SQL

SQL = Structured Query Language

DDL (DATA DEFINATION LANGUAGE):

* CREATE
* DROP
* ALTER

DML (DATA MANIPULATION LANGUAGE):

* INSERT
* DELETE
* UPDATE
* (SELECT)

DCL (DATA CONTROL LANGUAGE):

* GRANT (zum Rechte hinzufügen)
* REVOKE (zum Rechte entfernen)

DQL (DATA QUERY LANGUAGE):

* SELECT

SQL ≠ SELECT

* SOURCE

Ein Programm, dass gestartet wird ist ein Process



Hierarchisches Model:



Mit der Angabe PRIMARY KEY kann ein Datenfeld als Primärschlüssel festgelegt werden. Der Primärschlüssel ermöglicht es, einen Datensatz eindeutig zu identifizieren.

ALTER:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

<SQL_2021.pdf>

<SQL_2021_Schnelluebersichten.pdf>

Like Operator:

\* 🡪 % // beliebig viele Zeichen können vorkommen

? 🡪 \_ // Mindestens ein Zeichen muss vorkommen

WHERE name LIKE ‘S\_\_u\_t%‘

LIKE ‘%inp%’

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Es gibt in SQL fertige Funktionen

Funktionen:

* COUNT()
* MAX()
* MIN()
* SUM()
* AVG() //Average 🡪 Der Durchschnitt

HAVING ist eine Bedingung

z.B.:

SELECT ort AS Wohnort, COUNT(name) FROM t\_ma GROUP BY ort HAVING COUNT(name) >= 10;

DDL… Data definition Language

DCL… Data Control Language

DML… Data Manipulation Language

http://172.16.1.77/insy/sql/

CONSTRAINTS

Typen:

* PRIMARY KEY
* UNIQUE KEY
* FOREIGN KEY
* CHECK
* NOT NULL / DEFAULT

Index = braucht man zum Suchen (Baumstruktur)

Ohne Index muss man alles durchsuchen

Man macht den Index auf die Spalten, die sehr oft durchsuchen/verwendet werden

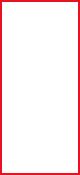
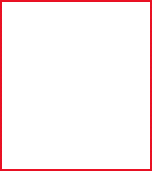
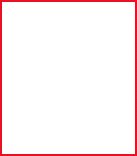
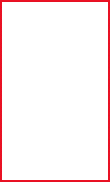
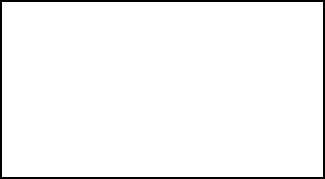
z.B.:

CREATE INDEX ix\_produkte\_bezeichnung ON wawi.produkte(bezeichnung);

CREATE INDEX

DROP INDEX

VIEWS:



z.B.:

CREATE VIEW wawi.garten

USER:

CREATE USER v2b IDENTIFIED BY 'v2b';

Kommentare:

-- Einzeiliger Kommentar

/\*

Mehrzeiliger Kommentar

\*/

Schema Name = Datenbankname

-- Ausgabe der aktuellen Systemzeit

SELECT NOW();

**WICHTIG FÜR DEN TEST**

zum Zusammenhängen von Texten wird die Funktion CONCAT() verwendet.

SELECT persnr, nachname, vorname, akadgrad,

**CONCAT**(akadgrad,' ',vorname,' ',UPPER(nachname)) AS "Mitarbeiter(in)"

FROM wawi.personal;

Wenn es keinen Akademischen Titel gibt:

**CONCAT**(IFNULL(akadgrad, '')

Zeit dazu/weg rechnen:

SELECT datum,

DATE\_ADD(datum, INTERVAL 1 DAY) AS plus\_ein\_tag,

DATE\_ADD(datum, INTERVAL 3 WEEK) AS plus\_drei\_wochen,

DATE\_ADD(datum, INTERVAL -2 MONTH) AS minus\_zwei\_monate,

DATE\_ADD(datum, INTERVAL 1 QUARTER) AS plus\_ein\_quartal,

DATE\_ADD(datum, INTERVAL 5 YEAR) AS plus\_5\_jahre

FROM wawi.bestellungen;

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

IN:

IN vergleich in einer Liste

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

SELECT nachname, vorname, akadgrad

FROM wawi.personal

WHERE akadgrad IS NULL;

-- die folgende Möglichkeit wird in ANSI-SQL nicht unterstützt

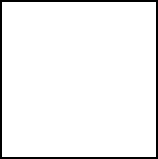
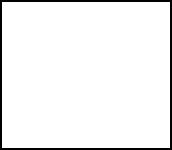
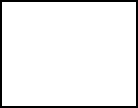
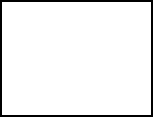
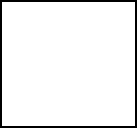
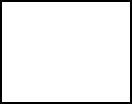
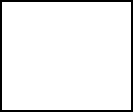
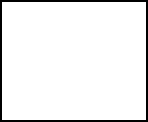
-- WHERE akadgrad = NULL;

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungCOUNT()** zählt **null** nicht mit



c bedeutet 0 oder 1

SELECT Name,Vorname,Abteilung FROM TabA,TabB;

JOIN

Cros join == ,

Equi join == auf gleichheit

Natural

SELECT VN,NN FROM Person WHERE Gehalt > Select AVG(Gehalt) FROM Person;

1 (1,1)



c (0,1)



m (1,\*)



mc (0,\*)



Diskussion

ER-Diagram:

* Design

Relationenmodell:

* Design mit Fremdschlüssel
* PK und FK hinzufügen
* Bei einer N zu M Verbindung muss man eine Zwischentabelle hinzufügen
* Zwischentabelle
  + ist eine Kombination beider Namen
  + Mindestens zwei Attribute
  + Die Kombination von beiden ist der PK

Alle Verbindungen nochmal so aufschreiben

Relationen

SQL-Script