



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Datenbanken

*- Übungen 6, 7 -
Datenbankanfragen*

sowie

*Individualaufgabe zur
Klausurzulassung Teil III*

Harm Knolle / Markus Schneider

DBS Übung 06+07.doc vom 11.06.2024 19:01:00

Druck vom 11.06.2024 20:19:00

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

Inhaltsverzeichnis

1	Datenbankanfragen mit SQL.....	4
1.1	Einspielen der Übungsdatenbank „dbs“.....	4
1.2	Das semantische konzeptionelle Schema der Übungsdatenbank.....	5
1.3	Das logisch relationale Schema der Übungsdatenbank	6
2	Datenbankanfragen auf der Übungsdatenbank.....	7
2.1	Erstellung der Datenbankanfragen.....	7
3	Datenbankanfragen auf der eigenen Datenbank	8
3.1	Auswahl Ihrer Übungstabellen.....	8
3.2	Import von Beispieldatensätzen.....	8
3.2.1	Einfügen und Lesen von Temporalwerten	9
3.3	Anfragen auf Ihren Datensätzen	10
4	Einreichung Ihrer Ergebnisse	11
4.1	Einreichung der Übungen 6 und 7	11
4.2	Einreichung zur Klausurzulassung Teil III	11

1 Datenbankanfragen mit SQL

Nachdem Sie in der letzten Übung ein physisches Schema für Ihre eigene Datenbank erstellt haben, geht es nun um das Üben konkreter Datenbankanfragen mit SQL. SQL bietet in diesem Zusammenhang vielfältige Möglichkeiten, die wir aufgrund der Komplexität dieses Themas auf zwei Übungseinheiten bzw. Dateien aufgeteilt haben, die Sie bitte auf LEA herunterladen:

- Übung 6: **db6_query_ue6.sql**
- Übung 7: **db7_query_ue7.sql**

Diese Übungen (siehe Kapitel 2) basieren auf einer vorbereiteten Übungsdatenbank, die Sie unter Ihrem eigenen moskito-Account bearbeiten sollen. Das Anlegen und der Aufbau der Datenbank sind in den Abschnitten 1.1 bis 1.3 beschrieben.

Für die **Klausurzulassung** sind zusätzliche Anfragen auf Ihrer eigenen Datenbank durchzuführen (siehe Kapitel 3). Bitte beachten Sie, dass es sich um eine Individualaufgabe handelt, die jedes Teammitglied unter dem eigenen moskito-Account und auf der eigenen Beziehungskette bearbeiten muss.

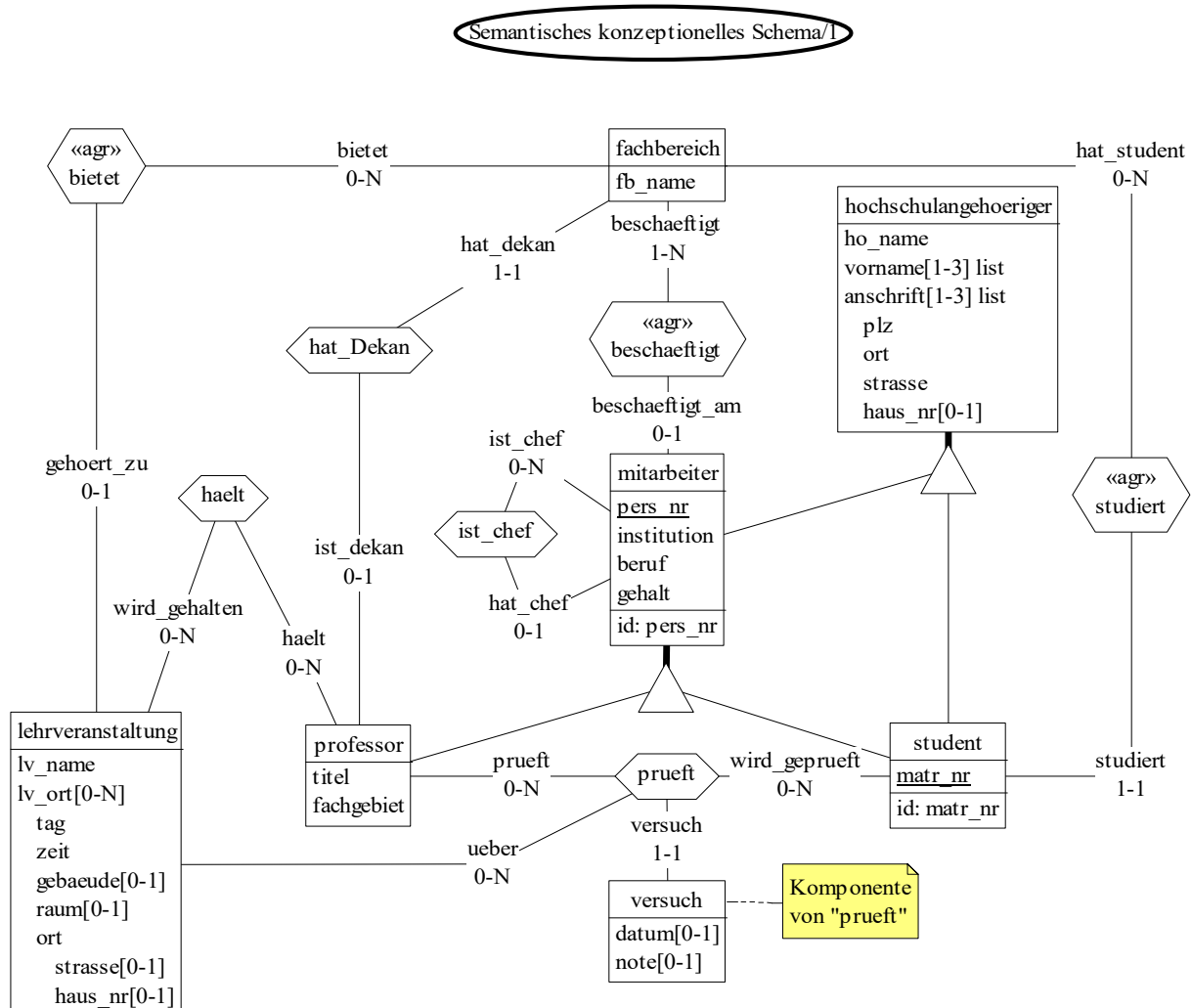
1.1 Einspielen der Übungsdatenbank „db6“

