

Lab 04 – Multi-Table Queries and Views

Objective:

The purpose of this lab is to introduce students to querying data from multiple tables. Relationships are used in relational databases to reduce redundant and repetitive data, but it is necessary to reconnect these tables when extracting data and obtaining information. Student will be able to:

- produce query results containing data from multiple tables using ANSI-92 joins and demonstrate their knowledge of inner, outer and full joins.
- To actively troubleshoot queries to handle potentially ambiguous fields across multiple tables through the use of aliases
- Students learn to create and modify views.

Submission:

Your submission will be a single SQL file with the solutions provided. (with a .sql file extension and all the codes and the output in PDF)

Your submission needs to include a comment header block and be commented top clearly indicate the answers to each question. Make sure every SQL statement terminates with a semicolon.

ALL questions must be answered using ANSI-92 JOINS unless otherwise stated. ANSI-89 are obsolete and should not be used in new query derivations. We only teach them in case you see them in the workplace, that you know what they are and how they work.

Tasks:

Select data from multiple tables

1. Create a query that shows employee number, first name, last name, city, phone number and postal code for all employees in France.
 - a. Answer this question using an ANSI-89 Join
 - b. Answer this question using an ANSI-92 Join
2. Create a query that displays all payments made by customers from Canada.
 - a. Sort the output by Customer Number.
 - b. Only display the Customer Number, Customer Name, Payment Date and Amount.
 - c. Make sure the date is displayed clearly to know what date it is. (i.e. what date is 02-04-19??? – Feb 4, 2019, April 2, 2019, April 19, 2002,)
3. Create a query that shows all USA customers who have not made a payment. Display only the customer number and customer name sorted by customer number.

4. a) Create a view (***vwCustomerOrder***) to list all orders with the following data for all customers:
Customer Number, Order number, order date, product name, quantity ordered, and price for each product in every order.
b) Write a statement to view the results of the view just created.
5. Using the ***vwCustomerOrder*** view, display the order information for customer number 124. Sort the output based on order number and then order line number. (Yes, I know orderLineNumber is not in the view)
6. Create a query that displays the customer number, first name, last name, phone, and credit limits for all customers who do not have any orders.
7. Create a view (***vwEmployeeManager***) to display all information of all employees and the name and the last name of their managers if there is any manager that the employee reports to. Include all employees, including those who do not report to anyone.
8. Modify the ***employee_manager*** view so the view returns only employee information for employees who have a manager. Do not DROP and recreate the view – modify it. (Google is your friend).
9. Drop both ***customer_order*** and ***employee_manager*** views.