## 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**: Return all records from the left table, and the matched records from the right table
* **RIGHT (OUTER) JOIN**: Return all records from the right table, and the matched records from the left table
* **FULL (OUTER) JOIN**: Return all records when there is a match in either left or right table
* The COUNT() function returns the number of rows that matches a specified criteria.
* The AVG() function returns the average value of a numeric column.
* The SUM() function returns the total sum of a numeric column.

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;

The MIN() function returns the smallest value of the selected column.

The MAX() function returns the largest value of the selected column.

### Modifizierte Chen-Notation (MC-Notation)

Die **Modifizierte Chen-Notation** (*Modified Chen Notation*, *MC-Notation*) ist eine Erweiterung der Chen-Notation, bei der die Aussage „kein oder ein Element“ mit dem Buchstaben **c** (*choice*, *can*), und die Aussage "ein oder mehr Element(e)" mit dem Buchstaben **m** (*must*, *multiple*) angegeben wird. Daher wird *MC* manchmal auch als *Must-Can* interpretiert.

**1:1** (1 zu 1)

Jede Entität der ersten Entitätsmenge steht mit **genau einer** Entität der zweiten Entitätsmenge in Beziehung, und umgekehrt.

**1:c** (1 zu [0 oder 1])

Jede Entität der ersten Entitätsmenge kann mit **höchstens einer** Entität der zweiten Entitätsmenge in Beziehung stehen. Jede Entität der zweiten Entitätsmenge steht mit **genau einer** Entität der ersten Entitätsmenge in Beziehung.

**1:m** (1 zu [mindestens 1])

Jede Entität der ersten Entitätsmenge steht mit **mindestens einer** Entität der zweiten Entitätsmenge in Beziehung. Jede Entität der zweiten Entitätsmenge steht mit **genau einer** Entität der ersten Entitätsmenge in Beziehung.

**1:mc** (1 zu [beliebig vielen])

Jede Entität der ersten Entitätsmenge kann mit **beliebig vielen** Entitäten der zweiten Entitätsmenge in Beziehung stehen. Jede Entität der zweiten Entitätsmenge steht mit **genau einer** Entität der ersten Entitätsmenge in Beziehung.

**c:c** ([1 oder 0] zu [0 oder 1]; entspricht *1:1* in Chen-Notation)

Jede Entität der ersten Entitätsmenge kann mit **höchstens einer** Entität der zweiten Entitätsmenge in Beziehung stehen, und umgekehrt.

**c:m** ([0 oder 1] zu [mindestens 1])

Jede Entität der ersten Entitätsmenge steht mit **mindestens einer** Entität der zweiten Entitätsmenge in Beziehung. Jede Entität der zweiten Entitätsmenge kann mit **höchstens einer** Entität der ersten Entitätsmenge in Beziehung stehen.

**c:mc** ([0 oder 1] zu [beliebig vielen]; entspricht *1:n* in Chen-Notation)

Jede Entität der ersten Entitätsmenge kann mit **beliebig vielen** Entitäten der zweiten Entitätsmenge in Beziehung stehen. Jede Entität der zweiten Entitätsmenge kann mit **höchstens einer** Entität der ersten Entitätsmenge in Beziehung stehen.

**m:m** ([mindestens 1] zu [mindestens 1])

Jede Entität der ersten Entitätsmenge steht mit **mindestens einer** Entität der zweiten Entitätsmenge in Beziehung, und umgekehrt.

**m:mc** ([mindestens 1] zu [beliebig vielen])

Jede Entität der ersten Entitätsmenge kann mit **beliebig vielen** Entitäten der zweiten Entitätsmenge in Beziehung stehen. Jede Entität der zweiten Entitätsmenge steht mit **mindestens einer** Entität der ersten Entitätsmenge in Beziehung.

**mc:mc** ([beliebig viele] zu [beliebig vielen]; entspricht *m:n* in Chen-Notation)

Jede Entität der ersten Entitätsmenge kann mit **beliebig vielen** Entitäten der zweiten Entitätsmenge in Beziehung stehen, und umgekehrt

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18)  
);

CREATE VIEW VerPerInfo (Name, PersonalID, Departmentname)

AS SELECT Mitarbeiter.Name, Mitarbeiter.PersonalID, Mitarbeiter.Departmentname

FROM Mitarbeiter, Verwaltungspersonal WHERE Mitarbeiter.PersonalID = Verwaltungspersonal.PersonalID;