

### 3.4.12 SQL JOIN

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.

Tables in a database are often related to each other with keys.

A primary key is a column (or a combination of columns) with a unique value for each row. Each primary key value must be unique within the table. The purpose is to bind data together, across tables, without repeating all of the data in every table.

Look at the "Persons" table shown in table 7.1:

**Persons table**

| P_Id | LastName   | FirstName | Address    | City      |  |
|------|------------|-----------|------------|-----------|--|
| 1    | Hansen Ola | Timoteivn | 10 Sandnes |           |  |
| 2    | Svendson   | Tove      | Borgvn 23  | Sandnes   |  |
| 3    | Pettersen  | Kari      | Storgt 20  | Stavanger |  |

Note that the "P\_Id" column is the primary key in the "Persons" table. This means that **no** two rows can have the same P\_Id. The P\_Id distinguishes two persons even if they have the same name.

Next, we have the "Orders" table shown in table 7.7:

Table 7.7: Orders Table

Source: <http://www.w3schools.com/>

| O_Id | OrderNo | P_Id |
|------|---------|------|
| 1    | 77895   | 3    |
| 2    | 44678   | 3    |
| 3    | 22456   | 1    |
| 4    | 24562   | 1    |
| 5    | 34764   | 15   |

Note that the "O\_Id" column is the primary key in the "Orders" table and that the "P\_Id" column refers to the persons in the "Persons" table without using their names.

Notice that the relationship between the two tables above is the "P\_Id" column.

The different types of Joins are listed below

- **JOIN:** Return rows when there is at least one match in both tables
- **LEFT JOIN:** Return all rows from the left table, even if there are no matches in the right table

- **RIGHT JOIN:** Return all rows from the right table, even if there are no matches in the left table

#### a. SQL INNER JOIN Keyword

The INNER JOIN keyword return rows when there is at least one match in both tables. The SQL INNER JOIN Syntax is:

```
SELECT column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

Example:

Using tables 7.1 and 7.7 above; if we want to list all the persons with any orders. We use the following SELECT statement:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
INNER JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

The result-set will look like this:

| <b>LastName</b> | <b>FirstName</b> | <b>OrderNo</b> |
|-----------------|------------------|----------------|
| Hansen          | Ola              | 22456          |
| Hansen          | Ola              | 24562          |
| Pettersen       | Kari             | 77895          |
| Pettersen       | Kari             | 44678          |

#### b. SQL LEFT JOIN Keyword

The LEFT JOIN keyword returns all rows from the left table (table\_name1), even if there are no matches in the right table (table\_name2). The SQL LEFT JOIN Syntax is:

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

In some databases LEFT JOIN is called LEFT OUTER JOIN.

Example: If we want to list all the persons and their orders - if any, from the tables 7.3 and 7.4 above. We use the following SELECT statement:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
LEFT JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

The result-set will look like this:

| <b>LastName</b> | <b>FirstName</b> | <b>OrderNo</b> |
|-----------------|------------------|----------------|
| Hansen          | Ola              | 22456          |
| Hansen          | Ola              | 24562          |
| Pettersen       | Kari             | 77895          |
| Pettersen       | Kari             | 44678          |
| Svendson        | Tove             |                |

The LEFT JOIN keyword returns all the rows from the left table (Persons), even if there are no matches in the right table (Orders).

### **c. SQL RIGHT JOIN Keyword**

The RIGHT JOIN keyword Return all rows from the right table (table\_name2), even if there are no matches in the left table (table\_name1). SQL RIGHT JOIN Syntax is:

```
SELECT column_name(s)
FROM table_name1
RIGHT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

Example: Let us list all the orders with containing persons - if any, from the tables 7.3 and 7.4 above. We use the following SELECT statement:

```
SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo
FROM Persons
RIGHT JOIN Orders
ON Persons.P_Id=Orders.P_Id
ORDER BY Persons.LastName
```

The result-set will look like this:

| <b>LastName</b> | <b>FirstName</b> | <b>OrderNo</b> |
|-----------------|------------------|----------------|
|-----------------|------------------|----------------|

|           |      |       |
|-----------|------|-------|
| Hansen    | Ola  | 22456 |
| Hansen    | Ola  | 24562 |
| Pettersen | Kari | 77895 |
| Pettersen | Kari | 44678 |
|           |      | 34764 |

The RIGHT JOIN keyword returns all the rows from the right table (Orders), even if there are no matches in the left table (Persons).

#### d. The SQL UNION Operator

The UNION operator is used to combine the result-set of two or more SELECT statements.

Notice that each SELECT statement within the UNION must have the same number of columns. The columns must also have similar data types. Also, the columns in each SELECT statement must be in the same order. SQL UNION Syntax:

```
SELECT column_name(s) FROM table_name1
UNION
SELECT column_name(s) FROM table_name2
```

Example: Let us consider tables 7.8:

**Table 7.8: Employees table**  
**Source:** <http://www.w3schools.com/>

##### (a) Employees\_Norway:

| E_ID | E_Name            |
|------|-------------------|
| 01   | Hansen, Ola       |
| 02   | Svendson, Tove    |
| 03   | Svendson, Stephen |
| 04   | Pettersen, Kari   |

##### (b) Employees\_USA:

| E_ID | E_Name            |
|------|-------------------|
| 01   | Turner, Sally     |
| 02   | Kent, Clark       |
| 03   | Svendson, Stephen |
| 04   | Scott, Stephen    |

