use sql_project;

Problem Statements to be solved using - Where Clause

1. Display the first name and department number for all customers whose last name is "De Haan" (Employees table).

```
select first_name , department_id  from employees
where last_name = 'De Haan' ;
```

2. Display all data from Departments table for Sales department (department_name column)

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

3. Display the first name, last_name, department number and salary for all employees who earn more than 9700 (Employees table).

```
select first_name , last_name , department_id , salary
from employees
where salary > 9700;
```

4. Display all data from Employees table for all employees who was hired before January 1st, 1992.

```
 select * from employees
 where hire_date < '1992-01-01';
```

5. Display the employee number, first name, job id and department number for all employees whose department number equals 20, 60 or 80 (Employees table)

select employee_id , first_name , job_id , department_id

from employees

where department_id IN(20,60,80);

6. Display the employee number, first name, job id and department number for all employees whose department number is not equal to 20, 60 and 80 (Employees table).

select employee_id , first_name , job_id , department_id

from employees

where department_id NOT IN(20,60,80);

7. Display the last name, phone number, salary and manager number, for all employees whose manager number equals 100, 102 or 103 (Employees table).

select last_name , phone_number , salary , manager_id

from employees

where manager_id IN(100,102,103);

8. Display the first name and salary for all employees whose first name ends with an e (Employees table).

select first_name , salary

from employees

where first_name like '%e';

9. Display the last name and department number for all employees where the second letter in their last name is i (Employees table).

```
select last_name , department_id
from employees
where last_name like '_i%';
```

10. Display all data from Employees table for all employees who have the letters : L, J, or H in their last name. Sort the query in descending order by salary.

```
select * from employees
where last_name like '%L%' or last_name like '%J%' or last_name like '%H%'
order by salary desc;
```

11. Display the first name, hire date, salary and department number for all employees whose first name doesn't have the letter A. Sort the query in ascending order by department number (Employees table).

```
select first_name , hire_date , salary , department_id
from employees
where first_name not like '%A%';
```

12. Display all data from Employees table for all employees without any department number

```
select * from employees
where department_id is null;
```

13. Display the first name concatenated with the last name, separated by comma, and salary, for all employees whose salary not in the range between 7000 and 15000. Sort the query in ascending order by the full name (Employees table).

```
select first_name, last_name , concat(first_name,' ',last_name) as full_name , salary

from employees

where salary not between 7000 and 15000

order by full_name;
```

14. Display the first name concatenated with the last name, separated by comma, the phone number concatenated with the email address, separated by hyphen, and salary, for all employees whose salary is in the range of 5000 and 10000. Name the column headings: "FULL_NAME", "CONTACTS" and "SAL" respectively (Employees table).

```
select first_name, last_name , concat(first_name,' ',last_name) as FULL_NAME, phone_number , email

concat(phone_number,' - ', email) as CONTACTS , salary as SAL

from employees

where  salary between 5000 and 10000;
```

15. Display all data from Employees table for all employees whose: salary is in the range of 6000 and 800 and their commission is not null or department number is not equal to 80, 90 and 100 and their hire date is before January 1st, 1990.

```
select * from employees

where salary between 800 and 6000

and commission_pct is not null

or department_id NOT IN(80,90,100)

and hire_date < '1990-01-01';
```

16. Display last name, job id and hire date for all employees who was hired during December 12th,

 1995 and April 17th, 1998.

select last_name , job_id , hire_date

from employees

where hire_date between '1995-12-12' and '1998-04-17';

17. Display the first name concatenated with last name, hire date, commission percentage,

   telephone, and salary for all employees whose salary is greater than 10000 or the third digit in their

    phone number equals 5. Sort the query in a descending order by the first name (Employees table).

    select first_name , last_name, concat(first_name,' ',last_name) as full_name , hire_date,

    commission_pct , phone_number , salary

    from employees

    where salary > 10000 or phone_number like '__5%'

    order by first_name desc;

  18. Display the last name and salary for all employees who earn more than 12000 (Employees

     table).

     select last_name , salary

     from employees

     where salary > 12000;

   19. Display the last name and department number for all employees whose department number is

      equal to 50 or 80. Perform this exercise once by using the IN operator, once by using the OR

      operator.

```
select last_name , department_id

from employees

where department_id IN (50,80);


select last_name , department_id

from employees

where department_id = 50 OR department_id = 80;
```

