

SQL Joins

SQL Joins

Hittills har vi bara jobbat med enstaka tabeller, men styrkan i Relationsdatabaser är att man kan lagra datan i flera olika tabeller och sedan kombinera dessa efter behov.

Det gör man med sk Joins

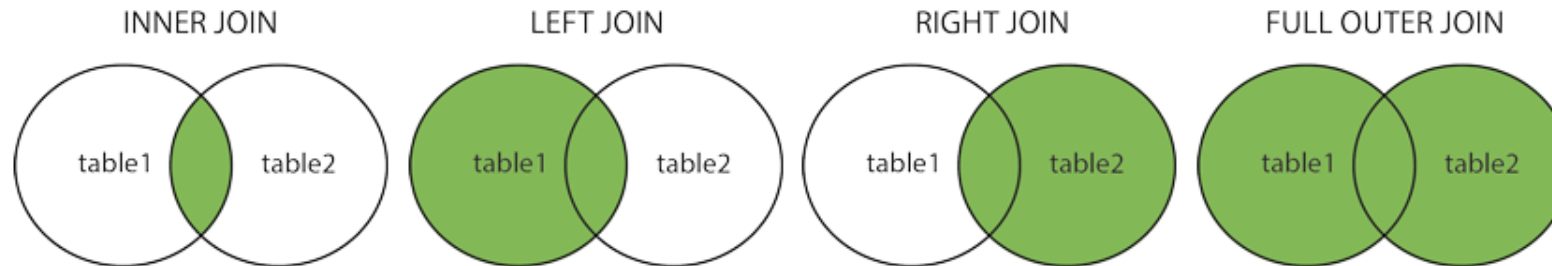
SQL Joins

JOIN används för att kombinera rader från flera tabeller baserat från en relaterad kolumn mellan dem.

Different Types of SQL JOINS

Here are the different types of the JOINS in SQL:

- **(INNER) JOIN** : Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN** : Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN** : Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN** : Returns all records when there is a match in either left or right table



INNER JOIN

En **INNER JOIN** returnerar endast de rader där det finns en match i båda tabellerna.

```
SELECT customers.name, orders.order_date  
FROM customers INNER JOIN orders  
ON customers.customer_id = orders.customer_id;
```

- ▶ Tabellen customers har kundinformation.
- ▶ Tabellen orders har orderinformation.
- ▶ Denna query hämtar namnet på kunden och orderdatumet för de kunder som har gjort en beställning.

LEFT JOIN

En **LEFT JOIN** returnerar alla rader från den vänstra tabellen, samt de matchande raderna från den högra tabellen. Om det inte finns någon match i den högra tabellen blir värdena **NULL**

```
SELECT customers.name, orders.order_date  
FROM customers LEFT JOIN orders  
ON customers.customer_id = orders.customer_id;
```

- ▶ Hämtar alla kunder, oavsett om de har gjort en beställning eller inte.
- ▶ Om en kund inte har gjort en beställning kommer order_date att vara NULL.

RIGHT JOIN

En **RIGHT JOIN** returnerar alla rader från den högra tabellen, samt de matchande raderna från den vänstra tabellen. Om det inte finns någon match i den vänstra tabellen blir värdena **NULL**

```
SELECT customers.name, orders.order_date  
FROM customers RIGHT JOIN orders  
ON customers.customer_id = orders.customer_id;
```

- ▶ Hämtar alla ordrar, även om det inte finns någon motsvarande kund i kundtabellen (t.ex. om en order lades innan kunden lades in i systemet).

FULL JOIN

En **FULL OUTER JOIN** returnerar alla rader från båda tabellerna. Om det inte finns någon match sätts värdena från den tabellen som saknar match till **NULL**

```
SELECT customers.name, orders.order_date  
FROM customers FULL OUTER JOIN orders  
ON customers.customer_id = orders.customer_id;
```

- ▶ Hämtar alla kunder och alla ordrar.
- ▶ Om en kund inte har gjort en beställning visas NULL för orderinformationen.
- ▶ Om en order inte har en motsvarande kund visas NULL för kundinformationen.

Kombinera joins

Vill man använda data från flera tabeller tillsammans så kan man helt enkelt kombinera flera joins.

```
SELECT
```

```
    Orders.OrderID, Customers.CustomerName, Products.ProductName, OrderDetails.Quantity
```

```
FROM
```

```
    Orders
```

```
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID
```

```
INNER JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
```

```
INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID;
```


Kombinera joins

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID

Detta ansluter ordrar med kunder för att få kundens namn kopplat till varje order.

INNER JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

Här länkas orderdetaljer till varje order för att visa vilka produkter som tillhör en specifik order.

INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID

Länkar produkterna till orderdetaljer för att hämta produktnamn och kvantitet.

Exempel

Antag att vi vill hämta orderdetaljer och totalsumma för varje beställd produkt på en specifik order:

SELECT

Orders.OrderID, Customers.CustomerName, Products.ProductName, Products.Price
OrderDetails.Quantity, (Products.Price * OrderDetails.Quantity) AS TotalPerProduct

FROM Orders

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID

INNER JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID

WHERE Orders.OrderID = 101;

Nästlad JOIN

En **nästlad join** innebär att vi använder en join som en del av en större SQL-fråga, vanligtvis genom att använda en subquery.

Detta är användbart när vi behöver kombinera data från flera tabeller där en relation kräver ett mellanled, eller när vi vill filtrera data baserat på resultatet av en annan join.

Nästlad JOIN Exempel

```
SELECT Customers.CustomerName, Products.ProductName, NestedDetails.Quantity
FROM Customers
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID
INNER JOIN (
    SELECT OrderDetails.OrderID, OrderDetails.ProductID, OrderDetails.Quantity
    FROM OrderDetails
    INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID
) AS NestedDetails ON Orders.OrderID = NestedDetails.OrderID
INNER JOIN Products ON NestedDetails.ProductID = Products.ProductID;
```