

Name: Abhishek Kanoujia

DATA ENGINEERING BATCH 1

DAY 4 ASSIGNMENT

```
CREATE DATABASE joinexample;
```

```
USE joinexample;
```

```
-- Creating the Student table
```

```
CREATE TABLE Student (  
    id INT,  
    name VARCHAR(50),  
    class INT,  
    city VARCHAR(50)  
);
```

```
-- Inserting values into the Student table
```

```
INSERT INTO Student (id, name, class, city) VALUES (3, 'Hina', 3, 'Delhi');
```

```
INSERT INTO Student (id, name, class, city) VALUES (4, 'Megha', 2, 'Delhi');
```

```
INSERT INTO Student (id, name, class, city) VALUES (6, 'Gouri', 2, 'Delhi');
```

```
-- Creating the Record table
```

```
CREATE TABLE Record (  
    id INT,  
    class INT,  
    city VARCHAR(50)  
);
```

-- Inserting values into the Record table

```
INSERT INTO Record (id, class, city) VALUES (9, 3, 'Delhi');
```

```
INSERT INTO Record (id, class, city) VALUES (10, 2, 'Delhi');
```

```
INSERT INTO Record (id, class, city) VALUES (12, 2, 'Delhi');
```

-- EQUI JOIN using WHERE clause

```
SELECT student.name, student.id, record.class, record.city
```

```
FROM student, record
```

```
WHERE student.city = record.city;
```

	name	id	class	city
►	Gouri	6	3	Delhi
	Megha	4	3	Delhi
	Hina	3	3	Delhi
	Gouri	6	2	Delhi
	Megha	4	2	Delhi
	Hina	3	2	Delhi
	Gouri	6	2	Delhi
	Megha	4	2	Delhi
	Hina	3	2	Delhi

-- Or EQUI JOIN using JOIN...ON

```
SELECT student.name, student.id, record.class, record.city
```

```
FROM student
```

```
JOIN record
```

```
ON student.city = record.city;
```

	name	id	class	city
▶	Gouri	6	3	Delhi
	Megha	4	3	Delhi
	Hina	3	3	Delhi
	Gouri	6	2	Delhi
	Megha	4	2	Delhi
	Hina	3	2	Delhi
	Gouri	6	2	Delhi
	Megha	4	2	Delhi
	Hina	3	2	Delhi

-- NON EQUI JOIN

SELECT student.name, record.id, record.city

FROM student, record

WHERE Student.id < Record.id ;

	name	id	city
▶	Gouri	9	Delhi
	Megha	9	Delhi
	Hina	9	Delhi
	Gouri	10	Delhi
	Megha	10	Delhi
	Hina	10	Delhi
	Gouri	12	Delhi
	Megha	12	Delhi
	Hina	12	Delhi

CREATE DATABASE joinexample2;

USE joinexample2;

-- Create Employee Table

CREATE TABLE Employee (

Emp_id INT PRIMARY KEY,

Emp_name VARCHAR(50),

```

    Dept_id INT
);

-- Insert Data into Employee Table
INSERT INTO Employee (Emp_id, Emp_name, Dept_id) VALUES
(1, 'Ram', 10),
(2, 'Jon', 30),
(3, 'Bob', 50);

-- Create Department Table
CREATE TABLE IF NOT EXISTS Department (
    Dept_id INT PRIMARY KEY,
    Dept_name VARCHAR(50)
);

-- Insert Data into Department Table
INSERT INTO Department (Dept_id, Dept_name) VALUES
(10, 'IT'),
(30, 'HR'),
(40, 'TIS');

-- INNER JOIN
SELECT Employee.Emp_id, Employee.Emp_name, Employee.Dept_id,
Department.Dept_name
FROM Employee
INNER JOIN Department ON Employee.Dept_id = Department.Dept_id;

```

	Emp_id	Emp_name	Dept_id	Dept_name
▶	1	Ram	10	IT
	2	Jon	30	HR

-- LEFT JOIN

```
SELECT Employee.Emp_id, Employee.Emp_name, Employee.Dept_id,
Department.Dept_name
```

```
FROM Employee
```

```
LEFT JOIN Department ON Employee.Dept_id = Department.Dept_id;
```

