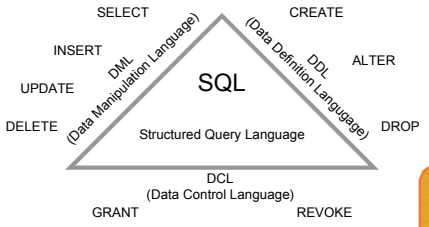
SQL (Structured Query Language) – Kurzreferenz

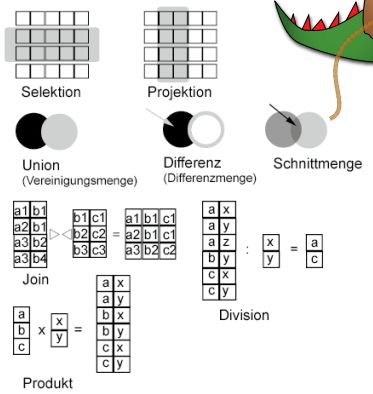
SQL-Definition



Structured Query Language (strukturierte Abfragesprache): Sprache für die Definition von DB-Objekten, Abfrage von Datensätzen und Zugriffskontrolle. Man kann die Befehle in drei Klassen einteilen: DCL (Data Control Language), DML (Data Manipulation Language) und DDL (Data Definition Language).

Relationale Algebra

- Die **Selektion** stellt eine der beiden Grundformen für die häufigste Abfrage bereit, nämlich die horizontale Auswahl von Zeilen einer Tabelle. Wichtig ist hierbei, dass die Anzahl der Spalten des eingehenden Schemas und des ausgehendes Schemas (also der Tabellen) gleich bleibt. Die Auswahl erfolgt dabei über eine Bedingung.
- Eine **Projektion** stellt eine Auswahl der Spalten aus einer Tabelle dar. Die Kardinalität des Schemas bleibt also nicht erhalten. Die Ergebnis- oder die Abfragetabelle wird mindestens eine Spalte weniger als die Ursprungstabelle enthalten.
- Der **Union** stellt eine Vereinigungsmenge aus zwei Tabellen her. Dabei gilt, dass die beiden eingehenden Tabellen das gleiche Schema, also die gleiche Spaltenstruktur, haben müssen.
- Die **Differenz** stellt das logische Gegenstück zur Vereinigung dar. Wiederum müssen die eingehenden Tabellen das gleiche Schema besitzen. Die Operation T1 – T2 der Differenzbildung verläuft dann so, dass nur die Datensätze in der Ergebnistabelle übrig bleiben, die zur Tabelle T1, aber nicht zur Tabelle T2 gehören.
- Die **Schnittmenge** ist insoweit mit der Differenzmenge verwandt, als dass ebenfalls zwei Tabellen mit gleichem Schema aufgrund ihrer Datensätze miteinander verglichen werden. Dabei bleiben in der Ergebnistabelle die Datensätze übrig, die in beiden Tabellen vorhanden sind. Die Schnittmenge ist nicht wirklich eine Basisoperation, sondern setzt sich zusammen aus zwei Differenzen: $T1 \cap T2 = T1 - (T1 - T2) = T2 - (T2 - T1)$
- Die **Division** stellt eine Operation dar, die auf zwei eingehende Tabellen angewandt werden kann, die einen Teil ihrer Daten gleich haben. Dabei wird eine Ergebnistabelle erzeugt, die Werte von T1 enthält, die nicht zu T2 gehören, aber als zugehörige Werte die von T1 enthalten.
- Das **kartesische Produkt** aus zwei eingehenden Tabellen ergibt eine Verknüpfung beider Wertemengen, wobei sämtliche Kombinationsmöglichkeiten gebildet werden.



Standard-Aggregatfunktionen

Funktion	Bedeutung
AVG	Durchschnitt der Spalte
MIN	Minimum der Spalte
MAX	Maximum der Spalte
SUM	Summe der Spalte
COUNT	Anzahl der Einträge einer Spalte

Sicherheit

Rechte

Rechte und Berechtigungen sind erlaubte SQL-Anweisungen (Operationen) in der Datenbank, die wiederum über SQL-Befehle Rollen oder Benutzern zugewiesen werden.

