No: ....   Date: ....

DATABASE SECTION 06 - Lab Exercises SQL 3

Name : Mohamad Azofri bin Abd Majid

Matric Numbers : B23CS0044

EXERCISE 1

→ Part 1 : Retrieving all columns from a table

1. Customers

```
SELECT * FROM CUSTOMERS ;
```

2. Teams

```
SELECT * FROM TEAMS ;
```

3. Items

```
SELECT * FROM ITEMS ;
```

→ Part 2 : Selecting Specific Columns

1.

```
SELECT ctr_number, first_name, last_name, email, phone_number FROM
CUSTOMERS ;
```

2.

```
SELECT name, number_of_players FROM TEAMS ;
```

3.

```
SELECT name, description, category FROM ITEMS ;
```

STANDARD

~~A~~ EXERCISE 2

-> Part 1 : ~~US~~ Using Arithmatic Operators

1.
```
SELECT first_name, last_name, ctr_number, current_balance,
current_balance / 12 AS monthly_payment FROM CUSTOMERS
```

2. ~~IE~~
```
SELECT first_name, last_name, ctr_number, current_balance,
            current_balance - 5.00 AS after_discount
    FROM CUSTOMERS
```

3. There will be a negative value ~~for~~ after reduce the current balance. This is not suitable to implement into the system since the value is invalid

-> Part 2 : Using Column Aliases

1.
```
SELECT firs_name AS "First Name", last_name AS "Last Name",
          current_balance AS "Balance", current_balance / 12 AS
          "Monthly Repayments"
FROM CUSTOMERS;
```

-> Part 3 : Using Literal Character Strings

1.
```
SELECT 'The ' || name || ' team has ' || number_of_players ||
         ' and receives a discount of ' || discount || ' percent'
       AS "Team Information"
FROM TEAMS;
```

2 Because the discount value is null

EXERCISE 3

→ Part 1 : Using the WHERE Clause

1.
```
SELECT * FROM CUSTOMERS WHERE ctr-number = 'c01986';
```

2.
```
SELECT first_name AS "First Name", last_name AS "Last Name",
       ctr_number AS "Customer Number"
FROM CUSTOMERS
WHERE current_balance > 100 ;
```

3.
```
SELECT id AS "Order ID", odr_date AS "Order Date",
       odr_time AS "Order Time"
FROM ORDERS
WHERE odr_date < TO_DATE('28-MAY-19');
```

→ Part 2 : Range Conditions : BETWEEN Operator

1.
```
SELECT id AS "Inventory ID", cost AS "Cost",
       units AS "Number Of Units"
FROM INVENTORY_LIST
WHERE cost BETWEEN 3.00 AND 15.00 ;
```

→ Part 3 : Membership Conditions : IN operator

1.
```
SELECT id AS "Inventory ID", cost AS "Cost",
       units AS "Number Of Units"
FROM INVENTORY_LIST
WHERE units IN(50, 100, 150, 200);
```

→ Part 4 : Membership Conditions : NOT IN Operator

1.
```
SELECT id AS “Inventory ID”, cost AS “Cost”,
        units AS “Number Of Units”
FROM INVENTORY_LIST
WHERE units NOT IN(50, 100, 150, 200);
```

→ Part 5 : Pattern Matching : LIKE Operator

1.
```
SELECT itm_number AS “Item Number”, name AS “Name”
FROM ITEMS
WHERE name LIKE ‘g%’;
```

→ Part 6 : Pattern Matching : Combining Wildcard Characters with the LIKE Operator

```
SELECT itm_number AS “Item Number”, name AS “Name”
FROM ITEMS
WHERE name LIKE ‘%o%’;
```

✲ EXERCISE 4

→ Part 1 : Using the NULL Conditions

```
1. SELECT 'The ' || name || ' team has ' || number_of_players ||
          ' and does not receive a discount.'
          AS "Team Information"
FROM TEAMS
WHERE discount IS NULL;

2. SELECT 'The ' || name || ' team has ' || number_of_players ||
          ' and receives a discount of ' || discount || ' percent.'
          AS "Team Information"
FROM TEAMS
WHERE discount IS NOT NULL;
```

→ Part 2 : Logical Operators : AND

```
1  SELECT ctr_number AS "Customer Number", address_line_1 AS "Street
          Address", zip_code AS "Postal Code" FROM CUSTOMERS_ADDRESSES
WHERE address_line_2 = 'Starford' AND city = 'Liverpool';
```

→ Part 3 : Logical Operators : OR

```
1. SELECT ctr_number AS "Customer Number", address_line_1 AS "Street
          Address", zip_code AS "Postal Code"
FROM CUSTOMERS_ADDRESSES
WHERE address_line_2 = 'Starford' OR city = 'Liverpool';
```

→ Part 4 : Logical Operators : NOT Equal To

```
1. SELECT ctr-number AS "Customer Number", address_line_1 AS "Street
          Address", zip-code AS "Postal Code"
   FROM CUSTOMERS_ADDRESSES
   WHERE city <> 'Liverpool';
```

✻ EXERCISE 5

```
1. SELECT name AS "Team Name", number_of_players AS "Number
          Of Players"
   FROM TEAMS
   ORDER BY name;

2. SELECT name AS "Team Name", number_of_players AS "Number
          Of Players"
   FROM TEAMS
   ORDER BY number_of_players DESC;

3. SELECT name AS "Team Name", number_of_players AS "Players"
   FROM TEAMS
   ORDER BY "Team Name" DESC;
```

EXERCISE 6

→ Part 1 : TOP-N-ANALYSIS

```
1. SELECT first_name || ' ' || last_name AS "Customer Name"
   FROM CUSTOMERS
   FETCH FIRST 3 ROWS ONLY
```

→ Part 2 : Using a Substitution Variable

```
1. ACCEPT commission NUMBER PROMPT 'Enter the commission rate : '

   SELECT first_name AS "First Name", last_name AS "Last Name"
   FROM SALES_REPRESENTATIVES
   WHERE commission_rate = &commission
   ORDER BY last_name ;
```