20. Display the first name and salary for all employees who doesn't earn any commission.

```
select first_name , salary

from employees

where commission_pct is null;
```

21. Display the first name, salary, and manager number for all employees whose manager
    number is not null

```
select first_name , salary , manager_id

from employees

where manager_id is not null;
```

Problem Statements to be solved using – SQL Scalar/Numeric Functions

1. Display the first name in lower case and last name in upper case, for all employees whose
   employee number is in the range between 80 and 150.

```
select lower(first_name) , upper(last_name) , employee_id
from employees
where employee_id between 80 and 150;
```

2. Display the first name and last name for all employees whose family name is King, perform
   this exercise with a case-insensitive search (regardless of the capitalization used for the
   values within last name column).

```
select first_name , last_name
from employees
where last_name = 'king';
```

3. Generating new email address

a. For each employee, display the first name, last name, and email address. The email
   address will be composed from the first letter of first name, concatenated with the three
   first letters of last name, concatenated with @oracle.com

```
select first_name, last_name, concat(Left(first_Name,1), Left(Last_Name,3), '@oracle.com')
as Email_Address
 from employees;
```

b. For each employee, display the first name, last name, and email address. The email
   address will be composed from the first letter of first name, concatenated with the three
   last letters of last name, concatenated with @oracle.com.

select first_name , last_name , concat(Left(first_name,1), Right(last_name,3)

,'@oracle.com.') as Email_Address

from employees;

4. Using the CONCAT function

a. For each employee, use the CONCAT function to display the first name concatenated
   With the last name.

   select first_name , last_name , concat(first_name, ' ',last_name) as full_name

   from employees;

b. For each employee, use the CONCAT function to display the first name concatenated
   the last name, concatenated with hire date.

   select first_name , last_name , hire_date , concat(first_name,' ',last_name,' –

   ',hire_date)

   from employees;

   5. Display the last name for all employees where last name's length is greater than 8
      characters.

   select last_name from employees

   where length(last_name) > 8;

6. Phone numbers:

a. For each employee, display the first name, last name, phone number and a new phone number using the REPLACE function. in the new phone number replace all occurrences of 515 with 815.

select first_name , last_name , phone_number , replace (phone_number,515,815)

 as New_Phone_Number

from employees;

b. For each employee, display the first name, last name, phone number and a new phone number using the REPLACE function. in the new phone number replace all prefixes of 515 with 815.

select first_name , last_name , phone_number , replace (phone_number,515,815)

as New_Phone_Number

from employees;

7. For each employee, display :

a. first name

 select first_name from employees;

 b. salary

select salary from employees;

c. salary after a raise of 12%

select salary ,salary*(12/100) as Salary_raise_by_12_perc from employees;

d. salary after a raise of 12%, expressed as a whole number (ROUND)

select salary ,round(salary*(12/100)) as Salary_raise_by_12_perc from

 employees;

e. salary after a raise of 12%, round down to the nearest whole number.

select salary , floor(salary*(12/100)) as rounded_down_Salary_raise_by_12_per

 from employees;

8. For each employee, display the first name, hire date, hire date minus 10 days,

hire date per month, and the day difference between current date and

hiredate.-

select first_name , hire_date,

adddate(hire_date,-10) as hire_date_minus_by_10_days,

timestampadd(Month,1,hire_date)as hire_date_plus_1_month ,

datediff(curdate(),hire_date) as

day_difference_between_curdate_and_hire_date

from employees;

9. For each employee, display the first name, last name, hire date, number of

months he works the company, and number of years he works in the company.

select first_name , last_name , hire_date ,

timestampdiff(Month,hire_date,curdate()) as

number_of_months_works_in_the_company ,

```
timestampdiff(year,hire_date,curdate())
  as  number_of_year_works_in_the_company
  from employees;
```

10. For each employee, display the first name, hire date, and hire date plus one
    year

```
select first_name , hire_date ,
timestampadd(Year,1,hire_date) as hire_date_plus_one_year
from employees;
```

11. For each employee, display the first name, hire date, hire date rounded up to
    the nearest year, and hire date rounded up to the nearest month.

```
select first_name , hire_date ,
Year(round(hire_date)) as 'Nearest_year' , Month(round(hire_date)) as
'Nearest_month'
from employees;
```

