department_id
from employees
```

```
where commission_pct is null
group by department_id;
```

5. How many people has job id with average salary between 3000 and 7000?

```
select count(*), job_id
from employees
group by job_id
having avg(salary) between 3000 and 7000;
```

6. Find number of employees with same name.

```
select count(*), first_name
from employees
group by first_name
having count(first_name)>1;
```

7. How many people with the same phone code work in departments 50 and 90?

```
select count(*), department_id, substr(phone_number,1,3) phone_code
from employees
where department_id in (50,90)
group by department_id, substr(phone_number,1,3)
having count(substr(phone_number,1,3))>1;
```

8. Display departments with sum salary of employees more than 5 in spring and autumn.

```
select sum(salary), department_id
from employees
where extract(month from hire_date) in (3,4,5,9,10,11)
group by department_id
having count(*)>5;
```

9. How many employees work in departments which has maximum salary more than 5000?

```
select count(*), department_id
from employees
group by department_id
having max(salary)>5000;
```

10.Change second letter of employees' names with the last letter and display.

```
select first_name,
substr(first_name,1,1)||substr(first_name,-1,1)||substr(first_name,3,length(first_name)-3)||
substr(first_name,2,1) changed_name
from employees;
```

Task 5

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

```
select last_name, job_title
from employees e
left join jobs j
on e.job_id=j.job_id
where commission_pct is not null
and department_id=30;
```

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

```
select dep.department_name, mng.first_name, mng.salary from employees emp
left join employees mng on emp.manager_id=mng.employee_id
left join departments dep on emp.department_id=dep.department_id
where extract(year from sysdate) - extract(year from mng.hire_date )>5
group by dep.department_name, mng.first_name,mng.salary;
```

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

```
select emp.first_name from employees emp
left join employees mng on mng.employee_id=emp.manager_id
where emp.hire_date<mng.hire_date;
```

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

```
a)  select first_name, job_title from employees emp
    left join jobs j on emp.job_id=j.job_id
    left join job_history jh on j.job_id=jh.job_id
    where end_date-start_date<180;

b)  select first_name, job_title
    from employees emp
    left join jobs jb
    on emp.job_id=jb.job_id
    left join job_history jh
    on jh.job_id=jb.job_id
    where months_between(jh.end_date,jh.start_date)<6;
```

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

```
select department_name, avgs, cnt
from departments dep
```

```
inner join (select department_id, round(avg(salary),2) avgs, count(*) cnt  from employees
where commission_pct>0
group by department_id) agg
on dep.department_id=agg.department_id;
```

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

```
select first_name,country_name
from employees emp
left join departments dep
on emp.department_id=dep.department_id
left join locations loc
on dep.location_id=loc.location_id
left join countries c
on loc.country_id=c.country_id;
```

Task 6

1. Display the first promotion year for each employee.

```
select first_name, min(end_date)
from employees emp
inner join job_history jh
on emp.employee_id=jh.employee_id
group by emp.first_name;
```

2. Display location, city and department name of employees who have been promoted more than once.

a)
```
select loc.location_id, employee_id, city, department_name
from locations loc
left join departments dep
on dep.location_id=loc.location_id
left join job_history jh
on dep.department_id=jh.department_id
group by loc.location_id, city, department_name, employee_id
having count(employee_id)>1
```

b)
```
select emp.first_name, dep.department_name, loc.city, loc.location_id from
employees emp
left join departments dep
on emp.department_id=dep.department_id
left join locations loc
on loc.location_id= dep.location_id
where emp.employee_id in
(select employee_id from job_history having count(*)&gt;1 group by employee_id);
```

3. Display minimum and maximum “hire_date” of employees work in IT and HR departments.

```
select department_name, min(hire_date), max(hire_date)
from employees emp
left join departments dep
on dep.department_id =emp.department_id
where department_name = 'IT' or department_name = 'Human Resources'
group by department_name;
```

4. Find difference between current date and hire dates of employees after sorting them by hire date, then show difference in days, months and years.

```
select  trunc(sysdate-hire_date) dif_day,
trunc(months_between(sysdate,hire_date)) dif_mon,
trunc(extract(year from sysdate) - extract(year from hire_date))dif_year
from employees order by hire_date;
```

5. Find which departments used to hire earliest/latest.

```
select department_name, hire_date from employees emp
left join departments dep
on dep.department_id=emp.department_id
where hire_date = (select
max(hire_date)from employees ) or
hire_date = (select min(hire_date)from employees );
```

6. Find the number of departments with no employee for each city.

