



Datenbanken

- Übung 5 -
*Konzeptuelles (physisches) Datenbank
Schema*

Harm Knolle / Markus Schneider

DBS Übung 05.docx vom 28.05.2024 14:41:00

Druck vom 28.05.2024 21:13:00

**Prof. Dr. H. Knolle
Hochschule Bonn-Rhein-Sieg
Fachbereich Informatik
Grantham-Allee 20
53757 Sankt Augustin**

Inhaltsverzeichnis

1	Logische und physische Datenbankschemata.....	3
2	Installation der Oracle-Entwicklungsumgebung	4
3	Erstellung des SQL-Installationsskripts	5
3.1	Grundsätzliche Vorgehensweise.....	5
3.2	Aufbau des Installationsskripts	5
3.2.1	Oracle-spezifische Steuerkommandos	6
3.2.2	Clear Database Section	6
3.2.3	Table Section	7
3.2.4	NOT NULL Constraint Section	8
3.2.5	Primary Key Constraint Section	8
3.2.6	Unique Key Constraint Section.....	9
3.2.7	Foreign Key Constraint Section.....	9
4	Erstellung des physischen Datenbankschemas.....	10
4.1	Einrichtung der Verbindung zum Übungs-Datenbanksystem	10
4.2	Ausführung des SQL-Installationsskripts	10
4.2.1	Erstellung der Datenbank.....	11
4.2.2	Überprüfung der Datenbank.....	11
5	Einreichung Ihrer Ergebnisse.....	12

1 Logische und physische Datenbankschemata

Das in der vierten Übung grafisch erstellte konzeptionelle Schema beschreibt die Speicherstrukturen der zu erstellenden Datenbank mittels logisch relationalen Konzepten wie z.B. Tabellen, Schlüssel und Fremdschlüssel. Das so beschriebene Schema eignet sich somit als Grundlage für die Implementierung relationaler Datenbanken.

Die Implementierung relationaler Datenbanken erfolgt mittels der Datenbanksprache SQL, indem die zuvor erstellten grafischen Konzepte in eine textuelle Form transformiert werden. Das Resultat ist ein SQL-Installationsskript mit Programmieranweisungen zur Erstellung der Datenbank. Derartige Skripte sind weiterhin konzeptionell logischer Natur, da sie die zukünftige Datenbank ebenfalls mit den o.g. logisch relationalen Konzepten beschreiben. Zur Abgrenzung des grafisch erstellten „logisch relationalen konzeptionellen Schemas“ aus Übung 4 wird die textuelle Repräsentation nicht mehr „Schema“, sondern „Datenbankschema“ genannt, genau genommen „logisch relationales konzeptionelles Datenbankschema“.

Mit der erfolgreichen Ausführung des SQL-Skripts über ein Datenbankmanagementsystem wird das interne „physische Datenbankschema“ auf den sekundären Speichermedien wie z.B. einer Festplatte erstellt. Anschließend steht die Datenbank zur Benutzung zur Verfügung und kann mit Daten befüllt werden. Das interne physische Datenbankschema ist nicht mehr logisch konzeptioneller Natur, da es u.a. beschreibt, wie die o.g. logischen Konzepte datenbankintern bzw. physisch realisiert werden (z.B. sortiert, geclustert, komprimiert, ...) und welche Indexstrukturen intern verwendet werden. Die genaue Kenntnis der entsprechenden internen Strukturen ist für den reinen Datenbankentwurf und den Datenbankbetrieb nicht mehr zwingend erforderlich, sondern spielt insbesondere im Rahmen der Datenbankadministration und der Datenbankoptimierung eine große Rolle.

In dieser Übung kommt ein Datenbankmanagementsystems des Herstellers Oracle zum Einsatz. An dieser Stelle sei angemerkt, dass sich die Anbieter von relationalen Datenbankmanagementsystemen, trotz der SQL-Standardisierung, vielfach nicht vollständig an die normierte SQL-Syntax halten, sodass sich die SQL-Skripte nicht immer von allen relationalen Datenbankmanagementsystemen ausführen lassen.

