

SQL CheatSheet

SQL: Structured Query Language

DDL: Data Definition Language

→ CREATE, ALTER, RENAME; DROP; TRUNCATE

DML: Data Manipulation Language

→ INSERT; UPDATE; DELETE, SELECT

DCL: Data Control Language

→ GRANT, REVOKE

TCL: Transactional Control Language

→ COMMIT; ROLLBACK

DATENBANK ANLEGEN:

CREATE DATABASE datenbank;

DATENBANK WECHSELN:

USE datenbank;

DATENBANK LÖSCHEN:

DROP datenbank;

TABELLEN ANLEGEN:

CREATE TABLE tabellenname

{
spaltenname,

PRIMARY KEY (spalten)

UNIQUE (spalten)

FOREIGN KEY (spalten)

REFERENCES ON tabelle(spalte)

[**ON DELETE** action]

[**ON UPDATE** action],

CHECK (bedingung)

};

SPALTENDEFINITION:

spaltenname **DATENTYP**

[**NOT NULL** |

NULL |

DEFAULT wert |

CHECK (bedingung) |

PRIMARY KEY |

UNIQUE |

REFERENCES tabl(spalte) [**ON**

UPDATE | **ON DELETE**];

DATENTYPEN:

SMALLINT	2 Byte (-32.000 – 32.000)
-----------------	---------------------------

INTEGER	4 Byte (4.2 Mrd)
----------------	------------------

FLOAT (7Stellen)	4 Byte
-------------------------	--------

DOUBLE PRECISION (15 Stellen)	8 Byte
---	--------

NUMERIC (Präzision, Skalierung)

DECIMAL (Präzision, Skalierung)

DECIMAL(8,3)

DATE	8 Byte
-------------	--------

	tt.mm.jj
--	----------

	tt-mm-jj
--	----------

	mm-tt-jj
--	----------

CHAR (Länge)	Max 255
---------------------	---------

	=tat. Speicher- verbrauch
--	------------------------------

VARCHAR (Länge)	Länge + 2 Byte
------------------------	----------------

BLOB	Binärdaten
-------------	------------

TEXT	Große Texte
-------------	-------------

TABELLE LÖSCHEN:

DROP TABLE tabellenname;

INTEGRITÄTSREGELN:

Primärschlüssel (PRIMARY KEY):

Jeder Datensatz kann eindeutig angesprochen werden

Fremdschlüssel (FOREIGN KEY):

Dient der Verknüpfung von Tabellen im relationalen Modell

UNIQUE:

Verhindert doppelte Werte

DEFAULT:

Standardwert für ein Feld

REFERENZIELLE INTEGRITÄT

FOREIGN KEY (id) **REFERENCES** tabelle(spalte)

[**ON UPDATE** | **ON DELETE**]

NO ACTION:

Ausführung der Anwendung wird abgebrochen

CASCADE:

Löschen: alle DS in anderen Tabellen, die auf diesen DS verweisen, werden gelöscht!

Update: alle angewiesenen DS werden ebenfalls aktualisiert

SET DEFAULT:

Alle referenzierten DS werden auf DEFAULT gesetzt (CREATE)

INDEX DEFINIEREN

CREATE INDEX indexname

ON tabl (spaltenliste);

Bsp.: **CREATE INDEX** idc_name **ON** kunde (name)

INDEX LÖSCHEN

DROP INDEX indexname

INTEGRITÄTSREGELN:

Indices werden nur auf Felder angelegt, in denen häufig gesucht wird!

Tabellen, die vor allem zur Speicherung von Informationen dienen und nicht abgefragt werden, benötigen in der Regel keinen Index (Logs)

Die Anlage von Indices lohnt nur bei einer großen Datenmenge in den jeweiligen Feldern.

Der Datentyp darf nicht TEXT/BLOB/NULL-Werte sein/zulassen

Indices sollten nur bei Feldern angewandt werden, in denen insbesondere einzelne Datensätze gesucht werden.

Zusammengesetzte Indices sollten dann eingesetzt werden, wenn häufige Abfragen über die jeweils gleichen Spalten durchgeführt werden.

