AM05 Data Mgmt – Lab 1

This lab has two parts. The first part asks you to write some SQL queries while the second asks you to interpret and create ERDs.

Part 1. SQL Queries

These queries are based on the ap database used in the hands on exercises for AM05 Session 1. The questions are the same as on p111 of Murach's MySQL (3rd Edition). I have kept the original question numbering so that you can easily find the solutions in the zip file you should have downloaded from the murach.com website if you already completed the hands on exercises. The solutions can be found in the mysql>ex_solutions>ch03 directory.

8. Write a SELECT statement that returns three columns from the Vendors table: vendor_name, vendor_contact_last_name, and vendor_contact_first_name. Then, run this statement to make sure it works correctly.

    Add an ORDER BY clause to this statement that sorts the result set by last name and then first name, both in ascending sequence. Then, run this state-ment again to make sure it works correctly. This is a good way to build and test a statement, one clause at a time.

9. Write a SELECT statement that returns one column from the Vendors table named full_name that joins the vendor_contact_last_name and vendor_contact_first_name columns.

    Format this column with the last name, a comma, a space, and the first name like this:

    `Doe, John`
    Sort the result set by last name and then first name in ascending sequence. Return only the contacts whose last name begins with the letter A, B, C, or E. This should retrieve 41 rows.

10. Write a SELECT statement that returns these column names and data from the Invoices table:

Due Date	The invoice_due_date column
Invoice Total	The invoice_total column
10%	10% of the value of invoice_total
Plus 10%	The value of invoice_total plus 10%

    Return only the rows with an invoice total that's greater than or equal to 500 and less than or equal to 1000. This should retrieve 12 rows.

    Sort the result set in descending sequence by invoice_due_date.

11. Write a SELECT statement that returns these columns from the Invoices table:

invoice_number	The invoice_number column
invoice_total	The invoice_total column
payment_credit_total	Sum of the payment_total and credit_total columns
balance_due	The invoice_total column minus the payment_total and credit_total columns

    Return only invoices that have a balance due that's greater than $50. Sort the result set by balance due in descending sequence.

    Use the LIMIT clause so the result set contains only the rows with the 5 largest balances.

Work with nulls and test expressions

12. Write a SELECT statement that returns these columns from the Invoices table:

invoice_number	The invoice_number column
invoice_date	The invoice_date column
balance_due	The invoice_total column minus the payment_total and credit_total columns
payment_date	The payment_date column

Return only the rows where the payment_date column contains a null value. This should retrieve 11 rows.

13. Write a SELECT statement without a FROM clause that uses the CURRENT_DATE function to return the current date in its default format.

Use the DATE_FORMAT function to format the current date in this format: `mm-dd-yyyy`

This displays the month, day, and four-digit year of the current date.

Give this column an alias of current_date. To do that, you must enclose the alias in quotes since that name is already used by the CURRENT_DATE function.

14. Write a SELECT statement without a FROM clause that creates a row with these columns:

starting_principal	Starting principal of $50,000
interest	6.5% of the principal
principal_plus_interest	The principal plus the interest

To calculate the third column, add the expressions you used for the first two columns.

