

Datenbanken

Prof. Dr. Markus Döhring,
Augustinus Nicander Hery, Bela Bauss

markus.doehring@h-da.de

Praktikum 3

Lernziele

Im Rahmen dieses Praktikums sollen Sie die folgenden Ziele erreichen:

1. Editieren von SQL-Skripten
2. Foreign Key Constraints und referentielle Integrität verstehen und anpassen
3. cascade Option bei Foreign Key Constraints verstehen und nutzen

1 Daten löschen und wiederholt einfügen

1. Schreiben Sie am Anfang Ihres Insert-Skriptes Befehle, die alle alten Daten zunächst aus den Tabellen löschen. Der Befehl hierzu lautet `delete from table TABELLENNAME;`. Überlegen Sie, ob die Reihenfolge eine Rolle spielt. Falls ja, in welcher Reihenfolge sollten die Tabelleninhalte gelöscht werden?

```
DELETE FROM ausstrahlung;  
DELETE FROM bietet_an;  
DELETE FROM episode;  
DELETE FROM hat_hauptschauspieler;  
DELETE FROM serie;  
DELETE FROM vertrag;  
DELETE FROM kunstler;  
DELETE FROM kunde;  
DELETE FROM sender;  
DELETE FROM streamingdienst;
```

2 Constraints (Integritätsbedingungen) und Referentielle Integrität

2.1 Auflisten von Constraints einer Tabelle

Lassen Sie sich die Constraints **jeder** Tabelle ausgeben und **dokumentieren Sie schriftlich**, welche Primary Key und Foreign Key Constraints vorhanden sind.

In der `psql`-Shell erhalten Sie mit dem Befehl `\d TABELLENNAME` eine Übersicht über die angegebene Tabelle.

In `pgAdmin` können Sie sich mittels Rechtsklick und dann „Properties“ alle Informationen über die angegebene Tabelle anzeigen lassen. Weitere Informationen zu Constraints, beispielsweise auf welche Tabelle ein Foreign Key referenziert, erhalten Sie nur, wenn Sie die Properties des Constraints selbst öffnen. Die Constraints werden Ihnen aufgelistet, wenn Sie neben der Tabelle das **+** Symbol klicken.

```
stdagrab0=> \d ausstrahlung
              Table "streaming.ausstrahlung"
  Column      |      Type      | Collation | Nullable | Defa
ult          |                 |           |          | ult
-----+-----+-----+-----+-----
seriename     | character varying(254) |          | not null |
nummer        | integer          |          | not null |
staffel       | integer          |          | not null |
name          | character varying(254) |          | not null |
ausstrahlungsnummer | integer          |          | not null |
startzeit     | timestamp without time zone |          | not null |
endzeit       | timestamp without time zone |          | not null |
Indexes:
    "pk_ausstrahlung" PRIMARY KEY, btree (seriename, nummer, staffel, name, ausstrahlungsnummer)
    "ausstrahlung_pk" UNIQUE, btree (seriename, nummer, staffel, name, ausstrahlungsnummer)
    "besteht_aus_fk" btree (seriename, nummer, staffel)
    "strahlt_aus_fk" btree (name)
Foreign-key constraints:
    "fk_ausstrah_besteht_a_episode" FOREIGN KEY (seriename, nummer, staffel) REFERENCES episode(seriename, nummer, staffel) ON UPDATE RESTRICT ON DELETE RESTRICT
    "fk_ausstrah_strahlt_a_sender" FOREIGN KEY (name) REFERENCES sender(name) ON UPDATE RESTRICT ON DELETE RESTRICT
```

stdagrabo=> \d bietet_an

Table "streaming.bietet_an"				
Column	Type	Collation	Nullable	Default
name	character varying(254)		not null	
seriennname	character varying(254)		not null	
nummer	integer		not null	
staffel	integer		not null	

Indexes:

"pk_bietet_an" PRIMARY KEY, btree (seriennname, name, nummer, staffel)
"bietet_an_fk" btree (name)
"bietet_an_fk2" btree (seriennname, nummer, staffel)
"bietet_an_pk" UNIQUE, btree (seriennname, name, nummer, staffel)

Foreign-key constraints:

"fk_bietet_a_bietet_an_episode" FOREIGN KEY (seriennname, nummer, staffel) REFERENCES episode(seriennname, nummer, staffel) ON UPDATE RESTRICT ON DELETE RESTRICT

"fk_bietet_a_bietet_an_streamin" FOREIGN KEY (name) REFERENCES streamingdienst(name) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d episode

Table "streaming.episode"				
Column	Type	Collation	Nullable	Default
seriennname	character varying(254)		not null	
nummer	integer		not null	
staffel	integer		not null	
titel	character varying(254)		not null	
imdbrating	numeric			

Indexes:

"pk_episode" PRIMARY KEY, btree (seriennname, nummer, staffel)
"besteht_aus_fk2" btree (seriennname)
"episode_pk" UNIQUE, btree (seriennname, nummer, staffel)

Foreign-key constraints:

"fk_episode_besteht_a_serie" FOREIGN KEY (seriennname) REFERENCES serie(seriennname) ON UPDATE RESTRICT ON DELETE RESTRICT

Referenced by:

TABLE "ausstrahlung" CONSTRAINT "fk_ausstrah_besteht_a_episode" FOREIGN KEY (seriennname, nummer, staffel) REFERENCES episode(seriennname, nummer, staffel) ON UPDATE RESTRICT ON DELETE RESTRICT

TABLE "bietet_an" CONSTRAINT "fk_bietet_a_bietet_an_episode" FOREIGN KEY (seriennname, nummer, staffel) REFERENCES episode(seriennname, nummer, staffel) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d hat_hauptschauspieler

Table "streaming.hat_hauptschauspieler"				
Column	Type	Collation	Nullable	Default
seriennname	character varying(254)		not null	
id	integer		not null	

Indexes:

"pk_hat_hauptschauspieler" PRIMARY KEY, btree (seriennname, id)
"hat_hauptschauspieler_fk" btree (seriennname)
"hat_hauptschauspieler_fk2" btree (id)
"hat_hauptschauspieler_pk" UNIQUE, btree (seriennname, id)

Foreign-key constraints:

"fk_hat_haup_hat_haupt_kunstler" FOREIGN KEY (id) REFERENCES kunstler(id) ON UPDATE RESTRICT ON DELETE RESTRICT

"fk_hat_haup_hat_haupt_serie" FOREIGN KEY (seriennname) REFERENCES serie(seriennname) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d kunde

Table "streaming.kunde"				
Column	Type	Collation	Nullable	Default
typ	character varying(254)		not null	
email	character varying(254)		not null	

Indexes:

"pk_kunde" PRIMARY KEY, btree (email)
"kunde_pk" UNIQUE, btree (email)

Referenced by:

TABLE "vertrag" CONSTRAINT "fk_vertrag_macht_ver_kunde" FOREIGN KEY (email) REFERENCES kunde(email) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d kunstler

Table "streaming.kunstler"				
Column	Type	Collation	Nullable	Default
id	integer		not null	
vorname	character varying(254)		not null	
nachname	character varying(254)		not null	

Indexes:

"pk_kunstler" PRIMARY KEY, btree (id)
"kunstler_pk" UNIQUE, btree (id)

Referenced by:

TABLE "hat_hauptschauspieler" CONSTRAINT "fk_hat_haup_hat_haupt_kunstler" FOREIGN KEY (id) REFERENCES kunstler(id) ON UPDATE RESTRICT ON DELETE RESTRICT

TABLE "serie" CONSTRAINT "fk_serie_hat_stoff_kunstler" FOREIGN KEY (id) REFERENCES kunstler(id) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d sender

Table "streaming.sender"				
Column	Type	Collation	Nullable	Default
name	character varying(254)		not null	

Indexes:

"pk_sender" PRIMARY KEY, btree (name)
"sender_pk" UNIQUE, btree (name)

Referenced by:

TABLE "ausstrahlung" CONSTRAINT "fk_ausstrah_strahlt_a_sender" FOREIGN KEY (name) REFERENCES sender(name) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d serie

Table "streaming.serie"				
Column	Type	Collation	Nullable	Default
seriennamen	character varying(254)		not null	
id	integer		not null	

Indexes:

"pk_serie" PRIMARY KEY, btree (seriennamen)
"hat_stoffentwickler_fk" btree (id)
"serie_pk" UNIQUE, btree (seriennamen)

Foreign-key constraints:

"fk_serie_hat_stoff_kunstler" FOREIGN KEY (id) REFERENCES kunstler(id) ON UPDATE RESTRICT ON DELETE RESTRICT

Referenced by:

TABLE "episode" CONSTRAINT "fk_episode_besteht_a_serie" FOREIGN KEY (seriennamen) REFERENCES serie(seriennamen) ON UPDATE RESTRICT ON DELETE RESTRICT

TABLE "hat_hauptschauspieler" CONSTRAINT "fk_hat_haup_hat_haupt_serie" FOREIGN KEY (seriennamen) REFERENCES serie(seriennamen) ON UPDATE RESTRICT ON DELETE RESTRICT

stdagrabo=> \d streamingdienst

Table "streaming.streamingdienst"				
Column	Type	Collation	Nullable	Default
name	character varying(254)		not null	

Indexes:

"pk_streamingdienst" PRIMARY KEY, btree (name)
"streamingdienst_pk" UNIQUE, btree (name)

Referenced by:

TABLE "bietet_an" CONSTRAINT "fk_bietet_a_bietet_an_streamin" FOREIGN KEY (name) REFERENCES streamingdienst(name) ON UPDATE RESTRICT ON DELETE RESTRICT

TABLE "vertrag" CONSTRAINT "fk_vertrag_macht_ver_streamin" FOREIGN KEY (name) REFERENCES streamingdienst(name) ON UPDATE RESTRICT ON DELETE RESTRICT

```
stdagrab=> \d vertrag
Table "streaming.vertrag"
  Column          |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name             | character varying(254) |           | not null |
 email            | character varying(254) |           | not null |
 monatspreis      | numeric                |           | not null |
 vertragslaufzeit | date                   |           | not null |
Indexes:
    "pk_vertrag" PRIMARY KEY, btree (name, email)
    "macht_vertrag_fk" btree (name)
    "macht_vertrag_fk2" btree (email)
    "vertrag_pk" UNIQUE, btree (name, email)
Foreign-key constraints:
    "fk_vertrag_macht_ver_kunde" FOREIGN KEY (email) REFERENCES kunde(email) ON UPDATE RESTRICT ON DELETE RESTRICT
    "fk_vertrag_macht_ver_streamin" FOREIGN KEY (name) REFERENCES streamingdienst(name) ON UPDATE RESTRICT ON DELETE RESTRICT
```

2.2 Referentielle Integrität

Versuchen Sie, folgende Datensätze direkt einzufügen bzw. zu verändern und **protokollieren Sie schriftlich**, was passiert bzw. welche Meldungen erscheinen:

1. Fügen Sie eine Ausstrahlung von S3E1 für die Serie „Lupin“ beim ZDF ein.
2. Verändern Sie den Stoffentwickler von „The Orville“ auf die Person Nr. 99.
3. Löschen Sie die Serie „Breaking Bad“.

Falls Sie eine Fehlermeldung bekommen, bzw. die Aktion fehlschlägt, sollte Ihr Protokoll eine Aussage darüber enthalten, warum dies geschieht.

Erstellen Sie für (nur) die FK-Beziehung zwischen den beiden Tabellen „Episode“ und „Serie“ und die CRUD-Operationen eine Matrix und entscheiden Sie für jede Kombination (also 8 Zellen), ob die referentielle Integrität grundsätzlich verletzt werden kann.

