

Beispiel:

In der Tabelle Fahrradeder soll die Anzahl der verschiedenen Hersteller gezählt werden.

Lösung in Standard-SQL:

```
SELECT COUNT(DISTINCT 'Hersteller') AS 'Anzahl Hersteller'
FROM 'fahrraeder'
```

Ergebnis:

Anzahl Hersteller
13

Das Schlüsselwort DISTINCT unterdrückt die doppelte Zählung der einzelnen Hersteller.

Hinweis:

Jet-SQL (Access) lässt die direkte Kombination von COUNT() und DISTINCT nicht zu. Deshalb muss man sich mit einer geschachtelten Lösung behelfen:

```
SELECT COUNT(*) AS 'Anzahl Hersteller'
FROM
(
SELECT DISTINCT Hersteller FROM Fahrradeder
);
```

Mit SELECT DISTINCT ... werden zunächst alle Namen der Hersteller ohne Duplikate gesucht. Die übergeordnete COUNT-Funktion zählt anschließend die Anzahl der Ergebnisse dieser Unterabfrage.

Hinweise:

Um die gesamte Anzahl aller Datensätze einer Tabelle zu erfassen, ist die Suche mit COUNT(*) zu verwenden.

Die Funktionen SUM() und AVG() können nur bei numerischen Spalteninhalten angewendet werden.

Die Funktionen MIN() und MAX() können auch bei Text- und Datumsfeldern verwendet werden, wobei Textfelder nach dem ASCII-Code behandelt werden: A ist „kleiner“ als Z und Kleinbuchstaben sind „größer“ als Großbuchstaben, allerdings „kleiner“ als Sonderzeichen. Ein späteres Datum ist „größer“ als ein früheres.

Beispiel:

Die folgende SQL-Anweisung ermittelt das Anschaffungsdatum des neuesten Fahrrades.

```
SELECT MAX(Anschaffungsdatum)
FROM Fahrradeder;
```

Rechenoperationen

Mit den Werten aus numerischen Spalten können Rechenoperationen wie z. B. Addition, Subtraktion, Multiplikation und Division durchgeführt werden.

Dies gilt auch für Datumswerte, da diese als Integerzahlen (ganze Zahlen) abgelegt sind.

Datumsfunktionen

Mit Datumsfunktionen lassen sich Datumswerte verarbeiten. Z. B. kann die Jahreszahl aus einem Datum herausgefiltert werden oder das aktuelle Datum dynamisch verarbeitet werden.

Beispiel:

Um alle Fahrräder aufzulisten, deren Wartungsdatum länger als 100 Tage zurückliegt, ist die folgende Anweisung notwendig:

```
SELECT Fahrradnummer, Bezeichnung, LetzteWartung
FROM Fahrradeder
WHERE LetzteWartung < DATE()-100;
```

Die Funktion DATE() liefert das Systemdatum. Von diesem Wert wird die Zahl 100 subtrahiert und das Ergebnis jeweils mit dem Inhalt des Feldes LetzteWartung verglichen. Alle Datensätze mit einem kleineren Wert werden ausgegeben.

Hinweis:

Datumsfunktionen gehören nicht zum SQL-Standard, sind jedoch in nahezu allen Datenbanksystemen vorhanden. Sie unterscheiden sich jedoch zum Teil erheblich. Sie sind sehr vom verwendeten Datenbanksystem abhängig. Gegebenenfalls ist im jeweiligen Handbuch bzw. der Hilfefunktion nachzuschlagen.

In Access gibt es neben der Funktion DATE(), die das aktuelle Datum liefert, weitere Datumsfunktionen:

Funktion	Ergebnis
DATE()	Aktuelles Datum
DAY(datum)	Tag vom Feldinhalt 'datum' als Zahl zwischen 1 und 31
MONTH(datum)	Monat als Zahl zwischen 1 und 12
YEAR(datum)	Jahr als vierstellige Zahl
ISDATE(datum)	Prüft, ob Datumsangabe korrekt ist
FORMAT(ausdruck, 'format')	Formatiert das Datum in ausdruck nach Vorgabe: D,DD: Tageszahl, DDD, DDDD: Wochentag, M, MM: Monatszahl, MMM, MMMM: Monatsname YY, YYYY: Jahreszahl

Mit der Anweisung FORMAT(ausdruck, 'format') können z. B. Datumswerte formatiert ausgegeben werden.

Beispiel:

Das Rückgabedatum der Felder der Tabelle Verleihdaten soll in der Form „Freitag, 26. April 2019“ ausgegeben und umbenannt werden.

```
SELECT FORMAT(Rückgabedatum, 'DDDD, DD. MMMM YYYY') AS Rechnungsdatum
FROM Verleihdaten;
```

Ergebnis:

Rechnungsdatum
Dienstag, 06. Februar 2018
Mittwoch, 04. Oktober 2019
Dienstag, 12. März 2021
Samstag, 27. April 2021