12. For each employee, display the first name, the day of his hire date, and the
    year of his hire date

```
select first_name, hire_date,
date_format(hire_date, '%D') as Day_of_his_hire_date,
date_format(hire_date, '%Y') as Year_of_his_Hire_date
from employees;
```

13. Display the last name in upper case, the salary in format model : '9,999.999', and

Hire date in format model: 'DD/MM/YYYY', for all employees whose last name

begins with the letter D or K.

alter table employees

modify salary decimal(8,3);

select upper(last_name) , salary,

date_format(hire_date, '%d/%m/%Y')

from employees

where last_name like 'D%'

or last_name like 'K%';

14. Commission Percentage

a. For each employee, display the first name, last name, salary and commission percentage. If an employee doesn't earn a commission, display 0 instead of NULL.
b. For each employee, display the first name, last name, salary and commission percentage. If an employee doesn't earn a commission, display "No Commission" instead of NULL.

a) For each employee, display the first name, last name, salary and commission percentage. If an employee doesn't earn a commission, display 0 instead of NULL.

select first_name , last_name , salary , commission_pct ,

coalesce(commission_pct , 0)

from employees;

b) For each employee, display the first name, last name, salary and commission percentage. If an employee doesn't earn a commission, display "No Commission" instead of NULL.

select first_name , last_name , salary , commission_pct ,

coalesce(commission_pct , 'No Commission') as Massage

from employees;

15. For each employee, display the first name, last name, salary, and a salary grade based on these conditions :

a. if the salary is between 0 and 5000 – salary grade level is A

b. if the salary is between 5001 and 15000 – salary grade level is B

c. if the salary is between 15001 and 20000 – salary grade level is C

d. for any other range – salary grade level is D

select first_name, last_name, salary,

if (salary between 0 and 5000,'A',

if(salary between 5001 and 15000,'B',

if(salary between 15001 and 20000,'C','D'))) as Salary_grade_level

from employees;

Problem Statements to be solved using – Basic Select Statements

1.Create a query to display the employee number, first name, last name phone number and   department number (Employees table).

```
select employee_id , first_name , last_name , phone_number ,
 department_id
from employees;
```

2. Create a query to display the first name, last name, hire date, salary, and salary after a raise of 20%. Name the last column (salary after a raise) heading as "ANNUAL_SAL" (Employees table). */

```
 select first_name , last_name , hire_date , salary ,
 round( salary*(20/100)) as 'salary_after_rise' ,
 round(salary*(20/100)) + salary as 'ANNUAL SALARY'
 from employees;
```

3. Create a query to display the last name concatenated with the first name, separated by space, and the telephone number concatenated with the email address, separated by hyphen. Name the column headings "FULL_NAME" and "CONTACT_DETAILS" respectively (Employees tables).

```
select last_name , first_name , concat(last_name, '  ' , first_name) as
'Full_Name' ,
phone_number , email , concat(phone_number, ' - ' , email) as
'CONTACT_DETAILS'
from employees;
```

4. Create a query to display the unique manager numbers from Employees table.

```
select distinct(manager_id) from employees;
```

5. Create a query to display the last name concatenated with job_id column, separated by space .Name this column heading as "EMPLOYEE_AND_TITLE" (Employees table).

```
select last_name, job_id , concat(last_name, '  ', job_id) as
'EMPLOYEE_AND_TITLE '
from employees;
```

6. Create a query to display the first name, last name, salary, and hire date concatenated with the literal string "HD", separated by space. Name the column headings "FN", "LN", "SAL", and "HD" respectively (Employees table).

```
select first_name as FN , last_name as LN , salary as SAL ,
concat(hire_date,'   ','HD') as HD
from employees;
```

7. Create a query to display the unique salaries in Employees tables.
```
select distinct(salary) from employees;
```

8. Create a query to display the unique combination of values in department_id and job_id   columns (Employees table)

select distinct department_id, job_id from employees;

Problem Statements to be solved using – Group By

1. Display the lowest last name alphabetically (Employees table).

select last_name
from employees
order by last_name
limit 1 ;

2. Display the highest last name alphabetically (Employees table).

select last_name
from employees
order by last_name desc
limit 1;

