Exercise: Basics of SQL, max 22p

Instructions

Return four .sql files containing answer to each section. Please comment also which command belongs to which task. For example

```
# Task 1

select name, commission from salesman;
```

Each exercise is worth of 0.5p.

Tasks

Before starting this exercise you need the tables from the previous exercise

SECTION 1: SELECT + ORDER BY and LIMIT, 6p

You can create section1.sql file including your answers for this section or also one .sql file including all the answers to all sections in exercise 2 is fine

1. Write SQL statement showing name and commission of all the salesmen.

2. Find salesmen living in 'Paris'. Show name and city of each person.

3. Show all columns from employees whose name includes ‘James’ or ‘Adam’.

4. Find the details of salesmen whose name starts with any letter within 'A' and 'L' (not inclusive). Show salesman_id, name, city, commission for each salesman. You can use between comparison for name field.

5. Find those salesmen whose commission is greater than or equal to 0.13. Show name, commission for each salesman

6. Find the orders, which are delivered by a salesman of ID 5001. Show ord_no, ord_date, purch_amt for each order.

7. Find the orders, which are delivered by a salesperson of ID 5001 and purchace amount is more than 1000. Show ord_no, ord_date, purch_amt for each order.

8. Find the products whose price is between 1000 and 4000. Show ord_no, ord_date, purch_amt for each product.

9. Find all the orders which purchase amount is less than 500€ and done before October 2012 or which purchase amount is greater than 2000 and done in October 2012. Show ord_no, purch_amt, ord_date for each order. Show the results in increasing order according to purch_amt.

10. Find the orders done in October 2012 and order the results by customer_id. Show all columns for each order.

11. Find 5 most recent orders. Show all the columns for each order

12. Find all customers. Order the results by city and cust_name.

SECTION 2: AGGREGATES + GROUP BY, 7p

1. Write SQL query to calculate average purchase amount of all orders.
2. Write SQL query to calculate total purchase amount of all orders.
3. Write SQL query to count the number of orders.
4. Write SQL query to count the number salesmen.
5. Write SQL query to count the number of orders after 2012-07-01.
6. Write SQL query to find the maximum purchase amount.
7. Write SQL query to find the lowest purchase amount ordered by each customer. Show customer ID and minimum purchase amount for each customer. Order the results by customer_id.
8. Write SQL query to find the highest purchase amount ordered on a particular date. Show order date and the highest purchase amount.
9. Write SQL query to find highest purchase amount by each customer. Show only those results where highest amount is more than 2000€. Show customer id and maximum purchase amount.
10. Write SQL query to find the maximum purchase amount by each salesman whose ID is between 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount for each row.
11. Write SQL query to count how many orders haven been made on '2012-08-17'. Return number of orders.
12. Show how many orders have been made each date.
13. Write SQL query to calculate average purchase amount of each salesman. Show salesman_id and average purchase amount. Order the results in decreasing order by average purchase amount
14. Write SQL query to find the purch_amt which are greater than the average order value of 10th October 2012. Return ord_no, purch_amt, ord_date, customer_id, salesman_id

SECTION 3: JOINS and UNIONS, VIEWS 6p

1. Write SQL query to find all the orders. Instead of showing customer_id you should show cuostomer's name. Show ord_no, purch_amt, ord_date, customer_name for each order. You need to join orders and customer tables.

2. Write SQL query to find all the orders. Instead of showing salesman id you should show salesman's name and instead of showing customer id you should show customer name. Return ord_no, purch_amt, ord_date, customer name and salesman name. You need to join orders, salesman and customers tables.

3. Find the salesperson and customer who belong to same city. Return Salesman, cust_name and city. Order the results by city.

4. Calculate the average purchase amount of each customer. Return customer name and average purchase. Return also the rows for those customers who have not purchased anything (Marion Cameron)
   Hint: You need information from two tables.
   Hint 2: You should include both id and cust_name to group by clause

5. Write SQL query to find all customers and their salesman. Show customer name and salesman name. You need information from salesman and customer tables.

6. Find all customers who are not living in the same city as his/her salesman. Show cust_name and his / her city + salesman's name and his/her city.

7. Find all the orders issued by the salesman 'Paul Adam'. Show ord_no, purch_amt, ord_date and salesman's name.

8. Write SQL query to find all the orders, which are generated by those salesmen, who live in the city of London. Show ord_no salesman's name and city.

9. Write a SQL query to find the commission of the salesmen whose customer is in Paris. Return salesman id, salesman name and commission

10. Create a view for those salesmen living in the city 'Paris'. Show Salesman id, name and city

11. Create a view for showing all customers with following information: customer_id, cust_name and salesman's name.

12. Create a view to compute average purchase amount and total purchase amount for each salesperson. Return name, average purchase and total purchase amount. (Assume all names are unique).

SECTION 4: UPDATES, DELETES 3p

1. Update salesman whose id is 5007 name to be Paul White and his city to London.

2. Update customer whose id is 3005 grade to be 300.

3. Change salesman whose id is 5007 id to be 5009. should notice the contradiction if you try to modify the id because it is referenced as foreign key from customer and orders table. There are two options to do the task given. (1p)

   Option 1: Drop the foreign key constraint from customer and order table first and then make a change and create the foreign key constraints again to customer and orders.

   Option 2: Insert “new” correct information to salesman. Then update the information to other two tables. And finally remove the outdated information.

4. Delete order number 70014 from orders.

5. Delete salesman whose id is 5010 and all the customers who has the reference to this salesman.
   Hint: You have to delete first the customers who has the reference to salesman_id 5010.