Part 3. ERD Challenges

In this part you can demonstrate your understanding of ERDs.
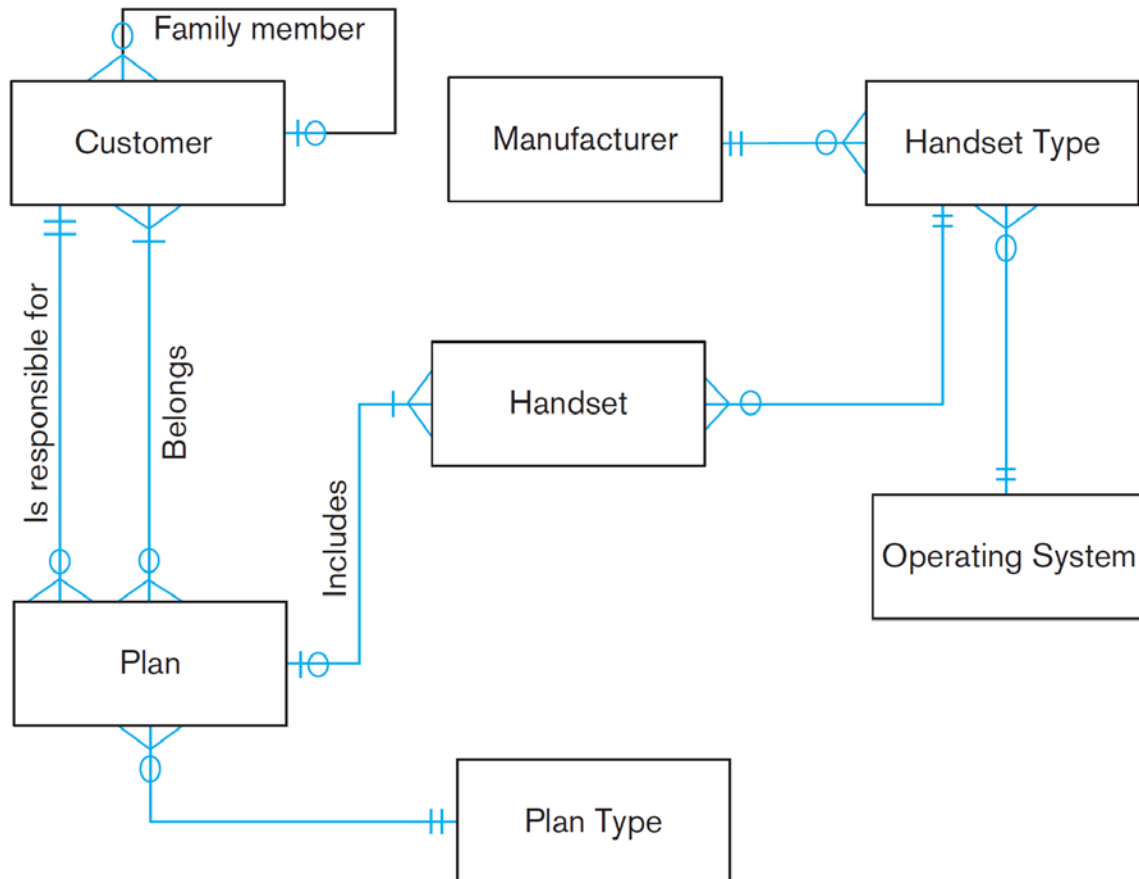
A cellular operator needs a database to keep track of its customers, their subscription plans, and the handsets (mobile phones) that they are using. The ERD above illustrates the key entities of interest to the operator and the relationships between them. Based on the figure, answer the following questions and explain the rationale for your response.

1. Can a customer have an unlimited number of plans?

2. Is it possible to create a plan without knowing who the customer is?

3. Is it possible to maintain data regarding a handset without connecting it to a plan?

4. Is the company able to track a manufacturer without maintaining information about its handsets?

5. Can the same operating system be used on multiple handset types?

6. There are two relationships between Customer and Plan. Explain how they differ.

7. Is it possible to link a handset to a specific customer in a plan with multiple customers?

8. Prepare an ERD for the following business context. Be sure to express the degree and minimum and maximum cardinalities for each relationship.

- A piano manufacturer wants to keep track of all the pianos it makes individually. Each piano has an identifying serial number and a manufacturing completion date. Each instrument represents exactly one piano model, all of which have an identification number and a name. In addition, the company wants to maintain information about the designer of the model. Over time, the company often manufactures thousands of pianos of a certain model, and the model design is specified before any single piano exists.

- A piano manufacturer employs piano technicians who are responsible for inspecting the instruments before they are shipped to the customers. Each piano is inspected by at least two technicians (identified by their employee number). For each separate inspection, the company needs to record its date and a quality evaluation grade.

- The piano technicians have a hierarchy of reporting relationships: Some of them have supervisory responsibilities in addition to their inspection role and have multiple other technicians report to them. The supervisors themselves report to the chief technician of the company.