Die Vorgehensweise zur Implementierung der Datenbank ist wie folgt:

- Installation der Oracle-Entwicklungsumgebung.
- Erstellung des Datenbankschemas auf Grundlage des in der Übung 4 spezifizierten logischen Schemas mit SQL (SQL-Installationsskript).
- Erstellung des physischen Datenbankschemas durch Ausführen des SQL-Installationsskripts.

2 Installation der Oracle-Entwicklungsumgebung

Im Rahmen dieser Übung wird mit der Entwicklungsumgebung „Oracle SQL-Developer“ gearbeitet, die zum einen über einen komfortablen Editor zum Erstellen der SQL-Skripte verfügt (siehe Kapitel 3). Zum anderen wird ein SQL-Kommandointerpreter angeboten, der die Kommunikation mit dem Datenbanksystem übernimmt und somit die Ausführung der SQL-Skripte übernimmt (siehe Kapitel 4).

Der SQL-Developer kann kostenlos unter der folgenden URL bezogen werden (Anmeldung erforderlich):

- <https://www.oracle.com/database/sqldeveloper/technologies/download/>

Bitte achten Sie bei der privaten Installation auf die Java-Laufzeitumgebung, die automatisch mitinstalliert wird. Das Produkt steht für Windows, MacOS und Linux zur Verfügung.

In den Übungsräumen des Fachbereichs ist es sowohl unter Windows als auch unter Linux vorinstalliert. Hier wird das Programm wie folgt gestartet:

1. Unter Windows: Start Menü → Developer → Oracle → SQL Developer (bitte nicht die „.exe“-Datei aufsuchen und starten).
2. Unter Linux: sqldeveloper auf Kommandozeile eingeben (es gibt kein Icon im Menu).

3 Erstellung des SQL-Installationsskripts

Erstellen Sie nun ein SQL-Installationsskript auf der Grundlage des von Ihnen entwickelten logischen relationalen Schemas der Übung 4 unter Zuhilfenahme des SQL-Developers.

3.1 Grundsätzliche Vorgehensweise

Gehen Sie bitte wie folgt vor:

- a) Laden Sie die Vorlage-Datei „`eigen_concept.sql`“ aus LEA herunter. Der Aufbau wird in Abschnitt 3.2 erläutert.
- b) Öffnen Sie den SQL-Developer (siehe Kapitel 2).
- c) Um Probleme mit der Zeichenkodierung zu vermeiden, stellen Sie bitte sicher, dass Ihr SQL Developer mit UTF-8 arbeitet (Menü: Extras → Voreinstellungen → Umgebung → Codierung).
- d) Öffnen Sie die Datei „`eigen_concept.sql`“ im SQL-Developer. Sie wird nun in einem eigenen SQL-Arbeitsblatt angezeigt, in dem Sie die Datei wie in einem Texteditor bearbeiten können.
- e) Programmieren Sie Ihre Datenbankobjekte in der Sprache SQL und fügen Sie diese dem Installationsskript hinzu. Schließen Sie jeden Befehl mit „;“ ab.
- f) Spezifizieren Sie zuerst die Tabellen, dann die Konzepte zur Überwachung der Pflichtattribute, dann die Primärschlüssel, dann eventuell vorhandene alternative (bzw. sekundäre) Schlüssel und zuletzt die Fremdschlüssel (**Frage**: warum diese Reihenfolge?).
- g) Man kann grundsätzlich mehrere Schemaobjekte im Rahmen einer Befehlszeile programmieren (z.B. Tabelle mit Primärschlüssel und allen Pflichtfeldern). Erstellen Sie jedoch jedes Schemaobjekt bitte im Rahmen einer gesonderten SQL-Anweisung (**Frage**: warum?).
- h) Verzichten Sie auf die Erstellung von Domänen, da Oracle an dieser Stelle eine eigene Syntax verwendet.