3. Display the number of rows in Employees table.

select count(*) from employees;

4. Display the number of values (exclude NULLs) in commission_pct column (Employees table).

```
select count(commission_pct) as Non_null_value
from employees
where commission_pct is not null;
```

5. Display the number of NULL values in commission_pct column (Employees table).

```
select count(*) as null_value
from employees
where commission_pct is null  ;
```

6. Display the highest, lowest, and average salary.Problem Statements to be solved using - GROUP BY and HAVING Clauses

```
select salary,
max(salary) as 'Highest  salary' ,
min(salary) as 'Lowest  salary',
round(avg(salary)) as 'Avg salary'
from employees
group by salary;
```

7. Average salary per department a. Display the department number and average salary for each department. b. Modify your query to display the results only for departments 50 or 80.

a) Display the department number and average salary for each department.

```
select department_id , avg(salary)
from employees
group by department_id;
```

b)Modify your query to display the results only for departments 50 or   80.

```
select department_id , avg(salary)
from employees
group by department_id
having department_id = 50 or department_id = 80;
```

8. Number of employees per job id

a. Display the job id and the number of employees for each job id.

b. Modify your query to display the results only for employees whose salary is greater than 10000.

c.   Modify your query again, this time display the results only for jobs With more than 2 people.

a. Display the job id and the number of employees for each job id.

```
select job_id , count(*) as 'employee count'
from employees
group by job_id ;
```

b.Modify your query to display the results only for employees whose salary is greater than 10000.

```
select job_id , count(*) as 'employee count'
from employees
where salary > 10000
group by job_id;
```

c.Modify your query again, this time display the results only for jobs with more than 2   people.

```
select job_id , count(*) as 'employee count'
from employees
group by job_id
having count(*) > 2;
```

9. Display the department number, job id, and the average salary for each department and job id.

```
select department_id , job_id , avg(salary)
from employees
group by department_id , job_id ;
```

10. Managers and highest salary a. Display the manager number and the highest salary for each manager number. b. Modify your query to display the results only for employees whose salary is   greater than 10000.

a. Display the manager number and the highest salary for each manager number

select manager_id , max(salary) as 'Highest salary'

from employees

group by manager_id;

c. Modify your query to display the results only for employees whose salary is greater than 10000.

select manager_id , max(salary) as 'Highest salary'

from employees

group by manager_id , salary

having salary > 10000;

11. Display the job id and minimum salary for each job id, for all jobs whose minimum salary greater than 7000

select job_id , min(salary) as 'Min salary'

from employees

group by job_id , salary

having min(salary) > 7000;

12. Display the department number, and the average salary for each department, for all departments whose number is in the range of 20 and 80, and their average salary is greater than 9000

select department_id , round(avg(salary)) as 'AVG SALARY'

from employees

group by department_id , salary

having department_id between 20 and 80 and avg(salary) > 9000;

Problem Statements to be solved using – SQL Sub-Queries

1. Display the first name and salary for all employees who earn more than employee number 103 (Employees table).

   select first_name , salary from employees

   where salary >(select salary from employees where employee_id = 103);

2. Display the department number and department name for all departments whose location number is equal to the location number of department number 90 (Departments table).

   select department_id , department_name

   from departments

   where location_id = (select location_id from departments where department_id = 90);

3. Display the last name and hire date for all employees who was hired after employee number 101 (Employees table).

   select last_name , hire_date

   from employees

   where hire_date > (select hire_date from employees where employee_id =101);

4. Display the first name, last name, and department number for all employees who work in Sales department (Employees and Departments table).

   select first_name , last_name , department_id from employees

   where department_id =(select department_id from departments

where department_name = 'sales');

5. Display the department number and department name for all departments located in Toronto (Departments table).

select department_id , department_name
from departments
where location_id = (select location_id from locations where city = 'Toronto');

6. Display the first name, salary and department number for all employees who work in the department as employee number 124 (Employees table).

select first_name , salary , department_id
from employees
where department_id = (select department_id from employees where employee_id = 124);

7. Display the first name, salary, and department number for all employees who earn more than the average salary (Employees table).

select first_name , salary , department_id
from employees
where salary > (select avg(salary) from employees);

