Chapter 3

How to retrieve data from a single table

Exercises

Enter and run your own SELECT statements

In these exercises, you'll enter and run your own SELECT statements.

1. Write a SELECT statement that returns four columns from the Products table: product_code, product_name, list_price, and discount_percent. Then, run this statement to make sure it works correctly.

   Add an ORDER BY clause to this statement that sorts the result set by list price in descending sequence. Then, run this statement again to make sure it works correctly. This is a good way to build and test a statement, one clause at a time.

2. Write a SELECT statement that returns one column from the Customers table named full_name that joins the last_name and 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 in ascending sequence.

   Return only the customers whose last name begins with letters from M to Z.

   NOTE: When comparing strings of characters, 'M' comes before any string of characters that begins with 'M'. For example, 'M' comes before 'Murach'.

3. Write a SELECT statement that returns these columns from the Products table:

product_name	The product_name column
list_price	The list_price column
date_added	The date_added column

   Return only the rows with a list price that's greater than 500 and less than 2000.

   Sort the result set in descending sequence by the date_added column.

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

    product_name   The product_name column

    list_price   The list_price column

    discount_percent   The discount_percent column

    discount_amount   A column that's calculated from the previous two columns

    discount_price   A column that's calculated from the previous three columns

Round the discount_amount and discount_price columns to 2 decimal places.

Sort the result set by discount price in descending sequence.

Use the LIMIT clause so the result set contains only the first 5 rows.

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

    item_id   The item_id column

    item_price   The item_price column

    discount_amount   The discount_amount column

    quantity   The quantity column

    price_total   A column that's calculated by multiplying the item price by the quantity

    discount_total   A column that's calculated by multiplying the discount amount by the quantity

    item_total   A column that's calculated by subtracting the discount amount from the item price and then multiplying by the quantity

Only return rows where the item_total is greater than 500.

Sort the result set by item total in descending sequence.

Work with nulls and test expressions

6. Write a SELECT statement that returns these columns from the Orders table:

    order_id   The order_id column

    order_date   The order_date column

    ship_date   The ship_date column

Return only the rows where the ship_date column contains a null value.

7. Write a SELECT statement without a FROM clause that uses the NOW function to create a row with these columns:

      today_unformatted   The NOW function unformatted

      today_formatted   The NOW function in this format:
                                       DD-Mon-YYYY

This displays a number for the day, an abbreviation for the month, and a four-digit year.

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

      price   100 (dollars)

      tax_rate   .07 (7 percent)

      tax_amount   The price multiplied by the tax

      total   The price plus the tax

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