# 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 Customer table: first\_name, billing\_address, last\_name, and customer\_id. 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 in ascending 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.

1. Write a SELECT statement that returns one column from the Customers table named join\_name that joins the last\_name and first\_name columns.

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

**Doe, John**

Sort the result set by first name in descending sequence.

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

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

1. Write a SELECT statement that returns these columns from the Products table: product\_name The product\_name column

list\_price The list\_price column

Return only the rows with a list price that’s greater than 400 and less than 1000.

1. 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

Round the discount\_amount columns to 3 decimal places.

Use the LIMIT clause so the result set contains only the first 7 rows.

1. 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

price\_total A column that’s calculated by multiplying the item price by the quantity

Only return rows where the item\_price is greater than 100.

## Work with nulls and test expressions

1. Write a SELECT statement that returns these columns from the Orders table: order\_id The order\_id column

ship\_date The ship\_date column

Return only the rows where the ship\_date column contains a null value.

1. 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.

1. 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