Im ersten Schritt bauen Sie die Übungsdatenbank „db6“ auf. Gehen Sie dabei wie folgt vor:

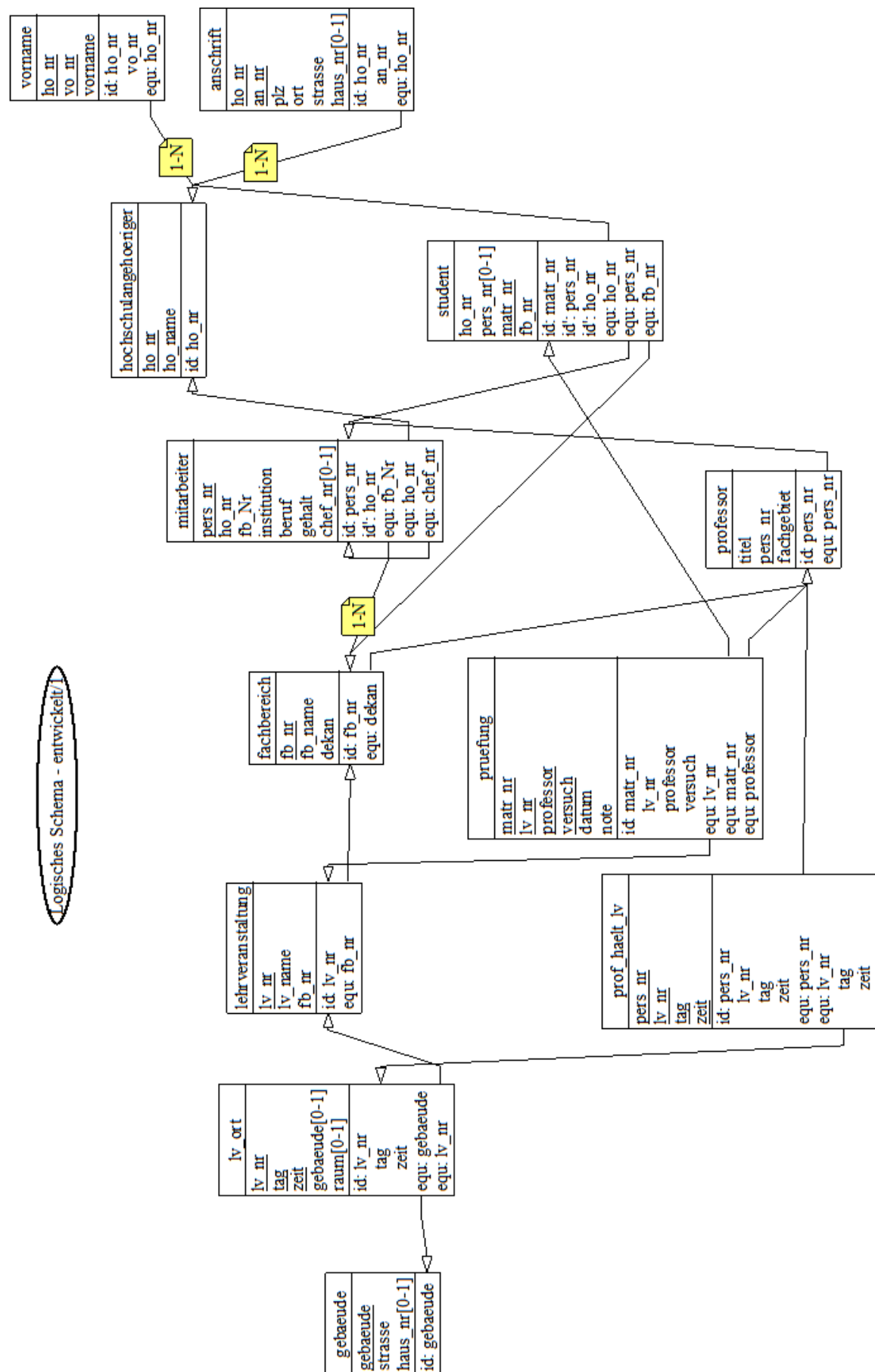
- a) Laden Sie Archiv **Uebungsdatenbank.zip** von LEA herunter und entpacken Sie es.
- b) Das Archiv **Uebungsdatenbank.zip** enthält die folgenden Dateien:
 - a. **db6.lun**: Projektdatei für DB Main mit semantischem und logischem Schema
 - b. SQL-Skripte zum Aufbau der Datenbank:
 - i. **db6_batch.sql**: Sammel-Skript, startet die anderen Skripte
 - ii. **db6_concept.sql**: das logisch relationale Datenbankschema
 - iii. **db6_test_concept**: Test mit Ausgabe der Tabellenstrukturen
 - iv. **db6_intern.sql**: das interne Schema (Indexe, für uns nicht relevant)
 - v. **db6_data.sql**: Einfügen von Beispieldaten
 - vi. **db6_test_data.sql**: Testzugriffe auf die Beispieldaten
- c) Öffnen Sie das Skript **db6_batch.sql** im Oracle SQL Developer (beachten Sie bitte, dass Sie **UTF-8** als Kodierung eingestellt haben, siehe Übungsblatt 5). Starten Sie das Skript über „Skript ausführen“.