	Emp_id	Emp_name	Dept_id	Dept_name
▶	1	Ram	10	IT
	2	Jon	30	HR
	3	Bob	50	NULL

-- RIGHT JOIN

```
SELECT Employee.Emp_id, Employee.Emp_name, Employee.Dept_id,
Department.Dept_name
```

```
FROM Employee
```

```
RIGHT JOIN Department ON Employee.Dept_id = Department.Dept_id;
```

	Emp_id	Emp_name	Dept_id	Dept_name
▶	1	Ram	10	IT
	2	Jon	30	HR
	NULL	NULL	NULL	TIS

-- FULL JOIN

```
SELECT Employee.Emp_id, Employee.Emp_name, Employee.Dept_id,
Department.Dept_name
```

```
FROM Employee
```

```
FULL JOIN Department ON Employee.Dept_id = Department.Dept_id
```

```
LIMIT 0, 1000;
```

-- Natural JOIN

SELECT Employee.Emp_id, Employee.Emp_name, Employee.Dept_id,
Department.Dept_name

FROM Employee

NATURAL JOIN Department;

	Emp_id	Emp_name	Dept_id	Dept_name
▶	1	Ram	10	IT
	2	Jon	30	HR

CREATE DATABASE IF NOT EXISTS Northwind;

USE Northwind;

-- Create the Suppliers table

CREATE TABLE IF NOT EXISTS Suppliers (

SupplierID INT PRIMARY KEY,

SupplierName VARCHAR(255),

ContactName VARCHAR(255),

Address VARCHAR(255),

City VARCHAR(255),

PostalCode VARCHAR(10),

Country VARCHAR(255)

);

-- Insert data into the Suppliers table

INSERT INTO Suppliers (SupplierID, SupplierName, ContactName, Address, City, PostalCode, Country)

VALUES

(1, 'Exotic Liquid', 'Charlotte Cooper', '49 Gilbert St.', 'London', 'EC1 4SD', 'UK'),

(2, 'New Orleans Cajun Delights', 'Shelley Burke', 'P.O. Box 78934', 'New Orleans', '70117', 'USA'),

(3, 'Grandma Kelly's Homestead', 'Regina Murphy', '707 Oxford Rd.', 'Ann Arbor', '48104', 'USA'),

(4, 'Tokyo Traders', 'Yoshi Nagase', '9-8 Sekimai Musashino-shi', 'Tokyo', '100', 'Japan');

-- Create the Products table

CREATE TABLE IF NOT EXISTS Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(255),

SupplierID INT,

```
CategoryID INT,  
Unit VARCHAR(255),  
Price DECIMAL(10, 2),  
FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)  
);
```

-- Insert data into the Products table

```
INSERT INTO Products (ProductID, ProductName, SupplierID, CategoryID, Unit, Price)  
VALUES  
(1, 'Chais', 1, 1, '10 boxes x 20 bags', 18),  
(2, 'Chang', 1, 1, '24 - 12 oz bottles', 19),  
(3, 'Aniseed Syrup', 1, 2, '12 - 550 ml bottles', 10),  
(4, 'Chef Anton's Cajun Seasoning', 2, 2, '48 - 6 oz jars', 22),  
(5, 'Chef Anton's Gumbo Mix', 2, 2, '36 boxes', 21.35),  
(6, 'Grandma's Boysenberry Spread', 3, 2, '12 - 8 oz jars', 25),  
(7, 'Uncle Bob's Organic Dried Pears', 3, 7, '12 - 1 lb pkgs.', 30),  
(8, 'Northwoods Cranberry Sauce', 3, 2, '12 - 12 oz jars', 40),  
(9, 'Mishi Kobe Niku', 4, 6, '18 - 500 g pkgs.', 97);
```

-- Create the OrderDetails table

```
CREATE TABLE IF NOT EXISTS OrderDetails (  
    OrderDetailID INT PRIMARY KEY,  
    OrderID INT,  
    ProductID INT,  
    Quantity INT,  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);
```

