JOINING TABLES

1. Write a SELECT statement that returns all columns from the Vendors table inner-joined with all columns from the Invoices table. This should return 114 rows.

2. Write a SELECT statement that returns these four columns:

Column Name	Data Returned
vendor_name	The vendor name from Vendors table
invoice_number	The invoice number from Invoices table
invoice_date	The invoice date from Invoices table
balance_due	invoice_total minus payment_total and credit_total

Use these aliases for the tables: v for Vendors and i for Invoices.
Return one row for each invoice with a non-zero balance. This should return 11 rows.
Sort the result set by vendor_name in ascending order.

3. Write a SELECT statement that returns these three columns:

Column Name	Data Returned
vendor_name	The vendor name from Vendors table
default_account	The default account number from Vendors table
description	The account description column from the General Ledger Accounts table.

Return one row for each vendor. This should return 122 rows.
Sort the result set by account_description and then by vendor_name.

4. Write a SELECT statement that returns these five columns:

Column Name	Data Returned
vendor_name	The vendor name from Vendors table
invoice_date	The invoice date from Invoices table
invoice_number	The invoice number from Invoices table
li_sequence	The invoice sequence column from the Invoice Line Items table
li_amount	The line itme amount column from the Invoice Line Items table

Use aliases for the tables. This should return 118 rows.
Sort the final result set by vendor_name, invoice_date, invoice_number, and invoice_sequence.

5. Write a SELECT statement that returns these three columns:

Column Name	Data Returned
vendor_id	The vendor_id column from the Vendors table
vendor_name	The vendor name from Vendors table
contact_name	A concatenation of he vendor_contact_first_name and vendor_contact_last_name with a space between

Return one row for each vendor whose contact has the same last name as another vendor's contact. This should return 2 rows. *Hint: Use a self-join to check that the vendor_id columns aren't equal, but the vendor_contact_last_name columns are equal.*
Sort the result set by vendor_contact_last_name.

6. Write a SELECT statement that returns these three columns:

Column Name	Data Returned
account_number	The account number column from the General Ledger Accounts table
account_description	The account description column from General Ledger Accounts table
invoice_id	The invoice ID column from the Invoice Line Items table

Return one row for each account number that has never been used. This should return 54 rows. *Hint: Use an outer join and only return rows where the invoice_id column contains a null value.*
Remove the invoice_id column from the SELECT clause.
Sort the final result set by the account_number column.

7. Use the UNION operator to generate a result set consisting of to columns from the Vendors table: vendor_name and vendor_state. If the vendor is in California, the vendor_state value should be "CA"; otherwise, the vendor_state value should be "Outside CA." Sort the final result set by vendor_name.

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

Column Name	Data Returned
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.

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

Column Name	Data Returned
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 return 11 rows.

11. Write a SELECT statement without a FROM clause that uses the CURRENT_DATE function to return the current date in its default format. (Cf: Murach 88-89)

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.

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

Column Name	Data Returned
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 expression you used for the first two columns.