- a. Die Schemastrukturen besitzen alle den Projekt-Präfix "**db**s_...".
- b. Das Skript **db**s_batch.sql führt die anderen Skripte der Reihe nach aus und legt eine Protokolldatei mit dem Namen **db**s_batch.log an. Ignorieren Sie bitte die Fehlermeldungen bzgl. der **DROP TABLE**-Statements beim ersten Durchlauf.
- d) Überprüfen Sie die Existenz der Datensätze (Oracle SQL Developer):
 - a. Tabelle im Navigator auswählen und anklicken.
 - b. Im Hauptfenster auf den Reiter „Daten“ klicken.

1.2 Das semantische konzeptionelle Schema der Übungsdatenbank



1.3 Das logisch relationale Schema der Übungsdatenbank



2 Datenbank Anfragen auf der Übungsdatenbank

2.1 Erstellung der Datenbank Anfragen

Zur Erstellung Ihrer Datenbank Anfragen gehen Sie bitte wie folgt vor:

- a) Öffnen Sie die jeweilige Datei (**db_query_ue6.sql** bzw. **db_query_ue7.sql**) im SQL Developer.
- b) Überführen Sie die in der Datei beschriebenen Anfragen in passende SQL-SELECT-Statements. Jeder SELECT-Befehl muss innerhalb der entsprechenden Aufgaben-Markierung stehen, damit der Praktomat diese findet, z.B. für Aufgabe 2.1.1 a):

```
-- <211a>
```

```
SELECT lv_name FROM db_tab_lehrveranstaltung;
```

```
-- </211a>
```

- c) Einige nützliche Hinweise:

- Grundlage für Ihre Anfragen ist das logische Schema auf der vorigen Seite.
- Schließen Sie jeden Befehl mit „;“ ab.
- Die physischen Tabellen besitzen alle den Präfix „**db_tab_**“.
- Zum Ausführen einer Anfrage setzen Sie den Cursor in die entsprechende Zeile und verwenden Sie „Anweisung ausführen“.

3 Datenbankabfragen auf der eigenen Datenbank

***** Ab hier beginnt die Individualaufgabe für die Klausurzulassung *****

Die folgende Aufgabe basiert auf Ihrer eigenen Datenbank als Ergebnis der Übungen 1 bis 5.