3.2 Aufbau des Installationsskripts

Die Vorlage-Datei „`eigen_concept.sql`“ enthält bereits die Grundstruktur für ein SQL-Installationsskript zum Anlegen Ihres physischen Datenbankschemas. Bitte ändern Sie die Reihenfolge der Abschnitte und die Abschnittsmarkierungen nicht, sie werden vom Praktomaten in dieser Form erwartet.

Das Skript enthält die folgenden Abschnitte:

- Oracle-spezifische Steuerkommandos
- Clear Database Section
- Table Section
- NOT NULL Constraint Section
- Primary Key Constraint Section
- Unique Key Constraint Section
- Foreign Key Constraint Section

Die SQL-Befehle zur Erzeugung und zum Löschen von Tabellen, sowie die Erstellung von Pflichtattributen, Primärschlüsseln, alternativen Schlüsseln sowie von Fremdschlüsseln wurden im Kapitel 5 der Vorlesung vorgestellt.

- Bitte verwenden Sie für Ihr Skript ausschließlich die in den folgenden Abschnitten angegebene Syntax, da der Praktomat nur eine Teilmenge der Oracle-DDL-Syntax (Data Definition Language von SQL) versteht.

3.2.1 Oracle-spezifische Steuerkommandos

Bevor mit der Installation der Datenbank über SQL-DDL-Befehle begonnen wird, führt das Skript einige Oracle-spezifische Steuerkommandos (SQL*Plus-Anweisungen) aus:

```
spool eigen_concept.log
set echo on
set linesize 80
set pagesize 50
```

Damit werden die Bildschirmausgaben einschließlich der Fehlermeldungen in eine Datei mit den Namen „**eigen_concept.log**“ protokolliert, eingestellt, dass jedes Kommando vor der Ausführung explizit ausgegeben bzw. protokolliert werden soll (**Frage**: wo finden Sie die Datei „**eigen_concept.log**“ in Ihrem Dateisystem?), der Zeilenumbruch der Protokolldatei auf 80 Zeichen gelegt und die Seitengröße der Protokolldatei auf 50 Zeilen begrenzt (**Frage**: warum macht man das?).

Das Skript endet mit der SQL*Plus-Anweisung (letztes Steuerkommando):

```
spool off
```

Damit wird das Protokoll mit dem Namen „**eigen_concept.log**“ geschlossen.

3.2.2 Clear Database Section

Ziel eines Installationsskripts besteht darin, eine Datenbank vollständig, fehlerfrei und reproduzierbar in einem Schritt auf einem System anzulegen. Das funktioniert nur dann, wenn

zuvor die Reste einer vorhandenen (ggf. unvollständigen oder fehlerhaften) Installation vollständig entfernt werden. Löschen Sie daher vor jeder Ausführung alle existierenden Schemaobjekte Ihrer Datenbank (auch die fehlerfrei angelegten Schemaobjekte). **Bevor** das Skript mit der Erstellung der Tabellendefinition beginnt, sollte es **alle** Tabellen einschließlich Ihrer „constraints“ löschen (**Frage:** Warum? Was bedeutet „cascade constraints“?).

Fügen Sie pro Tabelle die folgende Zeile vor allen „**CREATE TABLE**“-Anweisungen am Anfang des Skripts:

```
DROP TABLE <sqlTabelle> CASCADE CONSTRAINTS;
```

Verwenden Sie bitte für die Tabellenbezeichner folgende Namenskonvention, um aus den Bezeichnern Ihres logischen Schemas qualifizierte Bezeichner abzuleiten:

```
sqlTabelle: <kürzel>_tab_<table>
```

Dies ist in vielen Fällen eine sinnvolle Praxis, um Namenskollisionen zu vermeiden, z.B. kann es dadurch nicht zu Konflikten mit den Tabellen der Hochschuldatenbank kommen, die Sie in der Übung 6 anlegen werden. Im Rahmen des Übungsbetriebs verlässt sich der Praktomat auf nach dieser Regel qualifizierte Namen, um die Tabellen Ihres logischen Schemas in Ihrem SQL-Installationsskript wiederzufinden.