Benutzer

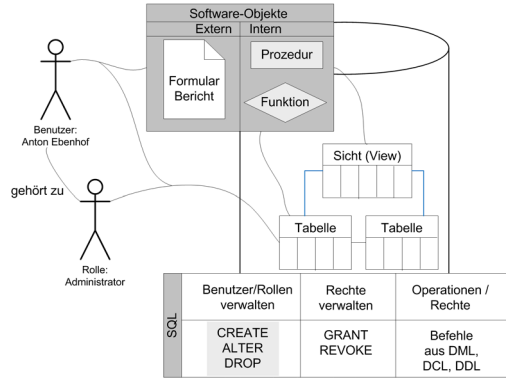
Benutzer sind natürliche oder fiktive Personen, die Zugriff auf die Daten oder die DB-Struktur erhalten.

Rollen

Rollen sind Bündel von Rechtestrukturen, die wiederum Benutzern zugewiesen werden können.

Programm-Objekte

Objekte wie Sichten (Views), Prozeduren und Funktionen in der Datenbank und Formulare und Berichte aus externen Anwendungen beschränken die Nutzung auf bestimmte Teile der Datenbank.



DDL – Data Definition Language

Datenstrukturen

Datenbank anlegen und löschen

```
CREATE DATABASE dbname
DROP DATABASE dbname
```

Tabellen anlegen, ändern und löschen

```
CREATE TABLE tabellenname
  ({spaltenname {domäne | datentyp}
      [spalteneinschränkung]
      [DEFAULT standardwert]
  | tabelleneinschränkung
   }, ... )

PRIMARY KEY [name] (spaltenname)
FOREIGN KEY [name] (spaltenname)
   REFERENCES (tabellenname.spaltenname)

ALTER TABLE tabellenname
   {ADD spaltenname datentyp}
  |{ALTER | CHANGE spaltenname_alt spaltenname_neu
datentyp [(laenge)]
      |{DEFAULT wert | DROP DEFAULT}
  |{DROP spaltenname {RESTRICT | CASCADE}
  |{RENAME tabellenname_neu}
  |{ADD tabelleneinschränkung}
  |{DROP CONSTRAINT einschränkung {RESTRICT | CASCADE}}

DROP TABLE [IF EXISTS] tabellenname
```

Die beiden Schlüsselwörter `RESTRICT` und `CASCADE` modifizieren die Löschaktion, wenn weitere Objekte wie Sichten (views) und Einschränkungen für Tabellen oder weitere Objekte (assertion) vorhanden sind. In diesen Fällen verhindert `RESTRICT` das Löschen, um die Funktionsweise der anderen Objekte nicht zu beeinflussen, während `CASCADE` alle Objekte löscht.

Einschränkung

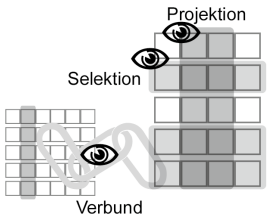
```
CREATE CONSTRAINT einschränkungsname
   bedingung
   [[INITIALLY DEFERRED | INITALLY IMMEDIATE]
    [NOT] DEFERRABLE]
DROP CONSTRAINT einschränkungsname {CASCADE | RESTRICT}
```

Domäne

```
CREATE DOMAIN domänenname [AS] datentyp
   [DEFAULT voreinstellung]
   [[einschränkung]]
      CHECK (bedingung)
         [[INITIALLY DEFERRED | INITIALLY IMMEDIATE]
         [[NOT] DEFERRABLE]
         ]
   ]
   [COLLATE collationname]
```

Sichten

Eine Sicht ist eine virtuelle Tabelle, die eine Abfrage in Form des Relationenschemas speichert und dieses bei Aufrufen ausführt. Ihr Aufruf erfolgt wie die Abfrage einer realen Tabelle und kann auch weitere Einschränkungen enthalten. Aktualisierungsanfragen können auch über Sichten realisiert werden.




```
CREATE VIEW sichtname
[(spalte1, spalte2,...)]
   AS abfrage
   [WITH [CASCADED | LOCAL]
    CHECK OPTION]
```

