20 SQL EXERCISES FOR PRACTICE

Table structure and schema:

Create a Table with a primary key column

```
CREATE TABLE departments

    ( department_id INTEGER PRIMARY KEY

    , department_name VARCHAR(30)

    , location_id INTEGER

    ) ;
```

Create a Table with a Foreign Key

```
CREATE TABLE employees

  ( employee_id INTEGER

  , first_name VARCHAR(20)

  , last_name VARCHAR(25)

  , email VARCHAR(25)

  , phone_number VARCHAR(20)

  , hire_date DATE

  , job_id VARCHAR(10)

  , salary INTEGER

  , commission_pct INTEGER

  , manager_id INTEGER
```

```
    , department_id INTEGER

    , constraint pk_emp primary key (employee_id)

    , constraint fk_deptno foreign key (department_id) references
departments(department_id)

    ) ;
```

Insert Records into Tables

```
## Insert insto Departments table

 INSERT INTO departments VALUES ( 20,'Marketing',  180);

 INSERT INTO departments VALUES ( 30,'Purchasing',  1700);

 INSERT INTO departments VALUES ( 40, 'Human Resources',  2400);

 INSERT INTO departments VALUES ( 50, 'Shipping',  1500);

 INSERT INTO departments VALUES ( 60 , 'IT',  1400);

 INSERT INTO departments VALUES ( 70, 'Public Relations',  2700);

 INSERT INTO departments VALUES ( 80 , 'Sales',  2500 );

 INSERT INTO departments VALUES ( 90 , 'Executive',  1700);

 INSERT INTO departments VALUES ( 100 , 'Finance',  1700);

 INSERT INTO departments VALUES ( 110 , 'Accounting',  1700);

 INSERT INTO departments VALUES ( 120 , 'Treasury' ,  1700);

 INSERT INTO departments VALUES ( 130 , 'Corporate Tax' ,  1700 );

 INSERT INTO departments VALUES ( 140, 'Control And Credit' ,  1700);

 INSERT INTO departments VALUES ( 150 , 'Shareholder Services', 1700);

 INSERT INTO departments VALUES ( 160 , 'Benefits', 1700);
```

```sql
  INSERT INTO departments VALUES ( 170 , 'Payroll' , 1700);
```

