

S(tructured)Q(uey)L(anguage) Aktionsabfragen

Lösche alle Kunden, die
im letzten Jahr kein Ware
bestellt haben.

Der Preis für Waren der
Kategorie
„Süßwaren“ erhöhen sich
um 10 %.

Neue Waren werden in den
Bestand aufgenommen.

Aktionsabfragen

- Aktionen wie einfügen, bearbeiten oder löschen werden immer nur auf einer Tabelle ausgeführt.
- Nach Ausführung der Aktion können die Änderungen nicht automatisiert rückgängig gemacht werden.
- Vor großen Änderungen sollte eine Kopie der Tabelle angelegt werden.

Möglichkeiten

- Aktualisierung von allen oder gefilterten Datensätzen.
Änderung von Werten in Datenfeldern.
- Einfügung von neuen Datensätzen.
- Löschung von allen oder gefilterten Datensätzen.

Umbenennung von Tabellen

```
ALTER TABLE "Employee" RENAME TO "Employee_old";
```

- Mit Hilfe des Befehls `ALTER TABLE` kann eine Tabelle verändert werden. Der Name der zu verändernden Tabelle folgt dem Befehl.
- `RENAME TO` nennt die zu ändernde Tabelle um. Die neue Bezeichnung folgt dem Befehl.

Kopie der Tabellenstruktur

```
CREATE TABLE "Employee" AS
  SELECT *
  FROM "Employee_old"
  WHERE "HireDate" IS NOT NULL;
```

- `CREATE TABLE` erzeugt eine neue Tabelle mit dem Namen `Employee`.
- Die Tabelle wird durch den Befehl `AS` in Bezug auf eine Auswahlabfrage erstellt.
- In Abhängigkeit der Tabellenstruktur in der Auswahlabfrage wird die Struktur der neuen Tabelle definiert.

... und Daten kopieren

```
CREATE TABLE "Employee" AS
  SELECT *
  FROM "Employee_old"
  WHERE "HireDate" IS NOT NULL;
```

- `CREATE TABLE` erzeugt eine neue Tabelle mit dem Namen `Employee`.
- Die Tabelle wird durch den Befehl `AS` in Bezug auf eine Auswahlabfrage erstellt.
- Die Auswahlabfrage legt fest, welche Datensätze aus der Datenquelle kopiert werden sollen.

Namen von Tabellen und Feldern

- In Auswahlabfragen müssen die Namen so wie in der Datenbank angegeben geschrieben werden.
- Bezeichner, die nicht durch Anführungszeichen begrenzt werden, beachten nicht die Groß- und Kleinschreibung.
- Bezeichner, die die Groß- und Kleinschreibung nutzen, müssen durch die Anführungszeichen begrenzt werden.

Kopie eines Ausschnitts einer Tabelle

```
CREATE TABLE artistwithoutalbum AS
SELECT "Name"
FROM "Artist"
WHERE ("Artist"."ArtistId" IN (
    SELECT "Artist"."ArtistId "
    FROM "Artist"
    LEFT OUTER JOIN "Album"
    ON ("Artist"."ArtistId" = "Album"."ArtistId")
    WHERE ("Album"."Title" IS NULL)
)
);
```

Erläuterung

- `CREATE TABLE tabelle AS.` Erzeuge die Tabelle `tabelle` als.
- Die Datenfelder der neuen Tabelle werden entsprechend der Feldliste ("`Name`") der Auswahlabfrage definiert. Die Feldliste folgt dem Befehl `SELECT`.
- Die Datensätze aus der Datenquelle "`Artist`" werden entsprechend der Ergebnistabelle der Auswahlabfrage kopiert. In diesem Beispiel wird die Ergebnistabelle entsprechend der Bedingung (`WHERE ...`) eingeschränkt.