Wäre das Kürzel Ihres Informationssystems z.B. „**HIS**“ (siehe Reiter „Thema“ in der **.ods**-Datei Ihres Diskursbereiches) und hieße eine Tabelle in Ihrem logischen relationalen Schema „**Person**“, so wäre der Oracle-Bezeichner „**his_tab_person**“ und der entsprechende **DROP TABLE**-Befehl müsste lauten:

```
DROP TABLE his_tab_personen CASCADE CONSTRAINTS;
```

Hinweis: Tabellennamen in SQL sind standardmäßig case-insensitive.

3.2.3 Table Section

In diesem Abschnitt ist für jede anzulegende Tabelle ein **CREATE TABLE**-Befehl mit folgender Syntax zu erstellen:

```
CREATE TABLE <sqlTabelle> (
    <spalte1> <datentyp1>,
    <spalte2> <datentyp2>,
    ...
);
```

Für den Oracle-Tabellenbezeichner gelten wieder die Konventionen aus Abschnitt 3.2.2, für das Konvertieren der DBMain-Datentypen in Oracle-Datentypen verwenden Sie bitte folgende Regeln:

DBMain	Oracle
char(n)	char(n)
varchar(n)	varchar(n)
numeric(n,d) bzw. num(n,d)	numeric(n,d)
boolean(1)	numeric(1)
date(n): Datumsangaben	date
date(n): Datumsangaben + Uhrzeit	date
date(n): Datum + Uhrzeit mit Millisekunden	timestamp
date(n): Andere Temporalwerte, z.B. Uhrzeit	char(n)

Hinweise

- In Oracle gibt es keinen speziellen **boolean**-Typ, solche Werte können als Zahl kodiert werden, wobei **false=0** und **true=1** gilt.
- Für die gängigsten Temporalwerte wird **date** oder **timestamp** verwendet.
- Für spezielle Temporalwerte (Uhrzeit ohne Datum, Wochentage, ...), die Sie in DBMain eventuell als **date(n)** kodiert haben, verwenden Sie bitte **char(n)**.
- Falls ein Spaltennamen ein reserviertes Keyword ist (z.B. start), muss dieser in Anführungszeichen gesetzt werden, damit er von Oracle akzeptiert wird.

3.2.4 NOT NULL Constraint Section

In diesem Abschnitt ist für jede nicht-optionale Spalte ein **NOT NULL Constraint** zu erstellen:

```
ALTER TABLE <sqlTabelle>
    ADD CONSTRAINT <kürzel>_nn_<tabelle>_<spalte>
    CHECK (<spalte> IS NOT NULL);
```

Hinweise

- Für Spalten mit einem **Primary Key Constraint** ist das nicht nötig, da bei diesen **NULL**-Werte grundsätzlich unzulässig sind.
- Falls der abgeleitete Name des Constraints die Oracle-Längenbeschränkung überschreitet, kürzen Sie diesen bitte einfach sinngemäß ab.

3.2.5 Primary Key Constraint Section

In diesem Abschnitt ist für jeden Primärschlüssel ein **PRIMARY KEY Constraint** zu erstellen:

```
ALTER TABLE <sqlTabelle>
    ADD CONSTRAINT <kürzel>_pk_<tabelle>
    PRIMARY KEY (<pkSpalte1>, <pkSpalte2>, ...);
```

Hinweis: Falls der abgeleitete Name des Constraints die Oracle-Längenbeschränkung überschreitet, kürzen Sie diesen bitte einfach sinngemäß ab.

