Homework 8

Due November 25 by 11:59 PM

For Questions 1-6, use subqueries or CTEs. For questions 7-10, use CASE expressions.

1) Find employees whose salary is higher than any employee in the 'IT' or 'Accounting' department. (***HR database, employees & departments tables***). *Use ANY operator.*

Note: Do not just use department IDs 6 and 11 for IT and Accounting departments. Write a nested subquery to retrieve department IDs for IT and Accounting.

```
select first_name, last_name, salary from employees
where salary > any (select salary from employees
join departments
using(department_id)
where department_id in
(select department_id from departments
where department_name in ('IT', 'Accounting')));
```

2) Find employees whose salary is higher than all employees in the 'Finance' department. (***HR database, employees & departments tables***). *Use ALL operator.*

Note: Do not just use department ID for Finance department. Write a nested subquery to retrieve department ID for Finance.

```
select first_name, last_name, salary from employees
where salary > all (select salary from employees
join departments
using(department_id)
where department_id in
(select department_id from departments
where department_name = 'Finance'));
```

3) Find the film titles that have the highest rental rate in their respective rating categories. Return film_id, title, rental rate and rating columns. Make sure rating is listed in descending order. (**dvdrental database, film table**)

```
select film_id, title, rental_rate, rating
from film f1
where rental_rate = (select max(rental_rate) from film f2
where f1.rating = f2.rating)
order by rating desc;
```

4) Create a CTE that calculates the total number of rentals made by each customer. Then, write a query using this CTE to find customers who have rented more than 35 times. (**dvdrental database, customer & rental tables**)

Hint: First, create a CTE using the rental table. Then, use this CTE to join it with the customers table to retrieve customer_id, first name, and last name of each customer along with the rental count for each customer.

```
with rental_count as
(
select customer_id, count(rental_id) as count_num
from rental
group by customer_id
)
select customer_id, first_name, last_name, count_num
from customer
join rental_count
using(customer_id)
where count_num > 35;
```

5) Find all categories that have at least one product with a unit price greater than $80. (**northwind database, categories & products tables**)

```
select category_id, category_name
from categories c
where exists(select category_id from products p
where c.category_id = p.category_id
and unit_price > 80);
```

```
select category_id, category_name
from categories
where category_id in (select category_id from products
where unit_price > 80);
```

6) Write a SQL code that shows all the payments together with how much the payment amount is below the maximum payment amount. (**dvdrental database, payment table**)

```
select amount, (select max(amount) from payment) - amount as diff from payment;
```

7) Find the number of films in "R", "PG", "PG-13" rating categories by using ***CASE-WHEN statement.*** (***dvdrental database, film table***). *Don't use COUNT function.*

```
select sum(
case
        when rating = 'R' then 1
        end) as R,
sum(case
        when rating = 'PG' then 1
```

```
        end) as PG,
sum(case
        when rating = 'PG-13' then 1
        end) as PG13
from film;
```

8) Calculate the average unit price of the products categorized as ***Cheap, Medium***, and ***Expensive.*** Label the products as "Cheap", "Medium", and "Expensive" based on their unit price according to the following conditions: (***northwind database, products table)***

   If the unit price is less than 10, label it as "Cheap"
   If the unit price is between 10 and 20, label it as "Medium"
   Otherwise, label it as "Expensive".

```
select
case
        when unit_price < 10 then 'Cheap'
        when unit_price between 10 and 20 then 'Medium'
        else 'Expensive'
        end as price_category,
avg(unit_price) as average_price
from products
group by price_category;
```

9) Write a query to retrieve customer's name (first_name + last_name) as full name, total number of payments made by each customer, and their corresponding category. (**dvdrental database, customer & payment tables**)

   Determine the category based on the following criteria:

   If a customer has made less than 15 payments, label them as "Regular" customer.

   If a customer has made between 15 and 25 payments, label them as "Frequent" customer.

   If a customer has made more than 25 payments, label them as "Premium" customer.

```
select concat_ws(' ', first_name, last_name) as full_name, count(payment_id) as num_of_payments,
case
when count(payment_id) < 15 then 'Regular'
when count(payment_id) between 15 and 25 then 'Frequent'
else 'Premium'
end as category
from customer
join payment
using(customer_id)
group by full_name;
```

10) Write a query to display each product's name, category, and ***stock status*** as 'Out of Stock', 'Low Stock', or 'In Stock'. (***northwind database, products & categories tables)***

    Determine the stock status based on the following criteria:

    If the number of units in stock (units_in_stock column) is 0, then Out of Stock.

    If the number of units in stock is less than 20, then Low Stock

    Otherwise, In Stock

```
select product_name, category_name,
case
when units_in_stock = 0 then 'Out of Stock'
when units_in_stock > 20 then 'Low Stock'
else 'In Stock'
end as stock_status
from products
join categories
using(category_id);
```