Lab 2
1-Table SELECT

1. Objectives

After the successful completion of this lab, you will be able to

- Use the tool Oracle SQL Developer to edit and run SQL statements, and find the information about tables.
- Write a query to retrieve data from a single table, involving the following topics
  - Base Table, Calculated Values, Column Alias, ROWNUM pseudo column
  - DISTINCT keyword, Dual table, Scalar functions, Operators (concatenation ||, BETWEEN AND, IS NULL), ORDER BY

2. Tasks to Complete

Complete the questions about 1-table SELECT queries on the MGS Database included in the later part of this document. These queries use the tables in user **mgs**.

NOTE: the links to online Oracle SQL Language references are available in the Modules\Resources Related to SQL Programming on Canvas.

3. Submission Requirements

Just submit a .sql file.

Mark each query based on the question number. Write your FULL name on the first page.

Sample:

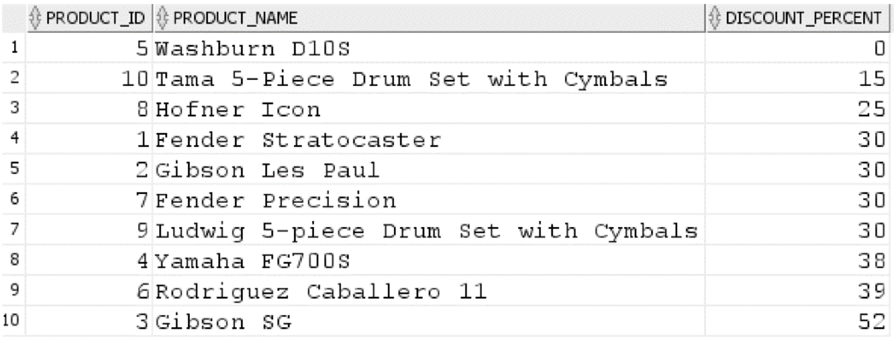
```
--Lab 2
--Your full name

--Q1
Your solution.

--Q2
Your solution.
…
```

Then submit this **SQL script file** by attaching it to the link **Lab 2** in folder **Assignments\Labs** on Canvas.

Single-Table SELECT queries on MGS Database

1. Print the product ids, product names, and discount percentages of all products. Sort the result first in the increasing order of discount percentages, then the increasing order of product ids.
   NOTE: remember to build and test a statement one clause at a time. First build and test SELECT clause and FROM clause, then add ORDER BY clause.

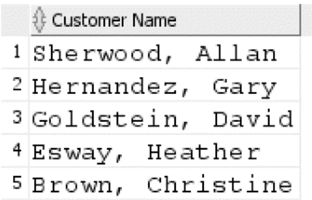
   Output:



	PRODUCT_ID	PRODUCT_NAME	DISCOUNT_PERCENT
1	5	Washburn D10S	0
2	10	Tama 5-Piece Drum Set with Cymbals	15
3	8	Hofner Icon	25
4	1	Fender Stratocaster	30
5	2	Gibson Les Paul	30
6	7	Fender Precision	30
7	9	Ludwig 5-piece Drum Set with Cymbals	30
8	4	Yamaha FG700S	38
9	6	Rodriguez Caballero 11	39
10	3	Gibson SG	52

2. Print the full names of all customers whose last names begin with letters from 'B' to 'T' in the decreasing order of last names.
   Display the query result in the following format:
   Heading of the query result: `Customer Name`
   Data format: last name, then a comma and a space, then first name, like below:
   `Trump, Donald`
   HINT: you need to use string value comparisons and string concatenation.

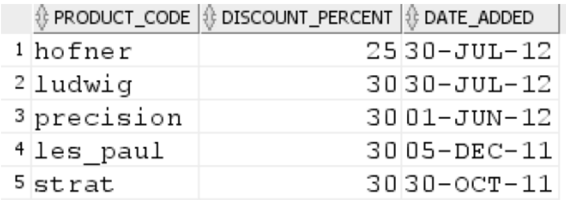
   Output:



	Customer Name
1	Sherwood, Allan
2	Hernandez, Gary
3	Goldstein, David
4	Esway, Heather
5	Brown, Christine

3. Print the product codes, discount percentages, and dates when the products were added for products whose discount percentages are at least 20% and at most 30%. Print the products added last (i.e. the latest product) first, the products added earliest (i.e. the oldest product) last.
   HINT: you need to sort the query result by date values.