Insert into Employees table

```sql
INSERT INTO employees VALUES (100, 'Steven', 'King', 'SKING', '515.123.4567',
'1987-06-17' , 'AD_PRES', 24000 , NULL, NULL, 20);


INSERT INTO employees VALUES (101, 'Neena' , 'Kochhar' , 'NKOCHHAR' ,
'515.123.4568' , '1989-11-21' , 'AD_VP' , 17000 , NULL , 100 , 20);


INSERT INTO employees VALUES (102 , 'Lex' , 'De Haan' , 'LDEHAAN' ,
'515.123.4569' , '1993-09-12' , 'AD_VP' , 17000 , NULL , 100 , 30);


INSERT INTO employees VALUES (103 , 'Alexander' , 'Hunold' , 'AHUNOLD' ,
'590.423.4567' , '1990-09-30', 'IT_PROG' , 9000 , NULL , 102 , 60);


INSERT INTO employees VALUES (104 , 'Bruce' , 'Ernst' , 'BERNST' ,
'590.423.4568' , '1991-05-21',  'IT_PROG' , 6000 , NULL , 103 , 60);


INSERT INTO employees VALUES (105 , 'David' , 'Austin' , 'DAUSTIN' ,
'590.423.4569' , '1997-06-25',  'IT_PROG' , 4800 , NULL , 103 , 60);


INSERT INTO employees VALUES (106 , 'Valli' , 'Pataballa' , 'VPATABAL' ,
'590.423.4560' , '1998-02-05',  'IT_PROG' , 4800 , NULL , 103 , 40);


INSERT INTO employees VALUES (107 , 'Diana' , 'Lorentz' , 'DLORENTZ' ,
'590.423.5567' , '1999-02-09',  'IT_PROG' , 4200 , NULL , 103 , 40);


INSERT INTO employees VALUES (108 , 'Nancy' , 'Greenberg' , 'NGREENBE' ,
'515.124.4569' , '1994-08-17',  'FI_MGR' , 12000 , NULL , 101 , 100);


INSERT INTO employees VALUES (109 , 'Daniel' , 'Faviet' , 'DFAVIET' ,
'515.124.4169' , '1994-08-12',  'FI_ACCOUNT' , 9000 , NULL , 108 , 170);


INSERT INTO employees VALUES (110 , 'John' , 'Chen' , 'JCHEN' ,
'515.124.4269' , '1997-04-09',  'FI_ACCOUNT' , 8200 , NULL , 108 , 170);


INSERT INTO employees VALUES (111 , 'Ismael' , 'Sciarra' , 'ISCIARRA' ,
'515.124.4369' , '1997-02-01',  'FI_ACCOUNT' , 7700 , NULL , 108 , 160);


INSERT INTO employees VALUES (112 , 'Jose Manuel' , 'Urman' , 'JMURMAN' ,
'515.124.4469' , '1998-06-03', 'FI_ACCOUNT' , 7800 , NULL , 108 , 150);
```

```sql
INSERT INTO employees VALUES (113 , 'Luis' , 'Popp' , 'LPOPP' ,
'515.124.4567' , '1999-12-07',  'FI_ACCOUNT' , 6900 , NULL , 108 , 140);


INSERT INTO employees VALUES (114 , 'Den' , 'Raphaely' , 'DRAPHEAL' ,
'515.127.4561' , '1994-11-08',  'PU_MAN' , 11000 , NULL , 100 , 30);


INSERT INTO employees VALUES (115 , 'Alexander' , 'Khoo' , 'AKHOO' ,
'515.127.4562' , '1995-05-12', 'PU_CLERK' , 3100 , NULL , 114 , 80);


INSERT INTO employees VALUES (116 , 'Shelli' , 'Baida' , 'SBAIDA' ,
'515.127.4563' ,'1997-12-13', 'PU_CLERK' , 2900 , NULL , 114 , 70);


INSERT INTO employees VALUES (117 , 'Sigal' , 'Tobias' , 'STOBIAS' ,
'515.127.4564' , '1997-09-10', 'PU_CLERK' , 2800 , NULL , 114 , 30);


INSERT INTO employees VALUES (118 , 'Guy' , 'Himuro' , 'GHIMURO' ,
'515.127.4565' , '1998-01-02',  'PU_CLERK' , 2600 , NULL , 114 , 60);


INSERT INTO employees VALUES (119 , 'Karen' , 'Colmenares' , 'KCOLMENA' ,
'515.127.4566' , '1999-04-08',  'PU_CLERK' , 2500 , NULL , 114 , 130);


INSERT INTO employees VALUES (120 , 'Matthew' , 'Weiss' , 'MWEISS' ,
'650.123.1234' ,'1996-07-18',  'ST_MAN' , 8000 , NULL , 100 , 50);


INSERT INTO employees VALUES (121 , 'Adam' , 'Fripp' , 'AFRIPP' ,
'650.123.2234' , '1997-08-09',  'ST_MAN' , 8200 , NULL , 100 , 50);


INSERT INTO employees VALUES (122 , 'Payam' , 'Kaufling' , 'PKAUFLIN' ,
'650.123.3234' ,'1995-05-01',  'ST_MAN' , 7900 , NULL , 100 , 40);


INSERT INTO employees VALUES (123 , 'Shanta' , 'Vollman' , 'SVOLLMAN' ,
'650.123.4234' , '1997-10-12',  'ST_MAN' , 6500 , NULL , 100 , 50);


INSERT INTO employees VALUES (124, 'Kevin' , 'Mourgos' , 'KMOURGOS' ,
'650.123.5234' , '1999-11-12',  'ST_MAN' , 5800 , NULL , 100 , 80);


INSERT INTO employees VALUES (125, 'Julia' , 'Nayer' , 'JNAYER' ,
'650.124.1214' , '1997-07-02',  'ST_CLERK' , 3200 , NULL , 120 , 50);


INSERT INTO employees VALUES (126, 'Irene' , 'Mikkilineni' , 'IMIKKILI' ,
'650.124.1224' , '1998-11-12', 'ST_CLERK' , 2700 , NULL , 120 , 50);


INSERT INTO employees VALUES (127, 'James' , 'Landry' , 'JLANDRY' ,
'650.124.1334' , '1999-01-02' , 'ST_CLERK' , 2400 , NULL , 120 , 90);
```

```
INSERT INTO employees VALUES (128, 'Steven' , 'Markle' , 'SMARKLE' ,
'650.124.1434' , '2000-03-04' , 'ST_CLERK' , 2200 , NULL , 120 , 50);


INSERT INTO employees VALUES (129, 'Laura' , 'Bissot' , 'LBISSOT' ,
'650.124.5234' ,'1997-09-10' , 'ST_CLERK' , 3300 , NULL , 121 , 50);


INSERT INTO employees VALUES (130, 'Mozhe' , 'Atkinson' , 'MATKINSO' ,
'650.124.6234' , '1997-10-12' , 'ST_CLERK' , 2800 , NULL , 121 , 110);
```

So, now we have 2 tables and some data ready to run our sql. It's time for some exercises.