... und Daten kopieren

```
CREATE TABLE "Employee" AS
  SELECT *
  FROM "Employee_old"
  WHERE "HireDate" IS NOT NULL;
```

- `CREATE TABLE` erzeugt eine neue Tabelle mit dem Namen `Employee`.
- Die Tabelle wird durch den Befehl `AS` in Bezug auf eine Auswahlabfrage erstellt.
- Die Auswahlabfrage legt fest, welche Datensätze aus der Datenquelle kopiert werden sollen.

Aktualisierung von allen Datensätzen

```
UPDATE "Employee"  
SET  
    "Title" = 'Sales Agent',  
    "Salary" = 2000.00;
```

- Die Abfrage beginnt mit dem Schlüsselwort `UPDATE`.
- Dem Schlüsselwort folgt die zu ändernde Tabelle. In diesem Beispiel `Employee`.
- Dem Schlüsselwort `SET` folgt eine Liste von Zuweisungen.

Liste von Zuweisungen

```
UPDATE "Employee"  
SET  
    "Title" = 'Sales Agent',  
    "Salary" = 2000.00;
```

- Die Zuweisungen werden in der Liste durch ein Komma getrennt.
- Dem Schlüsselwort `SET` folgt mindestens eine Zuweisung.

Neuberechnung von Attribut-Werten

feldname	=	ausdruck
Title	=	'Sales Agent'
PaidDate	=	DateTime('now')
Salery	=	Salery + ((Salery * 10) / 100)

Zuweisungen

```
UPDATE "Employee"  
SET  
    "Title" = 'Sales Agent',  
    "Salery" = 2000.00;
```

- Das Gleichheitszeichen kann mit „wird zugewiesen“ übersetzt werden.
- Links vom Gleichheitszeichen wird das Datenfeld angegeben, welches verändert werden soll.
- Rechts vom Gleichheitszeichen steht ein Wert. Mit Hilfe dieses Wertes wird der gespeicherte Wert überschrieben. Der Wert kann durch Funktion berechnet werden.

Neuer Wert

```
UPDATE "Employee"  
SET  
    "Title" = 'Sales Agent',  
    "Salary" = 2000.00;
```

- Der neue Wert wird in Abhängigkeit des Datentyps des zu ändernden Datenfeldes angegeben.
- Der Wert kann durch eine Funktion berechnet werden.

Beispiele für Zahlen

```
FeldGleitkommazahl = 1.99,
```

```
FeldGanzzahl = 0
```

- Als Dezimaltrennzeichen wird der Punkt genutzt.

Beispiele für Text

```
FeldText = 'D'
```

- Text wird durch ein Apostroph am Anfang und Ende begrenzt.
- Ein Text besteht aus alphanumerische und numerische Zeichen.

Beispiele für Datums- und Zeitangaben

```
FeldTimeStamp = '2009-01-01 00:00:00',  
FeldDate = '2009-01-01',  
FeldTime = '00:00:00',
```

- Datums und Zeitangaben werden durch ein Apostroph am Anfang und Ende begrenzt.
- Datumsangaben werden durch das Format yyyy-mm-dd angegeben.
- Zeitangaben werden in dem Format hh:mm:ss angegeben.

... von gefilterten Datensätzen

```
UPDATE "Employee"  
SET  
    "Title" = 'Sales Agent',  
    "Salery" = 2000.00  
WHERE ("Title" = 'Sales Support Agent');
```

- Mit Hilfe von `WHERE` können die Datensätze in Abhängigkeit der Bedingung eingeschränkt werden.
- Wenn die Bedingung auf keine Datensätze in der Tabelle zu treffen, werden keine Datensätze verändert.
- Die Bedingung kann mit Hilfe einer Auswahlabfrage getestet werden.

