Huy Bui	DATABASE
Information Technology	**Join operation**
T5616SN	February 22, 2018

1 Join operation

A JOIN clause is used to combine rows from two or more tables, based on a related column between them. For example, we can create the following SQL statement (that contains an INNER JOIN), that selects records that have matching values in both tables (CustomerID):

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

There are different types of the JOINs in SQL: (INNER) JOIN, LEFT (OUTER) JOIN, RIGHT (OUTER) JOIN, FULL (OUTER) JOIN

1.1 Inner Join

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match. LEFT JOIN Syntax:

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

1.2 Right Join

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match. Right Join Syntax:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

1.3 Full Join

The FULL OUTER JOIN keyword return all records when there is a match in either left (table1) or right (table2) table records. Full Join Syntax:

```
SELECT column_name(s)
FROM table1
FULL JOIN table2 ON table1.column_name = table2.column_name;
```

2 Practical part

1. List some information of all private owners, along with some information of their properties for rent.

   ```
   SELECT FamilyName,FirstName,TypeOfProperty,
   pro.Street AS TargetStreetAddress
   FROM PrivateOwner pri JOIN PropertyForRent pro
   ON pro.PrivateOwnerId=pri.Id
   ORDER BY FamilyName
   ```

2. Find the number of properties handled by each staff member.

   ```
   SELECT FamilyName,
   count(*) AS "Number of properties"
   FROM PropertyForRent pro JOIN Staff
   ON Staff.Id=pro.StaffId
   GROUP BY FamilyName
   ORDER BY "Number of properties" DESC
   ```

3. Check if each staff member has at least one property for rent to manage using LEFT JOIN.

   ```
   SELECT FirstName,FamilyName,Position,Street,TypeOfProperty
   FROM Staff s LEFT JOIN PropertyForRent p
   ON s.Id=p.StaffId
   ```

4. Check if each property for rent is managed by a staff member using RIGHT JOIN.

   ```
   SELECT FirstName,FamilyName,Street,TypeOfProperty
   FROM Staff s RIGHT JOIN PropertyForRent p
   ON s.Id=p.StaffId
   ORDER BY FirstName
   ```

5. List what is available for rent in the company. TypeOfProperty serves as major sort key and Rent as minor sort key.

   ```
   SELECT TypeOfProperty,Rent,b.Street
   FROM PropertyForRent p JOIN Branch b
   ON p.BranchId=b.Id
   ```

6. List the names of all clients who have viewed a property, along with any comments supplied. The list is organized in alphabetical order of FamilyName.

   ```
   SELECT FamilyName,FirstName,CommentsGiven
   FROM Client c JOIN Viewing v
   ON c.Id=v.ClientId
   ```

7. List the names of all clients who have viewed a property, but exclude those who did not give any comments.

```
SELECT FamilyName,FirstName,CommentsGiven
FROM Client c JOIN Viewing v
ON c.Id=v.ClientId
WHERE CommentsGiven IS NOT NULL
```

8. List the names of all staff and the street address where they work. The list is organized in alphabetical order of FamilyName.

```
SELECT FamilyName,FirstName,b.Street
FROM Staff s LEFT JOIN Branch b
ON s.BranchId=b.Id
ORDER BY FamilyName
```

9. List all owners for the properties for rent and what do they own.

```
SELECT FirstName,FamilyName,pro.Street
FROM PrivateOwner pri  JOIN PropertyForRent pro
ON pri.Id=pro.PrivateOwnerId
```

10. Show what kind of properties each branch offers. Street from Branch table serves as major sort key and column TypeOfProperty as minor sort key.

```
SELECT b.Street,TypeOfProperty,Rent
FROM Branch b JOIN PropertyForRent p
ON b.Id=p.BranchId
ORDER BY b.Street,TypeOfProperty
```