TABELLEN VEREINIGEN

SELECT spalte

FROM tabelle

UNION

SELECT spalte

FROM tabelle

DATENSÄTZE ABFRAGEN:

DELETE FROM tabelle

[**WHERE** auswahlbedingung]

SQL CheatSheet

ER-MODELL: ENTITY RELATIONSHIP MODEL ENTITÄT: Eigenständige Einheit im Datenmodell (z. B. Produkt) ENTITÄTSTYP: Zusammenfassung von Entitäten der gleichen Klasse ATTRIBUT: Eigenschaft einer Entität BEZIEHUNG: Entitäten können in Beziehung gesetzt werden, um deren Verhalten genauer zu beschreiben. NORMALISIERUNG Als Normalisierung bezeichnet man die Anwendung von Regeln zur Optimierung des Datenmodells und zur Eliminierung von Fehlern. Die Normalisierung erfolgt in mehreren Schritten: 1. Normalform Eine Relation befindet sich in der 1. Normalform, wenn keine Spalte mit gleichem Inhalt vorliegt (keine Wiederholungen) und Daten in einer Tabelle keine untergeordnete Relation bilden. Die Tabelle muss ein Schlüsselattribut besitzen, der eine Zeile einer Tabelle eindeutig identifiziert. 2. Normalform 1. Normalform muss erfüllt sein. Jedes Nicht-Schlüsselattribut ist vom Primärschlüssel funktional abhängig. 3. Normalform 2. Normalform muss erfüllt sein. Zwischen den Nicht-Schlüsselattributen der Tabelle bestehen keine Abhängigkeiten.	DATENSÄTZE EINFÜGEN: INSERT INTO tabellenname (spaltenliste) VALUES (werte); Mehrere Values (), (), (), ... DATENSÄTZE AKTUALISIEREN: UPDATE tabellenname SET spaltenname = wert [, spaltenname = wert] [WHERE bedingung] DATENSÄTZE LÖSCHEN: DELETE FROM tabelle [WHERE auswahlbedingung] UNTERABFRAGEN Unterabfragen ermöglichen die Verwendung eines Abfrageergebnisses unmittelbar in einer Anweisung: SELECT spalte FROM tabelle WHERE spaltenname VERGLEICH (SELECT abfrage) VIEW ANLEGEN Transaktionen: COMMIT ROLLBACK	DATENSÄTZE SELEKTIEREN SELECT [DISTINCT ¹] spaltenliste FROM tablname [as aliasname] [{INNER LEFT RIGHT} JOIN] Tabellenname aliasname [ON tabellenname.spaltenname = Tabelename1.spaltenname1] [WHERE auswahlbedingung] [GROUP BY spaltenname] [HAVING auswahlbedingung_group] [ORDER BY spaltenliste] SELECT name, vname FROM kontakte WHERE ort="München" ORDER BY name DATENSÄTZE LÖSCHEN: Gruppieren. GROUP BY Bsp: GROUP BY ort, plz Sortieren: SORT by spalte Bsp: SORT BY name [ASC DESC] VIEW ANLEGEN CREATE VIEW viewname (spalten) AS auswahlbedingungen [WITH CHECK OPTION]	AUSWAHLBEDINGUNGEN LIMIT: limitiert die Anzeige der Datensätze (zahl,zahl) WHERE: <ul style="list-style-type: none">- Vergleich: Preis < 100 preis = „Maier“- Bereichsprüfung: Preis BETWEEN 10 AND 100- Elementprüfung abt IN („Einkauf“, „Verkauf“)- Mustervergleich name LIKE „M%“- Nullwertprüfung preis IS NULL- Logische Operatoren <, >, =, >=, <=, <>- % = kein oder mehrere Zeichen- _ exakt ein Zeichen- AND OR NOT-Verknüpfungen FUNKTIONEN <ul style="list-style-type: none">- Count- Avg- Min- Max- Sum Mathematisch <ul style="list-style-type: none">- Abs- Ceiling- Floor- Round- Log- mod
--	---	---	--

¹ Vermeidung doppelter Datensätze

SQL CheatSheet
