# Order Entry Database – Intermediate SQL Queries

The simplified Order Entry database was designed to support transactions related to customer purchases of computer and electronics products either over the phone or online. The SQL code for all 12 problems should be entered into a single `MySQL7_OrdEntry.sql` file, separated by comments, but each problem should be run in Workbench independently. The results should be copy/pasted or exported into a single Excel file `Assign7_OrdEntry.xlsx`, one result per sheet.

Each of the questions is worth **0.25** points.

1. List all employees that have not taken any orders. Include the employee number, first and last name, and the number of his or her supervisor.
2. List all customer orders that were ordered by and are going to the same person, and were taken by employees with the commission rate of 4% or greater using Type I subquery. Include customer number, first and last name, order number, date and the name of the person the order is going to. Hint: Use either CONCAT function to combine customer first and last name into a full name.
3. List the customer that have orders for more than one Connex product using nested Type I queries. Include customer first and last name in the result and sort by customer last name.
4. List all the customers that have only shopped online using a Type II subquery. Include customer number, first and last name. Hint: The inner query finds all the customer and employee numbers from the inner order table, matches customer numbers from inner and outer query and checks whether employee number is not null. If that is the case, there are records for that customer with orders taken by an employee, which means that this is not a customer that shopped only online. The outer query lists all the customer numbers, first and last names for all the orders placed by each customer.
5. Find the average number of products and average sales by customer. Show customer number and name and use a grouping subquery within the FROM clause of the main grouping query. Hint: Create the subquery first to list all the orders with customers, number of products and total sales. Then use it in the FROM clause of the outer query to find the averages by customer.
6. List all the employees that have taken orders for ALL ColorMeg products using Type II subquery in the HAVING clause. List the employee number, first and last name.
7. Create a comma delimited list of customer last names by state. Hint: Use GROUP_CONCAT function.
8. Determine if all the customer numbers start with character C, followed by 7 digits. Hint: Use PT10 to walk each customer number, and then use ASCII and SUBSTR functions to extract the ASCII code for each of the characters in the employee number. ASCI code for C is 67, and digits 0 to 9 have ASCII codes between 48 and 57. Then flag each row and develop a scheme for checking the validity of all customer numbers. You should test with an additional customer having an incorrect customer number such as X123Y456.
9. Calculate the daily running sales total. First create Daily_Sales view that will calculate daily sales for each date. Then use a subquery that adds all daily sales from the view up to the current date. Hint: Use two different aliases for Daily_Sales view, which will allow you embed the running total subquery into the SELECT statement of the main query.
10. Calculate the mode of employee commission rates. Mode is the most frequent value. Hint: Create a view that selects different commission rates and counts their frequency. Then find the rate with the highest frequency found by a Type II subquery.
11. List all the order dates starting with the first date an order was placed, ending with the last date and all the days in between, including the ones when no sales were made. Hint: Create a PT100 table first, and then use it to loop from the starting date, adding a day each time, while less than the ending date.
12. Count number of orders for each day in the date range determined in the previous problem, including zeros for days when no orders were placed. Hint: Left-join with the date list from the previous problem.

Submission: You must submit `MySQL7_OrdEntry.sql` SQL script, and the `Assgn7_OrdEntry.xlsx` Excel file on Canvas by the designated due date.