

SAE Stoffsammlung

Gliederung

1	Normalformen	2
1.1	1. Normalform	2
1.2	2. Normalform	2
1.3	3. Normalform	2
2	SQL Befehle	2
2.1	Tabelle erstellen	2
2.2	Werte eintragen	3
2.3	Bestehende Werte ändern	3
2.4	Werte aus Tabelle löschen	3
2.5	Komplette Tabelle löschen	3
2.6	Alle Werte aus Tabelle löschen	3
2.7	Werte anzeigen	3
2.7.1	Alle anzeigen	3
2.7.2	Nur bestimmte Spalten anzeigen	3
2.7.3	Spalten Beschriftungen	3
2.7.4	Zeilen zählen	4
2.7.5	Werte aufsummieren	4

1 Normalformen

1.1 1. Normalform

Jedes Attribut der Relation muss einen atomaren Wertebereich haben, und die Relation muss frei von Wiederholungsgruppen sein.

Atomar bedeutet, dass ein Attribut nur eine Information enthalten darf, also dürfen bspw. Straße und Hausnummer nicht in einem Attribut gespeichert werden, sondern müssen aufgeteilt werden.

Mit **Wiederholungsgruppen** sind zum einen wiederholende Werte in einer Spalte gemeint (also bspw. in der Spalte Ort mehrmals der Wert Kirchheim) und zum anderen sich wiederholende Spalten (bspw. Spalten Standort1, Standort2, Standort3).

Durch diese Normalform werden Abfragen der Datenbank überhaupt erst möglich gemacht. Ohne atomare Daten könnte nicht jeder Aspekt der Daten einzeln abgefragt werden.

1.2 2. Normalform

1. Normalform muss erfüllt sein. Ein Nichtprimärattribut darf nicht funktional von einer Teilmenge eines Schlüsselkandidaten abhängen.

Das bedeutet: Attribute die nur von einem Teil eines Schlüssels und nicht von allen Schlüsseln in einer Tabelle (bei zusammengesetzten Schlüsseln) abhängig sind müssen ausgelagert werden, damit sie voll abhängig von ihrem Schlüssel werden.

Durch diese Normalform modelliert jede Relation nur einen Sachverhalt.

1.3 3. Normalform

2. Normalform muss erfüllt sein. Kein Nichtschlüsselattribut darf transitiv von einem Schlüsselkandidaten abhängig sein.

Das bedeutet: Wenn aus einem Attribut ein anderes Attribut ersichtlich wird, welches aber nicht durch den Schlüssel ersichtlich wird, ist es transitiv von dem ersten Attribut anhängig.

	CD_ID	Albumtitel	Interpret	Gründungsjahr
Bsp.:	1	Not That Kind	Anastacia	1999
	2	Wish you were here	Pink Floyd	1965
	3	Freak of Nature	Anastacia	1999

Das Gründungsjahr ist durch den Interpreten ersichtlich und nicht durch die CD_ID. Das heißt, Interpret und Gründungsjahr sollten in eine Tabelle ausgelagert werden.

2 SQL Befehle

2.1 Tabelle erstellen

```
CREATE TABLE tabellename
(id INT NOT NULL AUTO_INCREMENT,
wert VARCHAR(255) NOT NULL, PRIMARY (id));
```

2.2 Werte eintragen

```
INSERT INTO tabellenname (id, wert)
VALUES (1, 'Test');
```

2.3 Bestehende Werte ändern

```
UPDATE tabellenname
SET wert = 'Update_TEST'
WHERE id = 1;
```

2.4 Werte aus Tabelle löschen

```
DELETE FROM tabellenname
WHERE id = 1;
```

2.5 Komplette Tabelle löschen

```
DROP tabellenname;
```

2.6 Alle Werte aus Tabelle löschen

```
TRUNCATE tabellenname;
```

2.7 Werte anzeigen

2.7.1 Alle anzeigen

```
SELECT * FROM tabellenname;
```

2.7.2 Nur bestimmte Spalten anzeigen

```
SELECT werte FROM tabellenname;
```

2.7.3 Spalten Beschriftungen

```
SELECT werte AS "Text"
FROM tabellenname;
```

2.7.4 Zeilen zählen

Bei 3 Einträgen in Werte:

```
SELECT COUNT(werte) AS "Anzahl_Texte"  
FROM tabellenname;
```

Ergebnis: Anzahl Texte 3

2.7.5 Werte aufsummieren

Angenommen die Tabelle enthält in der Spalte anzahl die Werte: 1, 3, 3, 7

```
SELECT SUM(anzahl) AS "Summe_Anzahl"  
FROM tabellenname;
```

Ergebnis wäre dann: Summe Anzahl 14

2.7.6 Weitere Rechenfunktionen

Außer COUNT() und SUM() gibt es noch:

- AVG: Average, Durchschnitt über Werte bilden
- MIN: Den kleinsten Wert ausgeben
- MAX: Den größten Wert ausgeben
- ROUND(spalte, dezimalstellen): Gerundete Werte ausgeben

2.7.7 Werte miteinander verrechnen

Angenommen die Tabelle enthält in der Spalte anzahl die Werte: 4, 2 und in der Spalte preis die Werte 5, 4.

```
SELECT (anzahl * preis) AS "Preis_gesamt"  
FROM tabellenname;
```

Die Ergebnisse wären dann: 20, 8