1.

```
INSERT INTO ausstrahlung (serienname, nr, staffel, name, ausstrahlungsnummer,
startzeit, endzeit)
VALUES ('Lupin', 1, 3, 'ZDF', 1, '2023-06-01 17:30:00', '2023-06-01 18:30:00');
```

```
ERROR: Key (serienname, nummer, staffel)=(Lupin, 1, 3) is not present in table "episode".insert or update on table "ausstrahlung"
violates foreign key constraint "fk_ausstrah_besteht_a_episode"
```

```
ERROR: insert or update on table "ausstrahlung" violates foreign key constraint "fk_ausstrah_besteht_a_episode"
SQL state: 23503
```

```
Detail: Key (serienname, nummer, staffel)=(Lupin, 1, 3) is not present in table "episode".
```

Problem: Die bestimmte Folge von Lupin ist nicht in der Datenbank

2.

```
SET SEARCH_PATH TO STREAMING;  
UPDATE Serie SET id = 99 WHERE serienname = 'The Orville'
```

ERROR: Key (id)=(99) is not present in table "künstler".insert or update on table "serie" violates foreign key constraint "fk_serie_hat_stoff_kunstler"

ERROR: insert or update on table "serie" violates foreign key constraint "fk_serie_hat_stoff_kunstler"
SQL state: 23503
Detail: Key (id)=(99) is not present in table "künstler".

Problem: Es ist kein Künstler(Stoffentwickler) mit der ID 99 vorhanden

3,

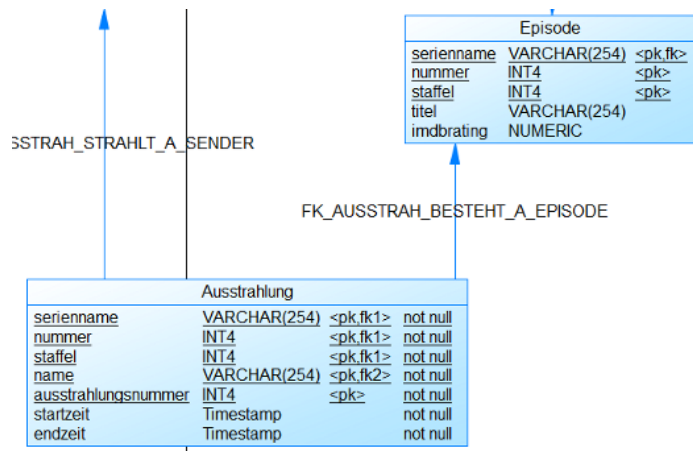
```
SET SEARCH_PATH TO STREAMING;  
DELETE FROM Serie WHERE serienname = 'Breaking Bad'
```

ERROR: Key (serienname)=(Breaking Bad) is still referenced from table "episode".update or delete on table "serie" violates foreign key constraint "fk_episode_besteht_a_serie" on table "episode"

ERROR: update or delete on table "serie" violates foreign key constraint "fk_episode_besteht_a_serie" on table "episode"
SQL state: 23503
Detail: Key (serienname)=(Breaking Bad) is still referenced from table "episode".

Problem: Andere Tabellen besitzen eine Referenz auf den Eintrag "Breaking Bad", das Löschen des Eintrags würde die referentielle Integrität verletzen. Da nicht mit Cascade gelöscht.

wurde, werden die Referenzen nicht zusätzlich entfernt und ein Fehler tritt auf.



	Create	Read	Update	Delete
Episode	Ja wenn keine dazugehörige Serie vorhanden	keine	Wenn Serienname in Episode geändert	Keine
Serie	Keine	keine	Wenn Serienname geändert: Seriennamen in Episode nicht geändert	Wenn Serie gelöscht und Episoden dazu existieren

2.3 Foreign Key Constraints mit cascade Option

Als nächstes werden Sie **Constraints** verändern. Hierzu kopieren Sie zunächst das SQL-Skript zum Erstellen der Datenbank. Anschließend editieren Sie **die Kopie**. Suchen Sie, an welcher Stelle der foreign key constraint definiert wird, der zwischen *Serie* und *Episode* besteht. An welcher weiteren Stelle müssten Sie den foreign-key constraint ändern, um die Serie “Breaking Bad” aus der Datenbank löschen zu können?

