INF230 assignment 2

1)

DDL (Data Definition Language) - CREATE, ALTER, DROP, RENAME

```
CREATE Product (
Product_ID INT NOT NULL,
Product_Name VARCHAR(20)
primary key (Product_ID));
```

DML (Data Manipulation Language) - SELECT, INSERT, UPDATE, DELETE

```
INSERT INTO Product
VALUES (1, 'Mandler');
```

TCL (Transaction Control Language) - ROLLBACK, COMMIT

```
DELETE FROM Student WHERE AGE = 20;
ROLLBACK;  -- TCL
```

DCL (Data Control Language) - GRANT, REVOKE

```
GRANT SELECT ON User69 FROM Users; -- GRANT → DCL
```

2)

keys are important for keeping a row in a local table unique from the other rows. The constraint of a primary key is that is must be unique and not a NULL value. The foreign key is used to reference another table, usually to another primary key. They can only be one primary key, but there can be several foreign keys referencing several tables. This is to keep the relation between the tables.

3)

The outer join actually generates the Cartesian product of the two relations. For example, customer has 91 rows and employees has 9 rows, so if there is no common column between two tables, outer join returns 819 rows. Therefore, if customers table has 10 rows and employee table has 10 rows, the results table have 10 * 10 = 100 rows.

4)

```
CREATE TABLE northwind.employee_work_hours (
employee_id INT NOT NULL,
login_hours INT,
working_date DATE,
FOREIGN KEY (employee_id) REFeRENCES
northwind.employees(EmployeeID));
```

```
SET foreign_key_checks = 0; # ignore foreign keys
INSERT INTO northwind.employee_work_hours (employee_id, login_hours,
working_date)
VALUES (1, 4, '2022-02-14'),
        (1, 4, '2022-02-07'),
        (2, 8, '2022-02-15'),
        (3, 2, '2022-02-10'),
        (4, 7, '2022-02-10'),
        (8, 5, '2022-02-13'),
        (9, 4, '2022-02-14'),
        (10, 2, '2022-02-11');
```

5)

```
SELECT P.ProductName, S.CompanyName, S.Country, S.Phone
FROM suppliers as S, products as P
WHERE S.supplierID = P.ProductID
ORDER BY S.Country DESC
LIMIT 25;
```

6)

```
SELECT C.ContactName, MAX(OD.Quantity) as Maximum_Quantitycustomers
FROM orders as O
INNER JOIN orderdetails as OD ON O.OrderID = OD.OrderID
INNER JOIN customers as C ON C.CustomerID = O.CustomerID
GROUP BY ContactName;
```

7-1)

```
SELECT CustomerID, AVG(OD.Quantity) AS AVG_Quantity
FROM orders AS O
INNER JOIN orderdetails AS OD ON O.OrderID = OD.OrderID
WHERE CustomerID IN (SELECT CustomerID
                     FROM customers) # Subquery 1: identical CustomerID
between customers and orders table
AND ProductID IN (SELECT ProductID   # Subquery 2: identical ProductID between
orderdetails and products tables
                  FROM products
                  WHERE CategoryID IN (SELECT CategoryID # Subquery 3:
identical CategoryID between products and categories tables
                                       FROM categories
                                       WHERE CategoryName = 'Beverages'))
GROUP BY CustomerID;

/* identical with
SELECT C.ContactName, AVG(OD.Quantity) AS AVG_Quantity
```

```
FROM orders AS O
INNER JOIN orderdetails AS OD ON O.OrderID = OD.OrderID
INNER JOIN customers AS C ON C.CustomerID = O.CustomerID
INNER JOIN products AS P ON OD.ProductID = P.ProductID
INNER JOIN categories AS CTG ON P.CategoryID = CTG.CategoryID
WHERE CTG.CategoryName = 'Beverages'
GROUP BY C.ContactName;
*/
```

7-2)

```
SELECT SUM(Quantity) AS TOTAL_quantity
FROM orders AS O
INNER JOIN orderdetails AS OD ON O.OrderID = OD.OrderID
WHERE ProductID IN (SELECT ProductID
                    FROM products
                    WHERE CategoryID IN (SELECT CategoryID
                                         FROM categories
                                         WHERE CategoryName = 'Produce'));
```

8)

```
CREATE TEMPORARY TABLE prod_count AS (
        SELECT orderdetails.ProductID as Prod_ID,
                count(orderdetails.ProductID) as count_dup
        FROM orderdetails
        GROUP BY orderdetails.ProductID
        HAVING count(orderdetails.ProductID) > 2);

SELECT distinct(ProductName)
FROM  products,prod_count, orderdetails
WHERE prod_count.Prod_ID = products.ProductID
AND (orderdetails.UnitPrice > 20 AND orderdetails.ProductID = products.ProductID);

/* Identical
SELECT P.ProductID, ProductName, CNT, UnitPrice
FROM (SELECT ProductID, count(ALL ProductID) AS CNT
        FROM orderdetails
        GROUP BY ProductID
        HAVING CNT > 3) tmp
INNER JOIN products AS P ON P.ProductID = tmp.ProductID
WHERE UnitPrice > 20;
*/
```

9)

```
SELECT FirstName
```

```
FROM northwind.employees
WHERE (TitleOfCourtesy = "Mrs." OR TitleOfCourtesy = "Ms.")
AND
            (SELECT sum(login_hours)
             FROM northwind.employee_work_hours
             WHERE employee_id = EmployeeID) > 2;
```

10)

```
SET SQL_SAFE_UPDATES = 0;
/*It removes employee id in 2 and 8*/
DELETE FROM employee_work_hours
WHERE employee_id IN (
        SELECT EmployeeID
        FROM employees
        WHERE BirthDate BETWEEN '1950-01-01' AND '1960-05-27');
```