Lab 3
1-Table SELECT, N-Table SELECT (INNER JOIN, Self Join)

1. Objectives

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

- Read and write queries that retrieve data from a single table using the following features
  - Pattern matching: %, -, LIKE
  - Scalar function: TO_CHAR
  - Duplicate removal: DISTINCT
- Read and write queries that retrieve data from two or more tables using the following features
  - INNER JOIN, Self Join, table alias, column qualification

2. Tasks to Complete

Complete the questions about 1-table SELECT and N-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 folder 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 3
--Your full name

--Q1
Your solution.

--Q2
Your solution.
…
```

Then submit this SQL script file by attaching it to the link **Lab 3** in folder **Assignments\Labs** on Canvas.

Single-Table SELECT queries on MGS Database

1. Print the IDs, street addresses (including apartment number and suite number), cities, and states of customers whose street addresses contains "**wood**". Sort the query result first in the increasing order of states, then in the increasing order of cities.

   HINTS
   - **You need to specify a pattern using special characters (% and/or _) and use a pattern comparison operator LIKE.**
   - **Single quotes for character literals. Double quotes for column aliases.**

   Output:

CUSTOMER_ID	LINE1	LINE2	CITY	STATE
8	291 W. Hollywood Blvd.	(null)	Los Angeles	CA
1	100 East Ridgewood Ave.	(null)	Paramus	NJ
1	21 Rosewood Rd.	(null)	Woodcliff Lake	NJ

2. Print the first names, last names, and email addresses of customers who have **r** as the third letter in their first names. Sort the result in the increasing order of first names.

   HINTS
   - **You need to specify a pattern using special characters (% and/or _) and use a pattern comparison operator LIKE.**
   - **Single quotes for character literals. Double quotes for column aliases.**

   Output:

FIRST_NAME	LAST_NAME	EMAIL_ADDRESS
Barry	Zimmer	barryz@gmail.com
Christine	Brown	christineb@solarone.com
Gary	Hernandez	gary_hernandez@yahoo.com

N-Table SELECT queries on MGS Database

3. Print the names, listing prices, and category names of all products. Sort the result first in the increasing order of category name, and then in the increasing order of product names.

   OUTPUT:

PRODUCT_NAME	LIST_PRICE	CATEGORY_NAME
Fender Precision	799.99	Basses
Hofner Icon	499.99	Basses
Ludwig 5-piece Drum Set with Cymbals	699.99	Drums
Tama 5-Piece Drum Set with Cymbals	799.99	Drums
Fender Stratocaster	699	Guitars
Gibson Les Paul	1199	Guitars
Gibson SG	2517	Guitars
Rodriguez Caballero 11	415	Guitars
Washburn D10S	299	Guitars
Yamaha FG700S	489.99	Guitars

4. Print the first name, last name, and all the postal addresses of the customer whose email address is heatheresway@mac.com. Print all components of an address: street address (including apartment number and suite number), city, state, and zip code.

NOTES&HINTS:

- **Each address in the table `addresses` can be for shipping only, billing only, or for both shipping and billing.**
- **Address_id is primary key in the table `addresses`.**
- **No two customers have the same address.**
- **Single quotes for character literals. Double quotes for column aliases.**

OUTPUT:

FIRST_NAME	LAST_NAME	LINE1	LINE2	CITY	STATE	ZIP_CODE
Heather	Esway	2381 Buena Vista St.	(null)	Los Angeles	CA	90023
Heather	Esway	291 W. Hollywood Blvd.	(null)	Los Angeles	CA	90024