Berechnung von Werten

```
UPDATE "Employee"  
SET "Salery" = "Salery" + (("Salery" * 10) / 100)  
WHERE ("Title" = 'Sales Agent');
```

```
UPDATE "Track"  
SET "UnitPrice" = "UnitPrice" +  
                (("UnitPrice" * 10) / 100)  
WHERE ("Milliseconds" > 3000000);
```

Mathematische Operatoren

Berechnung	Operator
Addition	$[Wert] + [Wert]$
Subtraktion	$[Wert] - [Wert]$
Multiplikation	$[Wert] * [Wert]$
Division	$[Wert] / [Wert]$
Potenz	$[Wert] ^ [Wert]$

- Weitere Operatoren:

<https://www.postgresql.org/docs/9.5/static/functions-math.html>

Nutzung von Funktionen

```
UPDATE "Invoice"  
SET "InvoiceYear" = date_part('year',  
"InvoiceDate");
```

- Der Rückgabewert der Funktion wird als Standardwert genutzt.
- Die implementierten Funktionen sind abhängig vom verwendeten Datenbanksystem.
- Funktionen, die in PostgreSQL implementiert sind, werden auf der Seite <https://www.postgresql.org/docs/current/static/functions.html> aufgelistet.

Test der Aktualisierungsabfrage

```
-- zu testende Abfrage
UPDATE "Employee"
SET
    "Title" = 'Sales Agent',
    "Salery" = 2000.00
WHERE ("Title" = 'Sales Support Agent');
```

```
-- wird getestet mit
SELECT
    "Title",
    "Salery"
FROM "Employee"
WHERE ("Title" = 'Sales Support Agent');
```

Einfügung von Datensätzen

```
INSERT INTO "Employee"  
("LastName", "FirstName", "HireDate", "Address",  
"City", "PostalCode")  
  
VALUES  
('Smith', 'Kate', Current_Date, 'Fisher Street 12',  
'Calgary', 'T3B 1Y7')
```

... in die Tabelle

```
INSERT INTO "Employee"  
("LastName", "FirstName", "HireDate", "Address",  
"City", "PostalCode")
```

- Die Anweisung beginnt mit dem Befehl `INSERT INTO`.
- Dem Befehl folgt ein Tabellennamen.
- In diese Tabelle werden die neuen Datensätze am Ende eingefügt.

... in die Datenfelder

```
INSERT INTO "Employee"  
("LastName", "FirstName", "HireDate", "Address",  
"City", "PostalCode")
```

- Dem Tabellennamen folgt in den runden Klammern eine Liste von Feldnamen.
- Die Feldnamen werden durch Kommata getrennt.
- Die Liste kann leer sein.

... mit Hilfe einer Wertliste

```
VALUES  
( 'Smith', 'Kate', Current_Date, 'Fisher Street 12',  
  'Calgary', 'T3B 1Y7' )
```

- Dem Schlüsselwort `VALUES` folgt eine Liste von Werten.
- Die Liste wird durch die runden Klammern begrenzt. Die Elemente in der Liste werden durch ein Komma getrennt.
- In diesem Beispiel werden Literale vom Datentyp Text genutzt und ein Wert durch eine Funktion berechnet.

Zuordnung

```
INSERT INTO employees  
(  
  "LastName",  
  "FirstName",  
  "HireDate",  
  "Address",  
  "City",  
  "PostalCode")
```

```
VALUES  
(  
  'Smith',  
  'Kate',  
  DateTime('now'),  
  'Fisher Street 12',  
  'Calgary',  
  'T3B 1Y7')
```

Hinweise

- Die neuen Werte werden von links nach rechts den Felder in der Liste zugeordnet. Die Feldliste und die Wertliste haben die gleiche Anzahl von Elementen.
- Falls die Feldliste leer ist, werden die Werte den Datenfelder in der Tabelle von links nach rechts zugeordnet.

Regeln

- Die Werte können entsprechend des Datentyps des zugeordneten Datenfeldes interpretiert werden.
- Für den Primärschlüssel in der Tabelle muss ein eindeutiger, nicht vorhandener Wert angegeben werden. Ausnahme: Der Schlüsselwert wird automatisch berechnet.
- Für alle Felder, die keine NULL-Werte (`NOT NULL`) erlauben, muss ein Wert angegeben werden. Ausnahme: Ein Standardwert ist für das Datenfeld gesetzt.
- Felder, die mit `UNIQUE` gekennzeichnet sind, erzeugen ein Fehler, falls der einzufügende Wert vorhanden ist.