Indizes

```
CREATE [UNIQUE] INDEX
indexname
  ON tabellenname
  {(spalte1[, spalte2,…])}
```

SQL (Structured Query Language) – Kurzreferenz

DDL – Data Definition Language

Sonstiges

Synonyme / Alias-Namen

```
CREATE {SYNONYM | ALIAS} synonymname FOR tabellenname
DROP {SYNONYM | ALIAS} synonymname
```

Kommentare

```
COMMENT ON {[TABLE | COLUMN]} DB-objektname
   IS kommentar
```

Sicherheit

Benutzer anlegen und löschen

```
{ALTER | CREATE} USER benutzer
   IDENTIFIED {BY kennwort | EXTERNALLY}
      {[DEFAULT TABLESPACE tabellenbereich]
      |[TEMPORARY TABLESPACE tabellenbereich]
      [QUOTA {Integer [K|M] | UNLIMITED} ON tabellenbereich]}
      [PROFILE profil]
      [DEFAULT ROLE { rolle [, rolle] ...
       | ALL [EXCEPT rolle [, rolle] ...] | NONE}]

DROP USER benutzer [CASCADE]
```

Rolle anlegen und löschen

```
{ALTER | CREATE} ROLE rollenname
   [NOT IDENTIFIED |{IDENTIFIED BY kennwort | EXTERNALLY}]

DROP ROLE rollenname
```

DCL – Data Control Language

Rechte vergeben

```
GRANT {ALL PRIVILEGES
       | SELECT
       | DELETE
       | {INSERT [(spalte1 [, spalte2,...])}
       | {UPDATE [(spalte1 [, spalte2,...])}
       | {REFERENCES [(spalte1 [, spalte2,...])}
        }[, ...]
      ON [TABLE] tabellenname
         TO {benutzer1 [, benutzer2, ...]} | PUBLIC
      [WITH GRANT OPTION]
```

Rechte entziehen

```
REVOKE [GRANT OPTION FOR]
   recht [, ...] ON tabelle
   FROM {benutzer1 [, benutzer2, ...]} | PUBLIC
      {CASCADE | RESTRICT}
```

DML – Data Manipulation Language

Daten abrufen

Grundstruktur der Abfrage

```
SELECT [ALL | DISTINCT]
       * | {spalte1 [, spalte2, ...]}
       FROM tabellennamen
      [WHERE suchbedingung]
      [ORDER BY [spalte1 [, spalte2, ...]] [ASC
| DESC]]
      [GROUP BY [spalte1 [, spalte2, ...]] [ASC
| DESC]]
      [HAVING gruppensuchbedingung]
```

Filter und Bedingungen

Mit der `WHERE`-Klausel können Eigenschaften von Daten für DB-Abfragen und Aktualisierungen vorgegeben werden. Verschiedene Operatoren für Vergleiche (<, = oder LIKE , IN), Verknüpfungen (`AND`, `OR`) oder Berechnungen (+, /) stehen zur Verfügung.

(Un-)Gleichheit	`=, !=`
Vergleich	`<, >, <=, >=`
Boolesche Operatoren	`AND, OR, NOT`
Mathematische Operatoren	`+, -, /, *`
Abfrage auf `NULL`	`IS [NOT] NULL`
Abfrage einer Werteliste (Ersatz von mit `OR` verbundenen Einzeltests)	`[NOT] IN (wert1, wert2, …)`
Abfrage eines Bereichs (inkl. Grenzen)	`[NOT] BETWEEN zahl1 AND zahl2`
Unscharfe Suche mit Platzhaltern für ein beliebiges Zeichen (Unterstrich) und mehrere beliebige Zeichen (Prozentzeichen)	`[NOT] LIKE 'zk'`

CASE-Anweisung

```
CASE
   WHEN bedingung1 THEN ergebnis1
   WHEN bedingung2 THEN ergebnis2
   [ELSE ergebnis3]
END
```

Prädikate für Unterabfragen

```
WHERE ... vergleichsoperator [ANY | ALL | SOME]
unterabfrage
WHERE ... vergleichsoperator [UNIQUE]
unterabfrage
WHERE ... vergleichsoperator [[NOT] EXISTS]
unterabfrage
```

