INFSCI 2710 Database Management, Fall 2024
Homework 2: SQL

Q1 [10 pt] Write an SQL query to return the department name and the average salary of the employees in each department. Order the result by average salary in descending order.

Query:

```
select department_name, avg(salary) average_salary
from Departments d
join Employees e
on d.department_id = e.department_id
group by department_name
order by average_salary desc;
```

Q2 [10 pt] Write an SQL query to return the name of employees who have worked on more than one project, along with the number of projects they have participated in.

Query:

```
select e.employee_name, count(ep.project_id) as number_of_projects
from Employees e
join Employee_project ep
on e.employee_id = ep.employee_id
group by e.employee_id, e.employee_name
having count(ep.project_id) > 1;
```

Q3 [10 pt] Write an SQL query to return each employee's name and the total hours they have worked across all projects.

Query:

```
select e.employee_name, sum(ep.hours_worked) as total_hours
from Employees e
join Employee_project ep
on e.employee_id = ep.employee_id
group by e.employee_id, e.employee_name;
```

Q4 [10 pt] Write an SQL query to return the name, position, and salary of the highest-paid employee.

Query:

```
select employee_name, position, salary
from Employees
where salary = (select max(salary) from Employees);
```

Q5 [10 pt] Write an SQL query to return the name of all employees and the names of projects they are involved in. If an employee is not involved in any projects, display NULL for the project name.

Query:

```
select e.employee_name, p.project_name
from Employees e
left join Employee_project ep
 on e.employee_id = ep.employee_id
left join Projects p
 on ep.project_id = p.project_id;
```

Q6 [10 pt] Write an SQL query to return all customers who have never left a review, showing their name and email.

Query:

```
select c.customer_name, c.email
from Customers c
left join Reviews r
on c.customer_id = r.customer_id
where r.customer_id is NULL;
```

Q7 [10 pt] Write an SQL query to return all hotel bookings that occurred between March 1, 2024, and June 30, 2024, showing the booking ID, customer name, hotel name, and total cost.

Query:
Following query is written considering only check_in when bookings done between the timelines

```
select booking_id, customer_name, hotel_name, total_cost
from Customers c
join Bookings b
on c.customer_id = b.customer_id
join Hotels h
on h.hotel_id = b.hotel_id
where check_in between '2024-03-01' and '2024-06-30'
order by booking_id, check_in;
```

Following query is written considering if check_in is done prior and check_out booking is done between the timelines

```
select booking_id, customer_name, hotel_name, total_cost
from Customers c
join Bookings b
on c.customer_id = b.customer_id
join Hotels h
```

```
on h.hotel_id = b.hotel_id
where check_in between '2024-03-01' and '2024-06-30'
 or check_out between '2024-03-01' and '2024-06-30'
order by booking_id, check_in;
```

Q8 [10 pt] Write an SQL query to return the top 3 longest hotel stays by customers, showing the customer name, hotel name, and number of nights stayed.

Query:

```
select c.customer_name,
    h.hotel_name,
    datediff(b.check_out, b.check_in) as nights_stayed
from Customers c
join Bookings b
on c.customer_id = b.customer_id
join Hotels h
on b.hotel_id = h.hotel_id
where datediff(b.check_out, b.check_in) >= (
 select min(nights_stayed)
 from (
   select distinct datediff(check_out, check_in) as nights_stayed
   from Bookings
   order by nights_stayed desc limit 3) as top3
       )
order by nights_stayed desc;
```

Q9 [10 pt] Write an SQL query to find all customers who have made more than one hotel booking, displaying the customer name and the number of bookings they have made.

Query:

```
select c.customer_name, count(booking_id) as number_of_bookings
from Customers c
join Bookings b
on c.customer_id = b.customer_id
group by c.customer_id
having count(booking_id) > 1;
```