

Beispiel:

Die folgende SQL-Anweisung zeigt die Verleihpreise der einzelnen Fahrräder in der Tabelle Preisgruppen um 5% erhöht an, wobei sie auf volle €-Beträge aufgerundet werden.

```
SELECT CINT(Preis*1.05)
FROM Preisgruppen;
```

Mathematische und logische Funktionen

Um mathematische Aufgaben zu lösen, z. B. die Quadratwurzel eines Feldinhaltes zu bilden, werden folgende Funktionen verwendet:

Funktion	Ergebnis
ABS (zahl)	Absoluter Wert (ohne negatives Vorzeichen)
SQR (zahl)	Wurzel einer Zahl: SQR(16) = 4
RND ()	Zufallszahl zwischen 0 und 1
ISNULL (ausdruck)	Wert = 1, wenn z. B. Feldinhalt leer ist, Wert = 0, wenn der Feldinhalt nicht leer ist.
EXP (Wert)	Ergibt e ^{Wert} mit der Eulerschen Zahl e = 2.71828... als Basis.
LOG (Wert)	Der natürliche Logarithmus, d. h. LN(EXP(Wert)) = Wert.
LOG10 (Wert)	Logarithmus von der Zahl Wert zur Basis 10
SIGN (Wert)	Liefert -1 für negative Werte und 1 für positive Werte
MOD (Wert, Divisor)	Restwert der (ganzzahligen) Division von Wert/Divisor

Beispiel:

Aus einer Tabelle mit Leitungsquerschnitten werden die Durchmesser errechnet.

Lösung:

```
SELECT Flaeche, 2*SQR(Flaeche/(3.1415*2)) AS Durchmesser
FROM Leitung ;
```

Die Funktion RND () erzeugt eine Zufallszahl als Gleitkommazahl >0 und <1, z. B. 0,3452419.
Die Funktion ISNULL (ausdruck) prüft, ob der Ausdruck, z. B. ein Feldinhalt, leer ist. Trifft dies zu, so liefert die Funktion den Wert 1 ('wahr'), sonst den Wert 0 ('falsch').

6.3.10 Gruppieren von Daten

Gruppieren bedeutet, die Daten nach Feldern mit gleichen Feldinhalten zusammenzufassen. Dies geschieht mit dem SQL-Ausdruck GROUP BY (= zusammenfassen aufgrund) gefolgt von einem Feldnamen.

Beispiel:

Die folgende Anweisung stellt das Einzugsgebiet des Unternehmens „Faradiso“ fest, indem der Wohnort und die Anzahl der Kunden aus den einzelnen Orten ausgegeben werden:

```
SELECT Ort, COUNT(KdNr) AS Anzahl
FROM Kunden
GROUP BY Ort;
```

Ort	Anzahl
Berlin	2
Biberach	2
Jettingen-Scheppach	2
Düsseldorf	35
Neckarsulm	1
Haan-Gruiten	12
Ulm	11
Zwickau	1
...	...

Zunächst wird durch den Ausdruck GROUP BY Ort aus allen Datensätzen, die den gleichen Eintrag im Feld Ort besitzen, jeweils eine Gruppe gebildet. Anschließend werden innerhalb jeder dieser Gruppen aufgrund der Funktion COUNT (Kd_Nr) die Anzahl der Datensätze anhand der Kd_Nr festgestellt.

Hinweis:

Alle Felder, die nach dem Schlüsselwort SELECT zusätzlich zur Aggregatfunktion aufgelistet werden, müssen auch hinter der Gruppierungsanweisung GROUP BY angegeben werden, da es sonst zu einer Fehlermeldung kommt.

Bedingungen für Gruppierungen – HAVING:

Die Anzeige der einzelnen Gruppen kann durch die Anweisung HAVING Bedingung zusätzlich von Bedingungen abhängig gemacht werden.

Beispiel:

```
SELECT Hersteller, SUM(Anschaffungspreis) AS Summe
FROM Fahrradeder
GROUP BY Hersteller
HAVING SUM(Anschaffungspreis) > 4000;
```

Hersteller	Summe
Hirsch	5.932,00 €
Panasonic	6.095,00 €

Die Anweisung HAVING SUM(Anschaffungspreis) >4000 bewirkt z. B., dass nur die Gruppen ausgegeben werden, deren aufsummierter Anschaffungspreis („Summe“) mehr als 4000 € beträgt.

Hinweis:

Die Bedingung nach dem Schlüsselwort HAVING muss immer auf die gesamte Gruppe anwendbar sein.

6.3.11 Abfragen über mehrere Tabellen (JOINS)

Joins (to join = verknüpfen) stellen logische Verbindungen zwischen mehreren Tabellen her. Um z.B. alle Verleihdaten mit den dazugehörigen Kundennamen auszugeben, müssen Felder der beiden Tabellen Kunden und Reservierungen angezeigt werden. Die Anweisung lautet dann:

```
SELECT.KdNr, Kunden.Nachname, Kunden.Vorname,
Reservierungen.AusleihDatum
FROM Kunden, Reservierungen;
```