**Task 2: SQL Basics** (Tuomas Pasanen)

This exercise just simply has pictures of the commands, under the specified task. Database is the same one detailed in task 1.

1. Write a SQL statement to display columns name and commission for all the salesmen.

A picture containing table

Description automatically generated

2. Find the salespeople who lives in the City of 'Paris'. Return salesperson's name, city.

Text

Description automatically generated

3. Find the details of all employees whose name includes ‘James’ or ‘Adam’. Return

salesman\_id, name.

Text

Description automatically generated

4. Find the details of those salespeople whose name starts with any letter within 'A' and 'L'

(not inclusive). Return salesman\_id, name, city, commission.

Text

Description automatically generated

5. Find those salesmen whose commission is greater than or equal to 0.13. Return name,

commission.

Text

Description automatically generated

6. Find the orders, which are delivered by a salesperson of ID. 5001. Return ord\_no,

ord\_date, purch\_amt

Text

Description automatically generated

7. Find the orders, which are delivered by a salesperson of ID. 5001 and purchace amount is more than 1000. Return ord\_no, ord\_date, purch\_amt

Text

Description automatically generated

8. Find the products whose price is in the range 1000 to 4000. Begin and end values are included. Return ord\_no, ord\_date, purch\_amt.

Text

Description automatically generated

9. Write a SQL query to 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. Return ord\_no, purch\_amt, ord\_date.

Text

Description automatically generated

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

Graphical user interface, text

Description automatically generated

11. Update customer’s whose id is 3005 grade to be 300.

Graphical user interface

Description automatically generated

12. Change salesman whose id is 5007 id to be 5009.

First drop the constraints:





Now alter the salesman\_id:

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

Now add the constraints back:

Text

Description automatically generated

Text

Description automatically generated

The constraints in action:

Text

Description automatically generated

13. Delete order number 70014 from orders.

Graphical user interface

Description automatically generated

14. Delete salesman whose id is 5010 and all the customers who has the reference to this salesman.

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

15. Write a SQL query to calculate average purchase amount of all orders. Return average purchase amount

Text

Description automatically generated

Rounded:

Text

Description automatically generated

16. Write a SQL query to calculate the average price for purchase amount of salesman 5001.

Text

Description automatically generated

17. Write a SQL query to calculate total purchase amount of all orders. Return total purchase amount.

A screenshot of a computer

Description automatically generated with medium confidence

18. Write a SQL query to count the number of orders.

Text

Description automatically generated

19. Write a SQL query to count the number of unique salespeople. Return number of salespeople.

Text

Description automatically generated

Since salesman\_id is a primary key, we don’t necessarily have to do DISTINCT(). There won’t be duplicates of id’s in any case.

20. Write a SQL query to count the number of orders after 2012-07-01.

Text

Description automatically generated

21. Write a SQL query to count the number of orders in October 2012.

Text

Description automatically generated

22. Write a SQL query to find the maximum purchase amount.

A screenshot of a computer

Description automatically generated with medium confidence

23. Write a SQL query to find the lowest purchase amount ordered by each customer. Return customer ID, minimum purchase amount.

Text

Description automatically generated

Ordered by the customer\_id.

24. Write a SQL query to find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.

Text

Description automatically generated

25. Write a SQL query to find highest order (purchase) amount by each customer in a particular order date. Filter the result by highest order (purchase) amount above 2000.00. Return customer id, order date and maximum purchase amount.

Text

Description automatically generated

26. Write a SQL query to find the maximum order (purchase) amount in the range 2000, 4000 (Begin and end values are included.) by combination of each customer and order date.

Text

Description automatically generated

27. Write a SQL query to find the maximum order (purchase) amount generated by each salesperson. Filter the rows for the salesperson ID is in the range 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount.

Text

Description automatically generated

28. Write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders.

Text

Description automatically generated

29. Write a SQL query to calculate average purchase amount of each salesman. Return salesman id and average purchase amount.

Text

Description automatically generated

30. Sort the previous result in decreasing order by the average purchase amount.

Text

Description automatically generated

“round” is the name of the result of the average, so we are ordering by it, instead of purch\_amt.

31. Write a SQL query to find all the orders. Instead of showing salesman id you should show salesman’s name. Return ord\_no, purch\_amt, ord\_date, customer\_id and salesman name. You need to join orders and salesman tables.

Graphical user interface, text

Description automatically generated

I used the AS keyword to import the salesman name as “salesman\_name”, instead of just “name”.

32. Write a 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.

Graphical user interface

Description automatically generated

33. Find the salesperson and customer who belongs to same city. Return Salesman, cust\_name and city.

Graphical user interface, text

Description automatically generated

34. Calculate the average purchase amount of each salesman. Return salesman id, salesman name and average purchase.

My solution rounded to 2 decimal places.

Text

Description automatically generated

35. Change the previous so that it also shows those salesman who have not sold anything.

Text

Description automatically generated with medium confidence

36. Write a SQL query to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission. You need information from salesman and customer tables.

A picture containing graphical user interface

Description automatically generated

37. Write a SQL query to find those salespersons do not live in the same city where their customers live and received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, salesman city, commission.

Text

Description automatically generated

38. Find all the orders issued by the salesman 'Paul Adam'. Return ord\_no, purch\_amt, ord\_date, customer\_id and salesman\_id. You can use subquery or join.

Paul Adam doesn’t exist, So I used Paul White. (See task 10)

With join:

Text

Description automatically generated

With subquery:

Graphical user interface, text

Description automatically generated

39. Write a SQL query to find all the orders, which are generated by those salespeople, who live in the city of London. Return ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id. You can use subquery or join.

With join:

Text

Description automatically generated

With subquery:

Graphical user interface, text

Description automatically generated

40. Write a SQL query to find the orders generated by the salespeople who works for customers whose id is 3007. Return ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id. A customer can works only with a salespeople.

Graphical user interface, text

Description automatically generated

41. Write a SQL query to find the order values greater than the average order value of 10th October 2012. Return ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id.

Text

Description automatically generated

42. Write a SQL query to find the commission of the salespeople work in Paris City (i.e. whose customer is in Paris). Return salesman id, salesman name and commission.

Text

Description automatically generated

43. Create a view for those salespersons living in the city 'Paris’

Text

Description automatically generated

44. 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).

Text

Description automatically generated