Berechnung von Werten

```
INSERT INTO "Artist"  
VALUES (  
    (SELECT Max("ArtistId") + 1 FROM "Artist"),  
    'Magtens Korridor'  
);
```

- Die Auswahlabfrage gibt exakt einen Wert zurück.
- In diesem Beispiel wird mit Hilfe der Aggregatfunktion `Max()` der größte Wert in dem Datenfeld `ArtistId` in der Tabelle `Artist` berechnet. Dieser Wert wird inkrementiert. Aggregatfunktionen fassen Werte von mehreren Datensätzen zusammen.

Berechnung von Werten

```
INSERT INTO "Artist"  
VALUES (  
    (SELECT Max("ArtistId") + 1 FROM "Artist"),  
    'Magtens Korridor'  
);
```

- Die Auswahlabfrage gibt exakt einen Wert zurück.
- In diesem Beispiel wird mit Hilfe der Aggregatfunktion `Max()` der größte Wert in dem Datenfeld `ArtistId` in der Tabelle `Artist` berechnet. Dieser Wert wird inkrementiert. Aggregatfunktionen fassen Werte von mehreren Datensätzen zusammen.

Kopieren von Datensätzen

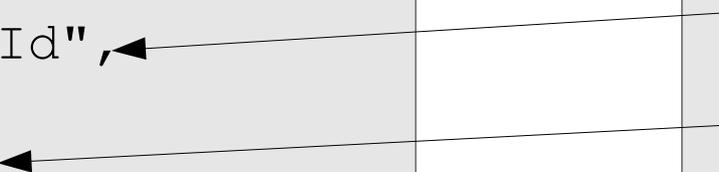
```
INSERT INTO "Artist" ("ArtistId", "Name")  
SELECT "id", "artistname" FROM "artistNew";
```

- Der Befehl `INSERT INTO` legt fest, wo die Daten eingefügt werden sollen.
- Mit Hilfe der Auswahlabfrage werden die zu kopierenden Daten gewählt.

Zuordnung der Felder

```
INSERT INTO "Artist"  
(  
  "ArtistId",  
  "Name"  
)
```

```
SELECT  
  "id",  
  "artistname"  
FROM "artistNew"
```



Hinweise

- Die Felder werden von links nach rechts zugeordnet.
- Die Anzahl der Felder in der Feldliste der `INSERT INTO`- und `SELECT`-Anweisung müssen gleich sein.

Kopieren von Datensätzen

```
COPY (SELECT * FROM artistwithoutalbum)
TO 'D:/tmp/artistwithoutalbum.csv'
WITH CSV DELIMITER ',';
```

- Mit Hilfe des Befehls `COPY` werden Datensätze in oder von einer Datei kopiert.
- Für den Kopiervorgang können Optionen angegeben werden.

Optionen

```
COPY (SELECT * FROM artistwithoutalbum)
TO 'D:/tmp/artistwithoutalbum.csv'
WITH CSV DELIMITER ',';
```

- Die Daten werden in diesem Beispiel in dem Format CSV kopiert.
- In diesem Beispiel wird ein Komma als Separator zwischen den verschiedenen Einträgen genutzt.
- Weitere Optionen werden im Web <https://www.postgresql.org/docs/9.2/static/sql-copy.html> erläutert.

Kopieren in eine Datei

```
COPY (SELECT * FROM artistwithoutalbum)
TO 'D:/tmp/artistwithoutalbum.csv'
WITH CSV DELIMITER ',';
```

- Dem Befehl `COPY` folgt in runden Klammern die Datenquelle. Mit Hilfe einer Auswahlabfrage werden die zu kopierenden Daten definiert.
- Dem Schlüsselwort `TO` folgt ein absoluter Pfad zu einer Datei. Falls die angegebene Datei nicht vorhanden ist, wird diese angelegt.