5. Print the first names, last names, billing addresses of all customers. Sort the result first in the increasing order of states, then in the increasing order of cities. Print all components of an address: street address (including apartment number and suite number), city, state, and zip code.

NOTES:

- **Each address in the table `addresses` can be for shipping only, billing only, or for both shipping and billing.**
- **Address_id is primary key in the table `addresses`.**
- **No two customers have the same address.**

OUTPUT:

FIRST_NAME	LAST_NAME	LINE1	LINE2	CITY	STATE	ZIP_CODE
Erin	Valentino	6982 Palm Ave.	(null)	Fresno	CA	93711
Heather	Esway	291 W. Hollywood Blvd.	(null)	Los Angeles	CA	90024
David	Goldstein	1374 46th Ave.	(null)	San Francisco	CA	94129
Frank Lee	Wilson	23 Mountain View St.	(null)	Denver	CO	80208
Barry	Zimmer	16285 Wendell St.	(null)	Omaha	NE	68135
Allan	Sherwood	21 Rosewood Rd.	(null)	Woodcliff Lake	NJ	07677
Gary	Hernandez	3829 Broadway Ave.	Suite 2	New York	NY	10012
Christine	Brown	19270 NW Cornell Rd.	(null)	Beaverton	OR	97006

6. Print all customers' first names, last names, the dates when they placed their orders, and the dates when their orders were shipped. Print the heading for the dates as SHIP_DATE, ORDER_DATE. Print all dates in the following format: 03/31/2012.

HINTS:

- **You need to use a scalar function `TO_CHAR` covered in Week 2 Module to convert the date values from the default format to the required format.**
- **Single quotes for character literals. Double quotes for column aliases.**
- **Null values in the result are NOT because of OUTER JOIN**

OUTPUT:

FIRST_NAME	LAST_NAME	ORDER_DATE	SHIP_DATE
Allan	Sherwood	03/28/2012	03/30/2012
Allan	Sherwood	03/29/2012	03/31/2012
Barry	Zimmer	03/28/2012	03/29/2012
Christine	Brown	03/30/2012	04/03/2012
David	Goldstein	04/03/2012	(null)
David	Goldstein	03/31/2012	04/02/2012
Erin	Valentino	03/31/2012	(null)
Frank Lee	Wilson	04/01/2012	04/03/2012
Gary	Hernandez	04/02/2012	(null)

7. Print customers' email addresses, the dates when their orders were shipped, the product id and the actual price (after the discount) of each product they ordered. Print the heading for the actual price as ACTURAL_PRICE.

   OUTPUT:

EMAIL_ADDRESS	SHIP_DATE	PRODUCT_ID	ACTUAL_PRICE
allan.sherwood@yahoo.com	31-MAR-12	6	253.15
allan.sherwood@yahoo.com	31-MAR-12	3	1208.16
allan.sherwood@yahoo.com	30-MAR-12	2	839.3
barryz@gmail.com	29-MAR-12	4	303.79
christineb@solarone.com	03-APR-12	2	839.3
david.goldstein@hotmail.com	(null)	1	489.3
david.goldstein@hotmail.com	02-APR-12	5	299
erinv@gmail.com	(null)	5	299
frankwilson@sbcglobal.net	03-APR-12	9	489.99
frankwilson@sbcglobal.net	03-APR-12	7	559.99
frankwilson@sbcglobal.net	03-APR-12	1	489.3
gary_hernandez@yahoo.com	(null)	10	679.99

8. Print the product code, name, and discount percentage of each product that has the **same discount percentage** as another product.
   Do not print a product more than once.
   Sort the query result in the increasing order of product codes.

   HINTS:

   - **You need to use a Self Join (i.e. Join a table with itself).**
   - **Make sure that you use table aliases.**
   - **Make sure that you use a table name to qualify common columns.**
   - **Use `DISTINCT` to remove potential duplicates.**
   - **Make sure that you eliminate the rows where a product is matched with itself.**

   OUTPUT:

PRODUCT_CODE	PRODUCT_NAME	DISCOUNT_PERCENT
les_paul	Gibson Les Paul	30
ludwig	Ludwig 5-piece Drum Set with Cymbals	30
precision	Fender Precision	30
strat	Fender Stratocaster	30