-- Insert data into the OrderDetails table


```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
```

```
VALUES
```

```
(1, 10248, 1, 12),  
(2, 10248, 2, 10),  
(3, 10248, 3, 5),  
(4, 10249, 4, 9),  
(5, 10249, 5, 40),  
(6, 10250, 6, 10),  
(7, 10250, 5, 35),  
(8, 10250, 7, 15),  
(9, 10251, 8, 6),  
(10, 10251, 9, 15);
```

-- Example using EXISTS

-- List suppliers with a product price less than 20

```
SELECT SupplierName
```

```
FROM Suppliers
```

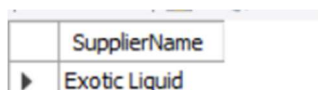
```
WHERE EXISTS (
```

```
    SELECT ProductName
```

```
    FROM Products
```

```
    WHERE Products.SupplierID = Suppliers.SupplierID AND Price < 20
```

```
);
```



	SupplierName
▶	Exotic Liquid

-- Example using EXISTS

-- List suppliers with a product price equal to 22

```
SELECT SupplierName
```

FROM Suppliers

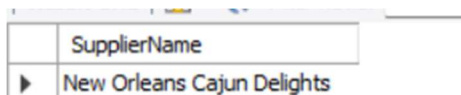
WHERE EXISTS (

SELECT ProductName

FROM Products

WHERE Products.SupplierID = Suppliers.SupplierID AND Price = 22

);



SupplierName
New Orleans Cajun Delights

-- Example using ANY

-- List ProductName if ANY records in OrderDetails have Quantity equal to 10

SELECT ProductName

FROM Products

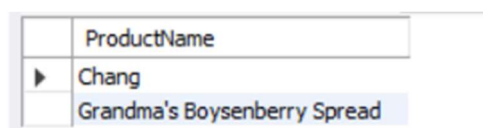
WHERE ProductID = ANY (

SELECT ProductID

FROM OrderDetails

WHERE Quantity = 10

);



ProductName
Chang
Grandma's Boysenberry Spread

-- Example using ANY

-- List ProductName if ANY records in OrderDetails have Quantity larger than 99

SELECT ProductName

FROM Products

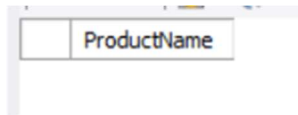
WHERE ProductID = ANY (

SELECT ProductID

FROM OrderDetails

WHERE Quantity > 99

);



ProductName

-- Example using ANY

-- List ProductName if ANY records in OrderDetails have Quantity larger than 1000

SELECT ProductName

FROM Products

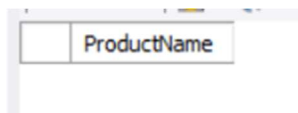
WHERE ProductID = ANY (

SELECT ProductID

FROM OrderDetails

WHERE Quantity > 1000

);



ProductName

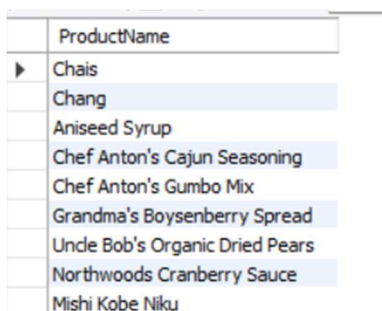
-- Example using ALL

-- List ALL product names

SELECT ALL ProductName

FROM Products

WHERE TRUE;

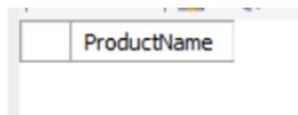


ProductName
Chais
Chang
Aniseed Syrup
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Grandma's Boysenberry Spread
Uncle Bob's Organic Dried Pears
Northwoods Cranberry Sauce
Mishi Kobe Niku

-- Example using ALL

-- List ProductName if ALL records in OrderDetails have Quantity equal to 10

```
SELECT ProductName
FROM Products
WHERE ProductID = ALL (
    SELECT ProductID
    FROM OrderDetails
    WHERE Quantity = 10
);
```



ProductName