Ändern Sie an diesen beiden Stellen den Passus `on delete restrict` zu `on delete cascade` ab.

Speichern Sie das Skript und laden Sie dieses in die Datenbank (Ihre alten Tabellen werden gelöscht und die neuen erstellt). Fügen Sie ebenfalls die Daten erneut ein, in dem Sie Ihr `insert`-Skript erneut ausführen.

1. Löschen Sie nun die Serie „Breaking Bad“. Dabei sollten keine weiteren Fehlermeldungen ausgegeben werden.
1. Überprüfen Sie, ob weitere Datensätze, evtl. auch in anderen Tabellen, verändert wurden. Falls Sie Änderungen entdecken, protokollieren Sie diese.
2. Interpretieren Sie, welche Auswirkungen `on delete cascade` auf Ihre Daten hat.

Probieren Sie nun, die Serie „The Orville“ zu löschen. Passiert dabei das Gleiche wie bei “Breaking Bad”? Wenn nein, was unterscheidet “The Orville” hier von der Serie „Breaking Bad“?

Stellen Sie vor dem nächsten Praktikum wieder den ursprünglichen Zustand her!

```
267
268 alter table Ausstrahlung
269     add constraint FK_AUSSTRAH_BESTEHT_A_EPISODE foreign key (seriename, nummer, staffel)
270     references Episode (seriename, nummer, staffel)
271     on delete cascade on update restrict;
```

```
278 alter table Episode
279     add constraint FK_EPISODE_BESTEHT_A_SERIE foreign key (seriename)
280     references Serie (seriename)
281     on delete cascade on update restrict;
```

```
298 alter table bietet_an
299     add constraint FK_BIETET_A_BIETET_AN_EPISODE foreign key (seriename, nummer, staffel)
300     references Episode (seriename, nummer, staffel)
301     on delete cascade on update restrict;
```

```
313 alter table hat_Hauptschauspieler
314     add constraint FK_HAT_HAUP_HAT_HAUPT_SERIE foreign key (seriename)
315     references Serie (seriename)
316     on delete cascade on update restrict;
```


1.

```
1 SET SEARCH_PATH TO STREAMING;  
2 DELETE FROM Serie WHERE serienname = 'Breaking Bad'
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 58 msec.

2.

In den folgenden Tabellen wurden alle Einträge in Bezug auf "Breaking Bad" gelöscht:

- Serie
- Episode
- bietet_an
- hat_Hauptschauspieler

Künstler bleibt unberührt.

3. Überall wo "on delete cascade" gesetzt wird, wird die Referenz mitgelöscht.

4. Es funktioniert (Fehler möglich wenn kein cascade bei ausstrahlung, da breaking bad keine ausstrahlung hat und für die vorherige aufgabe nicht notwendig war)

Abgabe

1. Legen Sie das Protokoll zu Teilaufgabe 2.1 und 2.2 auf der vorherigen Seite dem Betreuer vor. Sie müssen auf Nachfrage erklären können, welche Constraints auf einer Tabelle liegen. Sie müssen weiterhin erklären können, welche Modellierungsoption im ER-Modell zu diesem Constraint geführt hat.
2. Legen Sie für jeden verletzten Constraint aus Teilaufgabe 4.2 auf der vorherigen Seite dar, welcher Constraint genau verletzt wurde. Überlegen Sie weiterhin, warum ein Fehlschlagen der Operation sinnvoll in Bezug auf die Konsistenz und Korrektheit der Daten sein kann.
3. Erklären Sie dem Betreuer die Bedeutung von cascade. Geben Sie genau an, welche Datensätze aus welchen Tabellen durch das cascade gelöscht wurden.