Cairo University
Faculty of Computers and Artificial Intelligence
Information Systems Department

Database I, Year 2022/ 2023
Lab - 3
DML 1 (Select Statement)
DML 2 (Joins)

Part (1) DML1 (Select Statement)

Simple Select Statements

```
SELECT <Column list>
FROM <table names>
[WHERE <Condition>]
```

Example 1:
(using northwind database):

❖ Retrieve all information of all products

```
SELECT *
FROM [dbo].[products]
```

Example 2:

❖ Retrieve Product id and Product name of all products

```
SELECT ProductID, ProductName
FROM products
```

Example 3:

❖ Get all PRODUCTS where their names contain the letter H

```
SELECT *
FROM products
where ProductName like '%h%'
```

Example 4:
(Using DISTINCT Keyword**)**

❖ Get all unique CATOGRIES of the products

```
SELECT DISTINCT CategoryID
FROM [dbo].[products]
```

❖ Get all PRODCUTS where their Category is either 1 or 2

```
SELECT *
 FROM [dbo].[products]
 WHERE categoryid=1 or categoryid= 2
```

```
Another way:
SELECT *
FROM [dbo].[products]
WHERE categoryid IN (1,2)
```

Part (2) DML 2 (Joins)

SQL Joins:

SQL Joins are used to fetch/retrieve data from two or more data tables, based on a join condition. A join condition is a relationship among some columns in the data tables that take part in SQL join. Basically, database tables are related to each other with keys. We use this keys relationship in SQL Joins.

Types of Joins:

- **Inner Join**: returns only those tuples that match the join condition.
- **Outer Join:**
  - **Left outer Join:** every tuple in the left table must appear in the result; if it does not have a matching tuple; it is padded with NULL values for the attributes of the right table.
  - **Right Outer Join:** every tuple in the right table must appear in the result; if it does not have a matching tuple, it is padded with NULL values for the attributes of the left table.
  - **FULL Outer Join:** Keeps all tuples in both the left and the right relations when no matching tuples are found, padding them with NULL values as needed.
- **Cross Join:** Specifies the CARTESIAN PRODUCT operation, it generates all possible tuple combinations.

Join Examples(Using Northwind DB):

Example 1:

Get the names, prices and names of companies that supply these products.

```
SELECT Products.ProductName,Products.UnitPrice,Suppliers.CompanyName
FROM Products inner join  Suppliers
On Products.SupplierID = Suppliers.SupplierID
--OR
SELECT Products.ProductName,Products.UnitPrice,Suppliers.CompanyName
FROM Products, Suppliers
```

```
WHERE Products.SupplierID = Suppliers.SupplierID
```

Example 2:

Get the names of products categories and the name of the company that supply these categories.

```
Select Categories.CategoryName,Suppliers.CompanyName
from Categories inner join Products
on Categories.CategoryID =Products.CategoryID
inner join Suppliers
on Products.SupplierID=Suppliers.SupplierID;

--OR
SELECT Categories.CategoryName, Suppliers.CompanyName
FROM   Categories, Products, Suppliers
WHERE Suppliers.SupplierID = Products.SupplierID
          AND Products.CategoryID = Categories.CategoryID
```

Example 3:

Get names of all the customers and if they place any orders show their order date.

```
SELECT Customers.CompanyName, Orders.OrderDate
FROM Orders RIGHT OUTER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID

OR

SELECT Customers.CompanyName, Orders.OrderDate
FROM Customers LEFT OUTER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

Example 4:

Get names of all products and all categories ,showing which products belong to which categories if any:

```
SELECT Products.ProductName, Categories.CategoryName
FROM   Products FULL OUTER JOIN Categories
ON Products.CategoryID =Categories.CategoryID
```

Note: you need to insert a new Category and a new product to be able to know the changes at the results

- `INSERT INTO Categories (CategoryName) Values('New Category')`

- INSERT INTO Products (ProductName,Discontinued) Values ('New Product',0)

Example 5:

Get all combinations of Products and Suppliers ,Mentioning the names of products and the suppliers.

```
Select Products.ProductName,Suppliers.CompanyName
from Products CROSS JOIN  Suppliers
```