Solve SQL Exercises

1. Select employees first name, last name, job_id and salary whose first name starts with alphabet S

```
select first_name,

       last_name,

       job_id,

       salary

 from employees

 where upper(first_name) like 'S%';
```

2. Write a query to select employee with the highest salary

```
select employee_id,

       first_name,

       last_name,

       job_id,

       salary

 from employees
```

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

3. Select employee with the second highest salary

```
select employee_id,

        first_name,

        last_name,

        job_id,

        salary

 from employees

 where salary != (select max(salary) from employees)

 order by salary desc

 limit 1;
```

The above query selects only one person with the second-highest salary. But what if there are more than 1 person with the same salary? Or, what if we want to select the 3rd or 4th highest salary? So, let's try a generic approach.

4. Fetch employees with 2nd or 3rd highest salary

```
#change the input for 2nd, 3rd or 4th highest salary

set @input:=3;

select employee_id,

        first_name,

        last_name,

        job_id,

        salary
```

```
from employees e

where @input =(select COUNT(DISTINCT Salary)

        from employees p

        where e.Salary<=p.Salary);
```

5. Write a query to select employees and their corresponding managers and their salaries

Now, this is a classic example of **SELF JOIN** in SQL exercises. Also, I am using the **CONCAT** function to concatenate the first name and last name of each employee and manager.

```
select concat(emp.first_name,' ',emp.last_name) employee,

     emp.salary emp_sal,

     concat(mgr.first_name,' ',mgr.last_name) manager,

     mgr.salary mgr_sal

from employees emp

join employees mgr on emp.manager_id = mgr.employee_id;
```

6. Write a query to show count of employees under each manager in descending order

```
select

sup.employee_id employee_id,

    concat(sup.first_name,' ', sup.last_name)manager_name,

    COUNT (sub.employee_id) AS number_of_reportees

from employees sub

join employees sup
```

```
on sub.manager_id = sup.employee_id

group by sup.employee_id, sup.first_name, sup.last_name

order by 3 desc;
```

7. Find the count of employees in each department

```
select dept.department_name,

count(emp.employee_id) emp_count

from employees emp

join departments dept on emp.department_id = dept.department_id

group by dept.department_name

order by 2 desc;
```

8. Get the count of employees hired year wise

```
select year(hire_date) hired_year, count(*) employees_hired_count

from employees

group by year(hire_date)

order by 2 desc;
```

9. Find the salary range of employees

```
select min(salary) min_sal,

max(salary) max_sal,

round(avg(salary)) avg_sal

from employees;
```

10. Write a query to divide people into three groups based on their salaries

```
select concat(first_name,' ',last_name) employee,

salary,

case

when salary >=2000 and salary < 5000 then "low"

when salary >=5000 and salary < 10000 then "mid"

else

"high"

end as salary_level

from employees

order by 1;
```

11. Select the employees whose first_name contains "an"

```
select (first_name)

from employees

where lower(first_name) like '%an%';
```

12. Select employee first name and the corresponding phone number in the format (_ _ _)-(_ _ _)-(_ _ _ _)

```
select concat(first_name, ' ', last_name) employee,

replace(phone_number,'.','-') phone_number

from employees;
```

13. Find the employees who joined in August, 1994.

```
select concat(first_name, ' ', last_name) employee,

hire_date
```

```
from employees

where year(hire_date) = '1994'

and month(hire_date) = '08';
```

14. Write an SQL query to display employees who earn more than the average salary in that company

```
select

concat(emp.first_name,last_name) name,

emp.employee_id,

dept.department_name department,

dept.department_id,

emp.salary

from departments dept

JOIN employees emp on dept.department_id = emp.department_id

where emp.salary > (select avg(salary) from employees)

order by dept.department_id;
```

15. Find the maximum salary from each department.

```
select

dept.department_id,

dept.department_name department,

max(emp.salary)maximum_salary

from departments dept

JOIN employees emp on dept.department_id = emp.department_id
```

```
group by dept.department_name,

dept.department_id

order by dept.department_id ;
```

16. Write a SQL query to display the 5 least earning employees

```
select

first_name, last_name,

employee_id,

salary

from employees

order by salary

limit 5;
```

17. Find the employees hired in the 80s

```
select employee_id,
concat(first_name,' ' , last_name) employee,
hire_date
from employees
where year(hire_date) between 1980 and 1989;
```

18. Display the employees first name and the name in reverse order

```
select lower(first_name) name,

lower(reverse(first_name)) name_in_reverse

from employees;
```

19. Find the employees who joined the company after 15th of the month

```
select employee_id,
```

```
concat(first_name, ' ' , last_name) employee,

hire_date

from employees

where day(hire_date)> 15;
```

20. Display the managers and the reporting employees who work in different departments

```
select

 concat(mgr.first_name,' ',mgr.last_name) manager,

 concat(emp.first_name,' ',emp.last_name) employee,

 mgr.department_id mgr_dept,

 emp.department_id emp_dept

 from employees emp

 join employees mgr on emp.manager_id = mgr.employee_id

 where emp.department_id != mgr.department_id

 order by 1;
```

Source: https://oindrilasen.com/2021/04/sql-interview-practice/