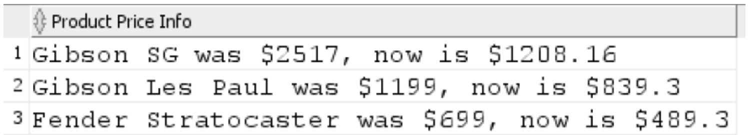
   Output:



	PRODUCT_CODE	DISCOUNT_PERCENT	DATE_ADDED
1	hofner	25	30-JUL-12
2	ludwig	30	30-JUL-12
3	precision	30	01-JUN-12
4	les_paul	30	05-DEC-11
5	strat	30	30-OCT-11

4. Print at most 3 products' names, listing prices, the actual prices after discounts in the decreasing order of the actual prices in the format below. These 3 products are NOT necessarily the 3 most expensive products.
HINT: you need to sort the query result by an expression, and use a pseudo-column to limit the number of rows in the query result.

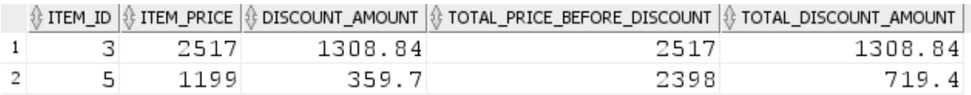
Output:



	Product Price Info
1	Gibson SG was $2517, now is $1208.16
2	Gibson Les Paul was $1199, now is $839.3
3	Fender Stratocaster was $699, now is $489.3

5. For each item with a total discount amount more than $600, print this item's item id, unit price before discount, unit discount amount, total price before discount, and total discount amount.
Display the total price before discount as TOTAL_PRICE_BEFORE_DISCOUNT, the total discount amount as TOTAL_DISCOUNT_AMOUNT.
Sort the query result in the decreasing order of the total discount amounts.
NOTE: Multiple units of the same item may be purchased in the same order. So for such an item, the total price before discount and the total discount amount includes all units of this item in the purchase order.

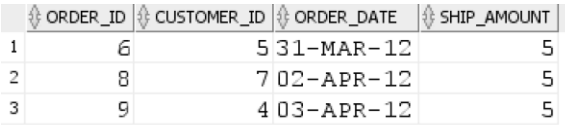
Output:



	ITEM_ID	ITEM_PRICE	DISCOUNT_AMOUNT	TOTAL_PRICE_BEFORE_DISCOUNT	TOTAL_DISCOUNT_AMOUNT
1	3	2517	1308.84	2517	1308.84
2	5	1199	359.7	2398	719.4

6. Print the order ids, customer ids, dates when orders were placed, and the shipping amounts for all orders whose shipping dates are unknown.

Output:



	ORDER_ID	CUSTOMER_ID	ORDER_DATE	SHIP_AMOUNT
1	6	5	31-MAR-12	5
2	8	7	02-APR-12	5
3	9	4	03-APR-12	5

7. Print a greeting message like below. You must print the heading as is and print the date value as the date when the query is executed. It means that if you run the query on different dates, then the date value in the query result will change. So the date value in the output below is for reference purpose.

Output:

```
GREETING
-------------------------
Hello! Today is Jan-19-2022
```

8. Given an ipad at $799.99, the tax rate 8%, print the price before tax, tax rate, tax amount, the price after tax like below. Your query must use the computer to calculate the tax amount and price after tax.

Please do NOT hard code the tax amount and the price after tax in your query.

NOTES:

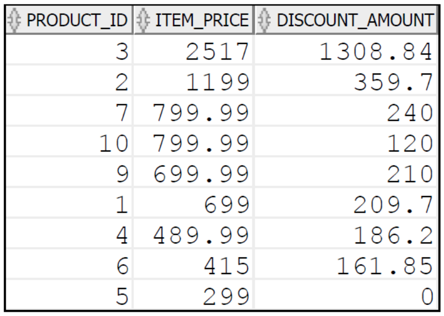
- DO NOT CREATE a NEW TABLE WHEN COMPLETING THIS QUESTION.
- DO NOT INSERT any new row into any existing table.
- And you do NOT NEED a new table or new rows to complete this question.

Output:

```
PRICE_BEFORE_TAX  TAX_RATE   TAX_AMOUNT      PRICE_AFTER_TAX
-----------------------------------------------------------
799.99            0.08       63.9992         863.9892
```

9. Print the product id, the unit price before the discount, and the discount amount of each product that was ordered. If a product was ordered more than once, print its information only once. Sort the result in the decreasing order of the unit price before the discount.
NOTE: The product returned must have been ordered.
HINT: you need to remove the duplicates using a keyword.

Output:



PRODUCT_ID	ITEM_PRICE	DISCOUNT_AMOUNT
3	2517	1308.84
2	1199	359.7
7	799.99	240
10	799.99	120
9	699.99	210
1	699	209.7
4	489.99	186.2
6	415	161.85
5	299	0