Absoluter Pfad

- Der Speicherort einer Datei wird von der Wurzel her beschrieben.
- Der Pfad beginnt mit dem Laufwerksbuchstaben. Dem Laufwerksbuchstaben folgt ein Doppelpunkt.
- Als Pfadtrennzeichen wird der Schrägstrich genutzt.
- Hinweis: Schreibrechte für den Speicherort müssen vorhanden sein. Andernfalls wird ein Fehler angezeigt.

Kopieren in eine Tabelle

```
COPY artistwithoutalbum("Name" )  
FROM 'D:/tmp/artistwithoutalbum.csv'  
WITH CSV DELIMITER ',';
```

- Dem Befehl `COPY` folgt das Datenziel. Die Tabelle, in die die Datensätze aus der Datenquelle kopiert werden soll, folgt. In den runden Klammern werden die Datenfelder angegeben. Die Anzahl der Datenfelder in den runden Klammern entspricht der Anzahl der Spalten in der zu kopierenden Datei.
- Dem Schlüsselwort `FROM` folgt die Angabe der Datenquelle. Die Datenquelle wird mit Hilfe eines absoluten Pfades definiert.

Löschen von allen Datensätzen

```
DELETE FROM "Track";
```

- Die Aktionsabfrage beginnt mit dem Schlüsselwort `DELETE`.
- Dem Schlüsselwort `FROM` folgt der Tabellename aus dem die Datensätze gelöscht werden.
- In diesem Beispiel werden alle Datensätze ohne Rückfrage aus der Tabelle gelöscht.

Löschen von mehreren Datensätzen

```
DELETE FROM "Track"  
WHERE "Track"."GenreId" = 3;
```

- Die Aktionsabfrage beginnt mit dem Schlüsselwort `DELETE`.
- Dem Schlüsselwort `FROM` folgt der Tabellename aus dem die Datensätze gelöscht werden.
- Mit Hilfe der `WHERE`-Klausel werden die Datensätze ausgewählt, die gelöscht werden. Alle Datensätze, die der Bedingung entsprechen, werden gelöscht.

Andere Möglichkeit

```
DELETE FROM "Artist"  
  
WHERE ("Artist"."ArtistId" IN (  
    SELECT "Artist"."ArtistId "  
    FROM "Artist"  
    LEFT OUTER JOIN "Album"  
    ON ("Artist"."ArtistId" = "Album"."ArtistId")  
    WHERE ("Album"."Title" IS NULL)  
    )  
);
```

Löschung in Abhängigkeit von SELECT

```
SELECT "Artist"."ArtistId "  
FROM "Artist"  
LEFT OUTER JOIN "Album"  
ON ("Artist"."ArtistId" = "Album"."ArtistId")  
WHERE ("Album"."Title" IS NULL)
```

- In der Ergebnistabelle wird der Primärschlüssel der Tabelle `Artist` angezeigt.
- Es werden alle Schlüsselwert aus der Tabelle `Artist` angezeigt, egal ob der Schlüssel als Detailschlüssel in der Tabelle `Album` genutzt wird oder nicht.
- Das Ergebnis wird mit Hilfe von `WHERE` auf Schlüsselwerte eingeschränkt, die nicht in der Tabelle `Album` vorkommen.

Bedingung zur Löschung

```
WHERE ("Artist"."ArtistId" IN (  
    SELECT "Artist"."ArtistId "  
    FROM "Artist"  
    LEFT OUTER JOIN "Album"  
    ON ("Artist"."ArtistId" = "Album"."ArtistId")  
    WHERE ("Album"."Title" IS NULL)  
    )  
    )
```

- Die Löschanfrage hat wiederum eine Bedingung. Es werden alle Datensätze gelöscht, deren Schlüsselwert in der Liste vorkommen.
- Die Werte werden mit Hilfe der Auswahlabfrage ermittelt.