

# Lab 2 Single-Table SELECT Questions & Outputs

## Single-Table SELECT queries on MGS Schema

### 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.

### Output:

| PRODUCT_CODE | PRODUCT_NAME                         | LIST_PRICE | DISCOUNT_PERCENT |
|--------------|--------------------------------------|------------|------------------|
| 1 sg         | Gibson SG                            | 2517       | 52               |
| 2 les_paul   | Gibson Les Paul                      | 1199       | 30               |
| 3 tama       | Tama 5-Piece Drum Set with Cymbals   | 799.99     | 15               |
| 4 precision  | Fender Precision                     | 799.99     | 30               |
| 5 ludwig     | Ludwig 5-piece Drum Set with Cymbals | 699.99     | 30               |
| 6 strat      | Fender Stratocaster                  | 699        | 30               |
| 7 hofner     | Hofner Icon                          | 499.99     | 25               |
| 8 fg700s     | Yamaha FG700S                        | 489.99     | 38               |
| 9 rodriguez  | Rodriguez Caballero 11               | 415        | 39               |
| 10 washburn  | Washburn D10S                        | 299        | 0                |

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.

### Output:

| FULL_NAME           |
|---------------------|
| 1 Sherwood, Allan   |
| 2 Valentino, Erin   |
| 3 Wilson, Frank Lee |
| 4 Zimmer, Barry     |

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.

#### Output:

| PRODUCT_NAME                           | LIST_PRICE | DATE_ADDED |
|--|------------|------------|
| 1 Tama 5-Piece Drum Set with Cymbals   | 799.99     | 30-JUL-22  |
| 2 Ludwig 5-piece Drum Set with Cymbals | 699.99     | 30-JUL-22  |
| 3 Fender Precision                     | 799.99     | 01-JUN-22  |
| 4 Gibson Les Paul                      | 1199       | 05-DEC-21  |
| 5 Fender Stratocaster                  | 699        | 30-OCT-21  |

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

Use the FETCH operator so the result set contains only the first 5 rows.

Sort the result set by discount price in descending sequence.

#### Output:

| PRODUCT_NAME                           | LIST_PRICE | DISCOUNT_PERCENT | DISCOUNT_AMOUNT | DISCOUNT_PRICE |
|--|------------|------------------|-----------------|----------------|
| 1 Gibson SG                            | 2517       | 52               | 1308.84         | 1208.16        |
| 2 Gibson Les Paul                      | 1199       | 30               | 359.7           | 839.3          |
| 3 Tama 5-Piece Drum Set with Cymbals   | 799.99     | 15               | 119.9985        | 679.9915       |
| 4 Fender Precision                     | 799.99     | 30               | 239.997         | 559.993        |
| 5 Ludwig 5-piece Drum Set with Cymbals | 699.99     | 30               | 209.997         | 489.993        |

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.

**Output:**

|   | ITEM_ID | ITEM_PRICE | DISCOUNT_AMOUNT | QUANTITY | PRICE_TOTAL | DISCOUNT_TOTAL | ITEM_TOTAL |
|---|---------|------------|-----------------|----------|-------------|----------------|------------|
| 1 | 5       | 1199       | 359.7           | 2        | 2398        | 719.4          | 1678.6     |
| 2 | 3       | 2517       | 1308.84         | 1        | 2517        | 1308.84        | 1208.16    |
| 3 | 1       | 1199       | 359.7           | 1        | 1199        | 359.7          | 839.3      |
| 4 | 11      | 799.99     |                 | 120      | 799.99      | 120            | 679.99     |
| 5 | 9       | 799.99     |                 | 240      | 799.99      | 240            | 559.99     |

**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.

**Output:**

| ORDER_ID | ORDER_DATE | SHIP_DATE |
|----------|------------|-----------|
| 6        | 31-MAR-22  | (null)    |
| 8        | 02-APR-22  | (null)    |
| 9        | 03-APR-22  | (null)    |

7. Write a SELECT statement that uses the SYSDATE function to create a row with these columns:

today\_unformatted    The SYSDATE function unformatted  
today\_formatted      The SYSDATE function in this format:  
                          MM/DD/YYYY

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

Use a FROM clause that specifies the Dual table.

**Output:**

| TODAY_UNFORMATTED | TODAY_FORMATTED |
|-------------------|-----------------|
| 04-SEP-24         | 09/04/2024      |

8. Write a SELECT statement 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.

Use a FROM clause that specifies the Dual table.

**Output:**

| PRICE | TAX_RATE | TAX_AMOUNT | TOTAL |
|-------|----------|------------|-------|
| 100   | 0.07     | 7          | 107   |