(c) **UNION of tables 7.8 a and b:**

| E_Name            |
|-------------------|
| Hansen, Ola       |
| Svendson, Tove    |
| Svendson, Stephen |
| Pettersen, Kari   |
| Turner, Sally     |
| Kent, Clark       |
| Scott, Stephen    |

If we want to list **all the different** employees in Norway and USA; we use the following SELECT statement:

```
SELECT E_Name FROM Employees_Norway
UNION
SELECT E_Name FROM Employees_USA
```

The result-set will look like this table 7.8c

### Activity C

1. Using tables 7.1 (Person's table) and 7.7 (Order's table), Demonstrate how to execute the following commands:
  - i. Join
  - ii. Left Join
  - iii. Right Join
  - iv. Full Join

## 4.0 Conclusion

When we wish to extract information from a database, we communicate with the Database Management System (DBMS) using a query language called SQL. SQL is the most frequently used programming language in the world, in the sense that every day, more SQL programs are written, compiled and executed than programs in any other computer programming language. SQL is used with *relational* database systems. In a relational database, all of the data is stored in tables.

## 5.0 Summary

In this unit, we have learnt:

- xxix. **Structured Query Language (SQL)** is a database computer language designed for managing data in relational database management systems (RDBMS).
- xxx. SQL has two major parts: Data Definition Language and Data Manipulation Language.
- xxxi. Data Definition Language (DDL) Used to create (define) data structures such as tables, indexes, clusters xxxii. Some of the available DDL commands are: Create, Use, Alter, and Drop
- xxxiii. SQL Constraints are used to limit the type of data that can go into a table. The following constraint types were considered: Not Null, Unique, Primary Key, Foreign Key xxxiv. Data Manipulation Language (DML) is used to manipulate (select, insert, update, delete) data in a Table.
- xxxv. The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.
- xxxvi. **JOIN:** Return rows when there is at least one match in both tables
- xxxvii. **LEFT JOIN:** Return all rows from the left table, even if there are no matches in the right table
- xxxviii. **RIGHT JOIN:** Return all rows from the right table, even if there are no matches in the left table. xxxix. The UNION operator is used to combine the result-set of two or more SELECT statements.

## 6.0 Tutor Marked Assignment

| PFNO | NAMES | STATUS    | HIREDATESALARY |      | COMM | DEPTNO |
|------|-------|-----------|----------------|------|------|--------|
| 1    | AJAYI | CLERK     | 17-Dec-80      | 800  |      | 10     |
| 2    | CHIM  | SALESMAN  | 20-Feb-81      | 1600 | 300  | 40     |
| 3    | JOHN  | MANAGER   | 2-Apr-81       | 1250 |      | 40     |
| 4    | WILL  | SALESMAN  | 28-Sep-81      | 1250 | 300  | 30     |
| 5    | KUDI  | MANAGER   | 1-May-81       | 2975 |      | 30     |
| 6    | TOLA  | MANAGER   | 9-Jun-81       | 2850 |      | 20     |
| 7    | ABDUL | ANALYST   | 27-Jun-90      | 3000 |      | 20     |
| 8    | JAKE  | PRESIDENT | 3-Dec-81       | 5000 |      | 10     |
| 9    | CLERK | SALESMAN  | 31-Jul-90      | 1234 | 500  | 40     |
| 10   | SHEU  | CLERK     | 3-Dec-81       | 1100 |      | 40     |
| 11   | CHIDI | CLERK     | 3-Dec-81       | 950  |      | 20     |
| 12   | HENRY | ANALYST   | 23-Jan-82      | 3000 |      | 20     |
| 13   | IDIA  | CLERK     | 23-Jan-82      | 1200 |      | 30     |
| 14   | KUTI  | SALESMAN  | 23-Jan-82      | 1600 | 600  | 20     |
| 15   | BELLO | CLERK     | 23-Jan-82      | 1250 |      | 10     |

## Employees Table

| DEPTNO | DNAME      | LOCATION |
|--------|------------|----------|
| 10     | ACCOUNTING | IBADAN   |
| 20     | RESEARCH   | LAGOS    |
| 30     | SALES      | MINNA    |
| 40     | OPERATION  | KADUNA   |

## Department Table

Use the above tables to answer the following questions:

Write an SQL statement that:

- i. List all clerks who work in department 20
- ii. List the names of all managers and analysts
- iii. List the names of analysts who are not working in department 20
- iv. List Employees with salary greater than 2500
- v. Which status are paid less than 2000 but more than 1000
- vi. What is the total remuneration of sales people
- vii. List all employees in department 10, order by their salary
- viii. Which employees work in Lagos
- ix. Where does KUDI work?
- x. Which position are paid higher than average salary

## 7.0 Further Reading and other Resources

**David M. Kroenke, David J. Auer** (2008). Database Concepts. New Jersey . Prentice Hall

**Elmasri Navathe** (2003). Fundamentals of Database Systems. England. Addison Wesley.

**Fred R. McFadden, Jeffrey A. Hoffer** (1994). Modern Database management. England. Addison Wesley Longman

**Graeme C. Simsion, Graham C. Witt** (2004). Data Modeling Essentials. San Francisco. Morgan Kaufmann

**Pratt Adamski, Philip J. Pratt** (2007). Concepts of Database Management. United States. Course Technology.