Verknüpfungen

- `INNER JOIN` bezieht nur solche Zeilen in die erste Ergebnisrelation ein, die über die Primärschlüssel-Fremdschlüssel-Beziehung verknüpft sind.
- `LEFT OUTER JOIN` verknüpft die Zeilen der linken Tabelle mit den zugehörigen Werten der rechten Tabelle, wobei evtl. nicht vorhandene Werte in der zweiten Tabelle NULL gesetzt werden.
- `RIGHT OUTER JOIN` entspricht von der Funktionsweise dem `LEFT OUTER JOIN`.
- `FULL OUTER JOIN` kombiniert beide vorher beschriebenen Verknüpfungsarten.
- `CROSS JOIN` richtet das kartesische Produkt der beiden eingehenden Tabellen ein.
- `NATURAL JOIN` verknüpft eingehende Tabellenspalten aufgrund ihres Namens.

Mengen-Operatoren für Abfragen

- `UNION` vereinigt die Ergebnismengen mit Duplikat-Entfernung.
- `UNION ALL` vereinigt die Ergebnismengen und behält Duplikate bei.
- `EXCEPT` (auch: `MINUS`) liefert die aus Abfrage A, die nicht in Abfrage B enthalten sind ohne Duplikate.
- `INTERSECT` liefert die Schnittmenge zweier Ergebnismengen ohne Duplikate.

Daten einfügen

```
INSERT INTO tabellenname
    [(spalte1 [, spalte2, ...])]
    {VALUES
   (1wert1 [, 1wert2, ...)}
  |{[VALUES]
   (2wert1 [, 2wert2, ...)}
  |SELECT-Anweisung
```

Daten aktualisieren

```
UPDATE tabellenname
   SET spalte1 = {wert | NULL | DEFAULT}
   [, spalte2 = {wert | NULL | DEFAULT}...]
   [WHERE suchbedingung
       [SELECT-Anweisung]
   ]
Daten löschen
DELETE FROM tabellename
   |WHERE suchbedingung
   |WHERE suchbedingung IN ( SELECT-Anweisung )
```

Transaktionen

Eine Transaktion sind zusammenhängende Handlungen innerhalb der Datenbank, die den DB-Inhalt von einem konsistenten in einen anderen konsistenten Zustand überführen.

Die Eigenschaften von Transaktionen beschreibt man mit dem Akronym `ACID`.

- Unteilbarkeit (`Atomicity`): Eine Transaktion umfasst einzelne oder mehrere Operationen in Form von SQL-Befehlen, die entweder komplett oder gar nicht ausgeführt werden.
- Einheitlichkeit (`Consistency`): Die einzelnen Operationen einer Transaktionen müssen in ihrer Gesamtheit eine Datenbank von einem konsistenten Zustand in einen anderen konsistenten Zustand überführen.
- Isolation (`Isolation`): Im Mehrbenutzerbetrieb werden verschiedene Transaktionen isoliert voneinander ausgeführt. Dadurch sind die Änderungen der einzelnen Transaktionen bzw. auch die Änderungen der einzelnen Operationen erst nach Abschluss der kompletten Transaktion für alle anderen Benutzer sichtbar.
- Dauerhaftigkeit (`Durability`): Sobald eine Transaktion abgeschlossen ist, bleiben ihre Änderungen dauerhaft in der Datenbank erhalten und dürfen nicht erneut durch Fehler beeinträchtigt werden oder verloren gehen.

`COMMIT` schreibt die DB-Aktionen fest und setzt sie damit für alle Benutzer, die zeitgleich mit der Datenbank arbeiten, sichtbar um und setzt die Datenbank in einen neuen konsistenten Zustand.

`ROLLBACK` setzt die DB-Aktionen wieder zurück und damit die Datenbank wieder in einen vorherigen konsistenten Zustand.

Zusammengestellt von Marco Skulschus
Layout und Satz: Nadine Kilian

Comelio GmbH
Goethestr. 34, 13086 Berlin
Web: www.comelio.com

Terrashop GmbH
Lise-Meitner-Str. 8, 53332 Bornheim
Web: www.terrashop.de

ISBN 978-3-939701-79-8