

Datenbanken 1: Praktikum 2

Thorsten Peter, Michael Roth, Johann Schaible

Lernziele

Im Rahmen dieser Praktikumaufgabe sollen Sie die Ziele erreichen:

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

Vorbemerkungen

Einreichen der Lösung Hiermit ist das Hochladen auf git gemeint, was bis zum Montag **vor** Ihrem eigentlichen Praktikumstermin zu erledigen ist.

Finale Abgabe Dies meint die **im** Praktikumstermin stattfindende Abgabe Ihres Praktikums beim jeweiligen Dozenten.

Einleitung - Verbale Beschreibung des semantischen Datenbankmodells

In diesem Praktikum wird das bestehende Modell von Praktikum 2 um weitere Entities und Daten erweitert.

Kunden (bzw. Passagiere) haben einen Vor- und Nachnamen und werden durch eine Kundennummer eindeutig identifiziert. Jeder Kunde hat ein Bonusmeilenkonto, welches beim Erstellen eines neuen Kunden standardmäßig auf 0 gesetzt werden soll.

Kunden können Abflüge buchen. Die Buchung kostet Geld, wobei der konkrete Betrag ein individueller Preis ist, der von vielen Faktoren abhängt (z.B. Flug, Abflug, Sitzklasse, Buchungszeitpunkt).

Flugzeuge müssen gewartet werden. Jede Wartung die für ein Flugzeug durchgeführt wurde, wird in Form eines Wartungsprotokolls archiviert. Jedes Wartungsprotokoll hat ein Datum und eine Flugfreigabe (ja/nein) abgespeichert. Da Flugzeuge regelmäßig gewartet werden müssen, gibt es für ein Flugzeug mehrere dieser Wartungsprotokolle.

1 ER-Modell

Fügen Sie folgendes Ihrem (**ER!**)-Modell hinzu:

1. Die Entity *Passagier* so wie oben beschrieben
2. Modellieren Sie die Entity *Wartung* so wie oben beschrieben.
3. Fügen Sie eine Relationship *Buchung* ein, so dass diese in Abhängigkeit zwischen Passagieren (bzw. Kunden) und Abflügen existiert.

2 Relationenmodell / Physisches Modell

1. Überprüfen Sie mittels „Check Model“¹ ihr Modell auf Fehler oder Warnungen und beheben Sie diese.

¹Rechtsklick auf den Hintergrund ⇒ Check Model oder F4

- Transformieren Sie Ihr ER-Modell in ein Relationenmodell². Achten Sie darauf, als Datenbank die Option „PostgreSQL 9.x“ auszuwählen.
- Überprüfen Sie das physische Modell ebenfalls mit Check Model¹ und beheben die dort angezeigten Fehler und Warnungen.

Hinweis: Falls eine Warnung „Index inclusion“ erscheint, können Sie diese ignorieren.

- Generieren Sie das SQL-Skript zum Erzeugen der Datenbankschemata³
- Erzeugen Sie die Datenbankschemata in der Postgres Datenbank, wie in den Tutorial Videos beschrieben.

Zur Erinnerung: PowerDesigner erzeugt zu Beginn des Scripts eine Anzahl an drop-Befehlen, die bestehende Tabellen löschen. Logischerweise sind die zu löschenden Objekte allerdings beim ersten Starten des Scriptes noch nicht vorhanden. PgAdmin bricht allerdings leider beim ersten Fehler ab, daher ist es wichtig, dass Sie **beim ersten Ausführen** die entsprechenden drop Befehle nicht ausführen.

Falls es beim erneuten Ausführen zu Fehlern kommt (insbesondere, dass eine Tabelle nicht gelöscht werden kann), editieren Sie das Skript und fügen hinter jeden `drop table` Befehl ein `cascade` hinzu.

3 Daten einfügen

Fügen Sie nun die Daten aus der beigelegten Datei `Daten-p2.txt` zusätzlich zu den Daten aus Praktikum 1 ein:

- Erweitern Sie das bestehende Insert Script aus Praktikum 1, so dass die neuen Daten ebenfalls in die Datenbank eingefügt werden.
- Schreiben Sie am Anfang des Scriptes 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?

Hinweis: In Postgres können Sie Datumswerte als strings übergeben, beispielsweise `'28.04.2017'`.