```
select count(*), loc.city from locations loc
left join departments dep
on loc.location_id= dep.location_id
where dep.department_id not in
(select emp.department_id from employees emp where emp.department_id is not null)
group by loc.city;
```

7. Create a category called “seasons” and find in which season most employees were hired.

```
select count(*) cnt , emp.seasons from employees e
left join
(select first_name,
case
when extract(month from hire_date) in (12,1,2) then 'Winter'
when extract(month from hire_date) in (3,4,5) then 'Spring'
when extract(month from hire_date) in (6,7,8) then 'Summer'
else 'Autumn'
end seasons
from employees) emp
on e.first_name=emp.first_name
group by seasons
```

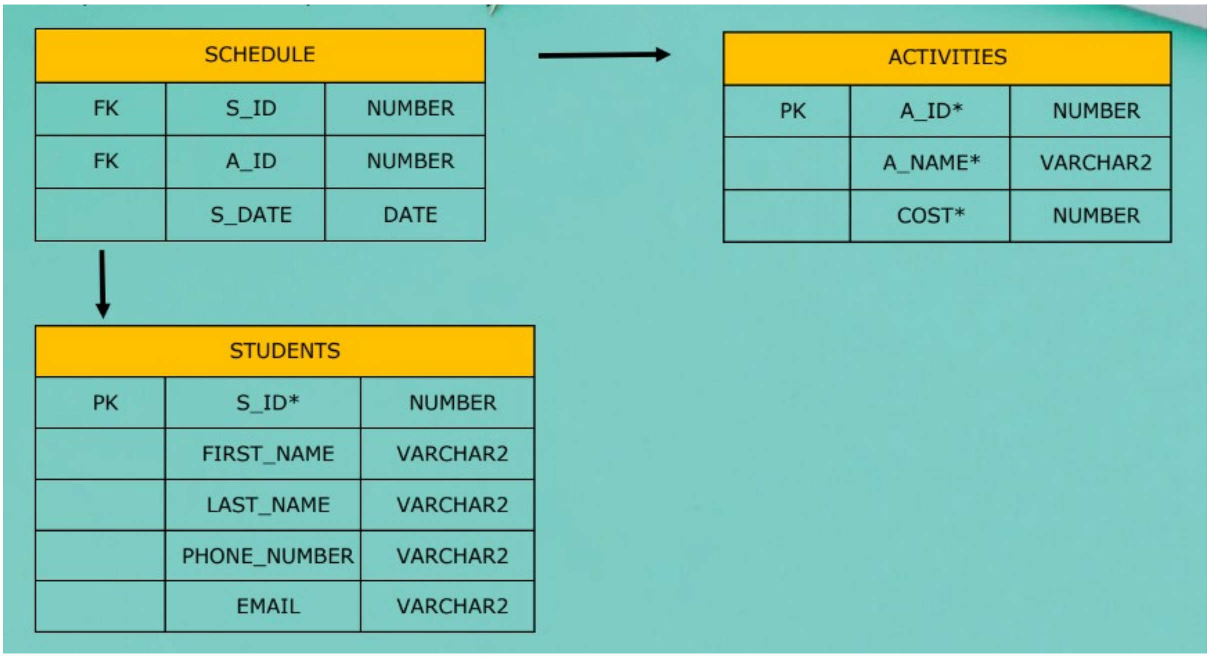
```
order by cnt desc;
```

8. Find the cities of employees with average salary more than 5000.

```
select round(avg(salary),2), city
from employees emp
left join departments dep
on emp.department_id=dep.department_id
left join locations loc
on dep.location_id=loc.location_id
group by city
having avg(salary)>5000;
```

Task 7

1. According to the given diagram create STUDENTS , ACTIVITIES and SCHEDULE tables. (PK – PRIMARY KEY, FK – FOREIGN KEY, * - NOT NULL)



SCHEDULE		
FK	S_ID	NUMBER
FK	A_ID	NUMBER
	S_DATE	DATE

ACTIVITIES		
PK	A_ID*	NUMBER
	A_NAME*	VARCHAR2
	COST*	NUMBER

STUDENTS		
PK	S_ID*	NUMBER
	FIRST_NAME	VARCHAR2
	LAST_NAME	VARCHAR2
	PHONE_NUMBER	VARCHAR2
	EMAIL	VARCHAR2