3.2.6 Unique Key Constraint Section

In diesem Abschnitt ist für jeden alternativen Schlüssel ein **UNIQUE KEY Constraint** zu erstellen:

```
ALTER TABLE <sqlTabelle>
    ADD CONSTRAINT <kürzel>_uk_<tabelle>_<spalten>
        UNIQUE (<ukSpalte1>, <ukSpalte2>, ...);
```

Hinweis: Falls der abgeleitete Name des Constraints die Oracle-Längenbeschränkung überschreitet, kürzen Sie diesen bitte einfach sinngemäß ab.

3.2.7 Foreign Key Constraint Section

In diesem Abschnitt ist für jeden Fremdschlüssel ein **FOREIGN KEY Constraint** zu erstellen:

```
ALTER TABLE <sqlTabelle>
    ADD CONSTRAINT <kürzel>_fk_<tabelle>_<zielTabelle>
        FOREIGN KEY (<fkSpalte1>, <fkSpalte2>, ...)
        REFERENCES <sqlZielTabelle>;
```

Hinweis: Falls der abgeleitete Name des Constraints die Oracle-Längenbeschränkung überschreitet, kürzen Sie diesen bitte einfach sinngemäß ab.

4 Erstellung des physischen Datenbankschemas

Der Erstellung der Datenbank erfolgt über den in Kapitel 2 installierten Oracle SQL-Developer.

4.1 Einrichtung der Verbindung zum Übungs-Datenbanksystem

1. Auf dem Übungs-Datenbanksystem wurde für Sie ein Benutzer-Account eingerichtet. Um sich am Übungs-Datenbanksystem mit dem Namen „oracledb“ und Ihrem Account anzumelden benötigen Sie in der Regel drei Parameter:
 - a. Adresse des Datenbanksystems:
 - i. Name des Rechners: **moskito.inf.h-brs.de**
 - ii. Port auf dem Rechner: **1521**
 - iii. Name des Datenbanksystems (SID): **oracledb**
 - b. Benutzerkennung: <**Ihre FB02-Benutzerkennung**>
 - c. Passwort: <**identisch mit Ihrer Benutzerkennung, also Passwort = Benutzerkennung! und nicht Ihr privates Passwort**>
2. Beachten Sie, dass der externe Zugang auf das Datenbanksystem eine VPN-Verbindung zum Hochschulnetz voraussetzt. Hinweise finden Sie auf den Webseiten des Fachbereichs unter
 - <https://faq.infcs.de/vpn>
3. Starten Sie den SQL-Developer (siehe Kapitel 2) und erstellen Sie die Verbindung mit Ihren persönlichen Verbindungsdaten und geben Sie der Verbindung einen „sprechenden“ Namen (Reiter „Verbindungen“ → „+“-Button). Mit dem Button „Test“ können Sie überprüfen, ob die Verbindung erfolgreich hergestellt werden kann.

Wenn die Verbindung erfolgreich ist, haben Sie Zugang zum Datenbanksystem „oracledb“ auf dem Rechner „moskito“ unter dem „Port 1521“.

4.2 Ausführung des SQL-Installationsskripts

In Oracle werden unterschiedliche Datenbanken über den Namen ihrer Eigentümer identifiziert. Der Name Ihrer Datenbank entspricht somit Ihrer Benutzerkennung. Da Sie als Benutzer vom Datenbankadministrator bereits angelegt worden sind, existiert Ihre Datenbank bereits, sie ist initial allerdings noch „leer“.

4.2.1 Erstellung der Datenbank

Durch die Ausführung des in Kapitel 3 entwickelten Installationsskripts wird nun das physikalische Datenbankschema erstellt, das der konzeptuellen Systemebene der Drei-Schichten-Architektur der Datenbank entspricht (siehe Vorlesungsskript):