8. Display the first name, salary, and department number for all employees whose salary equals one of the salaries in department number 20 (Employees table).

```
select first_name , salary , department_id

from employees

where salary IN (select salary from employees where department_id =20);
```

9. Display the first name, salary, and department number for all employees who earn more than maximum salary in department number 50 (Employees table).

```
select first_name , salary , department_id

from employees

where salary > (select max(salary) from employees where department_id =50);
```

10. Display the first name, salary, and department number for all employees who earn more than the minimum salary in department number 60 (Employees table).

```
select first_name , salary , department_id

from employees

where salary > (select min(salary) from employees where department_id = 60);
```

11. Display the first name, salary, and department number for all employees who earn less than the minimum salary of department number 90 (Employees table)

```
select first_name , salary , department_id

from employees

where salary < (select min(salary) from employees where department_id = 90);
```

12. Display the first name, salary and department number for all employees whose department is located Seattle (Employees, Departments and Locations table).

```
select first_name , salary , department_id

from employees

where department_id IN(select department_id from departments where location_id = (select

location_id from locations where city = 'seattle'));
```

13. Display the first name, salary, and department number for all employees who earn less than the
    average salary, and also work at the same department as employee whose first name is Kevin

```
select first_name , salary , department_id

from employees

where salary < (select avg(salary) from employees where department_id =(select department_id

where first_name = 'kevin'));
```

Problem Statements to be solved using – Inner JOIN

1. Employees and departments (Employees & Departments tables)

a. For each employee, display the first name, last name, department number and department name.

```
select e.first_name , e.last_name , e.department_id  , d.department_name

 from employees e

join departments d

 on e.department_id = d.department_id ;
```

b. Display the first name, last name, department number and department name, for all employees in
   departments 50 or 90.

```
select e.first_name , e.last_name , e.department_id , d.department_name
```

```
from employees e

join  departments d

on e.department_id = d.department_id

where e.department_id = 50 or e.department_id = 90;
```

2. Departments and locations (Departments, Employees & Locations tables)

a. For each department, display the department name, city, and state province.

```
select d.department_name, l.city, l.state_province
from departments d
join locations l
on d.location_id = l.location_id;
```

b. For each employee, display the full name, department name, city, and state province

```
select concat(e.first_name,'   ',e.last_name) as 'full_name', d.department_name , l.city, l.state_province
from employees e
join departments d
on e.department_id = d.department_id
join locations l
on d.location_id = l.location_id ;
```

c. Display the full name, department name, city, and state province, for all employees whose last
   name contains the letter a.

```
select concat(e.first_name,'   ',e.last_name) as 'full_name', d.department_name , l.city, l.state_province
from employees e
join departments d
```

on e.department_id = d.department_id

join locations l

on d.location_id = l.location_id

where last_name like '%a%' ;

Problem Statements to be solved using – Outer JOIN

4. Employees & departments

a. Display the first name, last name, department number and department name, for all employees
   including those without any department.

```
select e.first_name , e.last_name , e.department_id , d.department_name
from employees e
left join  departments d
on e.department_id = d.department_id ;
```

b. Modify your query to display all departments including departments without any employees.

```
select e.first_name , e.last_name , e.department_id , d.department_name
from employees e
right join  departments d
on e.department_id = d.department_id ;
```

Problem Statements to be solved using – Self JOIN

5. Employees and managers (Employees table)

a. For each employee, display the last name, and the manager's last name.

```
select e1.last_name as employee_last_name , e2.last_name as manager_last_name
from employees e1
 join employees e2
 on e1.manager_id = e2.employee_id ;
```

b. Modify your query to display all employees including those without any manager.

```
select e1.last_name as "Employee Name",
(select e2.last_name from employees e2
 where e1.manager_id = e2.employee_id) AS "Manager Name"
 from employees e1;
```

6. Display the first name, last name, and department number for all employees who work in the same department as employee whose last name is "King".

```
select department_id
from employees
where last_name ='king';
```

```
select e.first_name, e.last_name, e.department_id
from employees e
join (select department_id  from employees   where last_name ='king') d
on e. department_id = d.department_id;
```

7. Display the last name and salary for all employees who earn less than employee number 103

```
select e1.last_name,e1.salary
from employees e1
join employees e2
ON e1.salary < e2.salary
where e2.employee_id = 103;
```