```
create table STUDENTS
( s_id number NOT NULL,
CONSTRAINT s_id_pk PRIMARY KEY (s_id),
first_name varchar2(20),
last_name varchar2(20),
phone_number varchar2(20),
email varchar2(30)
);

create table ACTIVITIES
( a_id number NOT NULL,
CONSTRAINT a_id_pk PRIMARY KEY (a_id),
a_name varchar2(20) NOT NULL,
c_ost number NOT NULL
);

create table SCHEDULE
( s_id number,
CONSTRAINT st_s_fk FOREIGN KEY (s_id) REFERENCES students (s_id),
a_id number,
CONSTRAINT ac_a_fk FOREIGN KEY (a_id) REFERENCES activities (a_id),
s_date date);
```

2. Insert data into students table from employees table.

```
insert into STUDENTS (s_id, first_name, last_name, phone_number, email)
select employee_id, first_name, last_name, phone_number, email from EMPLOYEES;
```

3. Change phone number to ‘***’ for students with s_id > 200.

```
UPDATE STUDENTS
set phone_number ='***'
where s_id>200;
```

4. Update first name and last names of students in Upper cases.

```
UPDATE STUDENTS
set first_name =upper(first_name),
last_name = upper(last_name);
```

5. Based on the students table populated with the following data, update the email to 'DSA' for all records whose s_id is greater than 150.

```
UPDATE STUDENTS
set email ='DSA'
where s_id>150;
```

6. Create PROGRAMMERS table using records from EMPLOYEES where job_id contains ‘PROG’ substring

```
create table  PROGRAMMERS
as (SELECT * from employees
where job_id like '%PROG%');
```

7. Delete records from students table where s_id is between 150 and 160.

```
delete students
where s_id between 150 and 160;
```

8. a) Insert some date into SCHEDULE, then truncate and see results.

```
insert into schedule (s_id)
select s_id from students;

truncate table schedule;
```

b) Drop schedule table

```
drop table schedule;
```

9. For any date given, write a script to find:
a) The first and the last days of the next year;

```
select add_months (trunc (sysdate, 'year'), 12) "first day of the year", add_months (trunc
(sysdate, 'year'), 24) - 1 "last day of the year" from dual;
```

b) The first and the last days of the next month;

```
select add_months (trunc (sysdate, 'month'), 1) "first day of the month" ,  add_months (trunc
(sysdate, 'month'), 2) - 1 "last day of the month" from dual;
```

c) The first and the last days of the previous month.

```
select add_months (trunc (sysdate, 'month'), -1) "first day of the previous month" ,
add_months (trunc (sysdate, 'month'), 0) - 1 "last day of the month" from dual;
```

10. Create a table named “Participants” which consists of first_name, last_name and salary (have to more than 10000).

```
create table Participants
(
first_name VARCHAR2(20),
last_name VARCHAR2(20),
salary number check (salary > 10000)
);
```

Task 8

1. Return the name of the employee with the lowest salary in department 90.

```
Select first_name, salary from employees
where salary =( select min(salary) from employees where department_id =90);
```

2. Select the department name, employee name, and salary of all employees who work in the human resources or purchasing departments. Compute a rank for each unique salary in both departments.

```
select department_name, first_name, salary, dense_rank()
over (partition by department_name order by salary) from employees emp
left join departments dep
on emp.department_id=dep.department_id
where department_name in ('purchasing','human resources');
```

3. Select the 3 employees with minimum salary for department id 50.

```
select * from (select first_name, salary , DENSE_RANK()
over (order by salary) ranking
from employees where department_id =50)
where ranking in (1,2,3);
```

4. Show first name, last name, salary and previously listed employee’s salary who works in “IT_PROG” over hire date.

```
select first_name, last_name, salary, lag(salary, 1, 0)
over (order by hire_date) prev_sal
```

```
from employees
where job_id = 'it_prog';
```

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

```
select first_name, j.* from employees emp
left join jobs j on emp.job_id=j.job_id where employee_id in
(select employee_id from job_history where job_id='IT_PROG');
```

6. Make a copy of the employees table and update the salaries of the employees in the new table with the maximum salary in their departments.

```
create table new_table as (select * from employees);
```

```
update new_table nt set salary = (select max(salary) from employees e where
e.department_id=nt.department_id);
```

7. Make a copy of the employees table and update the salaries of the employees in the new table with a 30 percent increase.

```
update new_table set salary = salary*1.3;
```