- a) Öffnen Sie Ihre zuvor erstellte Skript-Datei „`eigen_concept.sql`“ im Oracle SQL Developer (Datei → Öffnen ...). Stellen Sie sicher, dass Sie UTF-8 eingestellt haben (siehe Abschnitt 3.1).
- b) Führen Sie das Skript aus (Button „Skript ausführen“) und beobachten Sie die Ausgabe auf dem Bildschirm.
- c) Öffnen Sie die erstellte Protokolldatei „`eigen_concept.log`“.
- d) Achten Sie auf Fehler bei der Ausführung (einfach zu erkennen am Prefix „`ORA-<n>`“).
 - a. Versuchen Sie den Grund der Fehler zu erkennen und korrigieren Sie Ihr Installationsskript.
 - b. Wiederholen Sie die Prozedur, bis das Skript fehlerfrei durchläuft und das Datenbankschema vollständig angelegt worden ist.
 - c. Beim ersten Durchlauf fehlen Ihre Tabellen, deshalb werden die `DROP TABLE` Anweisungen am Beginn des Scripts fehlschlagen (diese Probleme werden vom Praktomaten ignoriert).

4.2.2 Überprüfung der Datenbank

Überprüfen Sie ob Ihr Datenbankschema vollständig und korrekt angelegt worden ist. Gehen Sie hierbei bitte wie folgt vor:

- a) Die Tabellen finden Sie im Navigator (linke Seite im Oracle SQL Developer).
- b) Die Spalten mit den Datentypen finden Sie in der Anzeige der besitzenden Tabelle unter dem Reiter „Spalten“.
- c) Die Constraints finden Sie in der Anzeige der besitzenden Tabelle unter dem Reiter „Spalten“.

5 Einreichung Ihrer Ergebnisse

Sie entscheiden, ob Sie Ihre Ergebnisse im Rahmen der Übung aktiv präsentieren und somit zur Diskussion stellen und Feedback erhalten oder „nur“ im Rahmen der Klausurzulassung einreichen wollen.

Die Einreichung Ihrer Ergebnisse erfolgt über den Praktomaten (Link im LEA-Kurs), der während des Uploads eine formale Umfangs- und Konsistenzprüfung durchführt und Ihnen zu dieser Ebene ein unmittelbares Feedback gibt. Es handelt sich dabei allerdings nicht um eine semantisch-inhaltliche Prüfung Ihrer Lösung. Für die inhaltliche Diskussion und für Feedback zu dieser Ebene ist Ihre Arbeitsgruppe, das LEA-Forum und der Übungsbetrieb vorgesehen.

Eine Diskussion Ihrer Ergebnisse ist im Rahmen des Übungstermins möglich, wenn Sie:

1. Eine formal gültige Einreichung rechtzeitig über den Praktomaten durchführen.
2. In der enthaltenen .ods-Datei bestätigen, dass Sie bereit sind, Ihre Ergebnisse im Rahmen des Übungstermins zu präsentieren und Sie Feedback und eine Diskussion wünschen. Für diese Angabe existiert in der Tabelle „Thema“ ein entsprechendes Feld.

Bitte beachten Sie, dass die tatsächliche Präsentation Ihrer Ergebnisse insbesondere von der Menge der zur Verfügung gestellten Ergebnisse abhängig ist und leider nicht garantiert werden kann.

Eine Einreichung umfasst Ihre Ergebnisse der Übungen 1-4 sowie ein ausführbares und korrekt aus Ihrem logischen relationalen Schems abgeleitetes SQL-Installationsskript „`eigen_concept.sql`“ (siehe Kapitel 2). Legen Sie das zugehörige physische Datenbankschema auf dem Übungs-Datenbanksystem durch Ausführung Ihres SQL-Installationsskripts an (siehe Abschnitt 4.2).

Sie brauchen nicht das vollständige logische Schema als SQL-Installationsskript umzusetzen, es reicht für die Übung, wenn Sie einen Ausschnitt von mindestens 7 Tabellen transformieren. Wichtig ist, dass die von Ihnen umgesetzten Tabellen und Fremdschlüssel eine einzige Zusammenhangskomponente bilden und die relationale Entsprechung Ihrer Beziehungskette aus Übung 2 enthalten (siehe Übung 4, Abschnitt 4.1).