4 Constraints und Referentielle Integrität

4.1 Auflisten von Constraints einer Tabelle

Lassen Sie sich die Constraints **jeder** Tabelle ausgeben und **protokollieren 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.

4.2 Referentielle Integrität

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

- Einen Abflug für die Flugnummer „DB-123“ am 01.10.2018.
- Einen Flug von IATA-Code „FRA“ nach „ABC“.

²Wählen Sie `Tools >> Generate Physical Data Model` oder `Strg + Shift + P`

³Wählen Sie `Database >> Generate Database` oder `Strg + G`

3. Einen weiteren Flug von IATA-Code „ABC“ nach „FRA“.
4. Ein Wartungsvorgang mit Flugfreigabe für die Maschine „D-XXXX“.
5. Löschen Sie den Flughafen London Heathrow.⁴

Falls Sie eine Fehlermeldung bekommen, bzw. die Aktion fehl schlägt, muss Ihr Protokoll eine Aussage darüber enthalten **warum** dies geschieht.

4.3 Foreign Key Constraints mit cascade Option

Als nächstes werden Sie sich mit **Constraints** beschäftigen. Hierzu kopieren Sie zunächst das SQL-Skript zum Erstellen der Datenbank. Anschließend editieren Sie **die Kopie**. Suchen Sie, an welcher Stelle die foreign key constraints definiert werden, die zwischen `Flug` und `Flughafen` bestehen. Suchen Sie darüber hinaus den foreign key constraint, welcher zwischen `Abflug` und `Flug` besteht.

Ändern Sie an diesen 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 den Flughafen „Chigaco O’Hare International Airport“ mit dem IATA-Code „ORD“. Dabei sollten keine weiteren Fehlermeldungen ausgegeben werden.
2. Überprüfen Sie, ob weitere Datensätze, evtl. auch in anderen Tabellen verändert wurden. Falls Sie Änderungen entdecken, protokollieren Sie diese.
3. Interpretieren Sie, welche Auswirkungen `on delete cascade` auf Ihre Daten hat.

Probieren Sie nun, den Flughafen Frankfurt (IATA-Code „FRA“) zu löschen. Passiert dabei das Gleiche wie beim vorherigen Flughafen? Wenn nein, was unterscheidet den Flughafen FRA von ORD? Protokollieren Sie die gewonnenen Erkenntnisse.

Stellen Sie vor dem nächsten Praktikum wieder den Zustand vor Teilaufgabe 4.3 her!

Einreichen der Lösungen

Laden Sie rechtzeitig, also am Montag **vor** den Praktikumstermin, folgende Dateien auf git **in einen Unterordner** **P2** hoch:

- Das in Aufgabe 1 auf Seite 1 mit PowerDesigner erweiterte ER-Diagramm als `.jpeg` oder `.png` Datei
- Das in Aufgabe 2 auf Seite 1 mit PowerDesigner generierte Relationenmodell ebenfalls als `.jpeg` oder `.png` Datei
- Das von PowerDesigner erstellte SQL Create Script
- Das von Ihnen in Aufgabe 3 auf der vorherigen Seite erweiterte SQL Insert Script
- Das Protokoll aus Aufgabe 4 auf der vorherigen Seite
 - Als Format wählen Sie entweder `.pdf`, `.md` oder `.adoc`.
 - Sie können wahlweise eine Datei für die gesamte Aufgabe anfertigen oder drei verschiedene Dateien, eine für jede Unteraufgabe.

⁴Syntax: `delete from flughafen where iata='LHR';`

Abgabe der finalen Lösung im Praktikum

- Zeigen Sie dem Betreuer Ihr ER Modell aus diesem Praktikum vor.
- Demonstrieren Sie dem Betreuer, dass die zusätzlichen Daten aus diesem Praktikum eingefügt worden sind. Verwenden Sie dazu live vorgeführte SQL-Select Statements ähnlich wie in Praktikum 1.
- Legen Sie das Protokoll zu Teilaufgabe 4.1 auf Seite 2 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.
- Legen Sie für jeden verletzten Constraint aus Teilaufgabe 4.2 auf Seite 2 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.
- Erklären Sie dem Betreuer die Bedeutung von cascade. Geben Sie genau an, welche Datensätze aus welchen Tabellen durch das cascade gelöscht wurden.