Zur Entwicklung der Datenbankabfragen für Ihre eigene Datenbank verwenden Sie bitte die Vorlagedatei „**eigen_query.sql**“, die Sie ebenfalls auf LEA finden. Gehen Sie bitte beim Arbeiten mit dieser Datei analog zu Kapitel 2 vor, beachten Sie insbesondere die Zeichenkodierung und die Abschnittsmarkierungen für den Praktomaten.

3.1 Auswahl Ihrer Übungstabellen

Als Grundlage für Ihre eigenen SQL-Anfragen dient die relationale Umsetzung Ihrer in Übung 2 ausgewählten Beziehungskette. Die zugehörigen physischen Tabellen haben Sie in Übung 5 unter Ihrem persönlichen moskito-Account angelegt.

Aus Ihrer ausgewählten Beziehungskette resultieren mindestens vier Tabellen. Sollte Ihre Beziehungskette eine versionierte N:M-Beziehung enthalten, kann auch eine fünfte Tabelle entstanden sein, sofern Sie eine separate Tabelle für die Versionierungsentitäten umgesetzt haben. In diesem Fall vernachlässigen Sie bitte diese fünfte Tabelle.

Die vier Tabellen sollen im Folgenden mit Beispieldatensätzen befüllt werden, wobei die Fremdschlüssel-Constraints zwischen den vier Tabellen aktiviert bleiben sollen, damit die Konsistenz der Daten gemäß Ihrer Modellierung gewährleistet ist. Daraus ergibt sich, dass Sie eine geeignete Einfügereihefolge bestimmen müssen.

Fremdschlüssel-Constraints, die aus der Kette der vier ausgewählten Tabellen herauszeigen, sollten Sie für die Übung deaktivieren, um keine Datensätze in weitere Tabellen einfügen zu müssen. Verwenden Sie dazu folgenden SQL-Befehl:

```
ALTER TABLE <sqlTable> DISABLE CONSTRAINT <fkConstraint>;
```

Dadurch werden die Referenzen auf die „externen“ Tabellen nicht mehr geprüft und stören somit nicht beim Einfügevorgang.

3.2 Import von Beispieldatensätzen

Fügen Sie mindestens 20 Datensätze in die von Ihnen ausgewählten vier Tabellen ein. Gehen Sie bitte wie folgt vor:

- a) Die 20 Datensätze sollen semantisch sinnvoll auf die vier Tabellen aufgeteilt werden.

- b) Verwenden Sie die einfachste Befehlsform zum Einfügen von Datensätzen:

```
INSERT INTO <Tabelle> VALUES ( <Wert_1>, ..., <Wert_n> );
```

- c) Die Schlüsselwerte (Primär-, Fremdschlüssel und sofern vorhanden alternative Schlüssel) legen Sie bitte manuell fest.
- d) Achten Sie beim Einfügen der Datensätze wegen der Fremdschlüssel auf eine geeignete Reihenfolge der Einfügevorgänge.
- e) Um bei der Aufgabe 3.3 c) den „outer join“ sinnvoll anwenden zu können und damit bei der Aufgabe 3.3 f) mindestens ein Datensatz gefunden wird, achten Sie bitte darauf, dass nicht alle an der N:M-Beziehung beteiligten Datensätze von der N:M-Beziehungstabelle aus referenziert werden. Es müssen sowohl in der „linken“ als auch in der „rechten“ Tabelle Datensätze existieren, die keinen „Partner“ finden und in der Beziehungstabelle nicht auftauchen.
- f) Schließen Sie die Insert-Befehle mit „**COMMIT**;“ ab. Das ist notwendig, damit die Daten tatsächlich gespeichert werden und z.B. vom Praktomaten gefunden werden.
- g) Sollten die Insert-Vorgänge scheitern oder möchten Sie die Insert-Vorgänge wiederholen, so ist es am Einfachsten, Ihre gesamte Datenbank zu löschen und neu aufzusetzen, in dem Sie Ihre „**eigen_concept.sql**“ neu ausführen (siehe Übung 5). Dadurch wird sichergestellt, dass keine Reste in Ihrer Datenbank verbleiben. Anschließend können Sie die Datensätze erneut einfügen.

3.2.1 Einfügen und Lesen von Temporalwerten

Temporalwerte (**DATE** oder **TIMESTAMP**) können in Oracle-Befehlen nicht direkt angegeben werden, sie müssen zunächst über die Funktion **TO_DATE** oder **TO_TIMESTAMP** aus einer Zeichenkette konvertiert werden. Um z.B. eine **DATE**-Spalte mit einem Datumswert zu belegen, ist dies folgendermaßen möglich:

```
INSERT INTO ... VALUES (... , TO_DATE('02.10.2023' , 'DD.MM.YYYY') , ...);
```

Beim Aufruf werden zwei Parameter übergeben, eine String-Repräsentation des Temporalwertes und eine passende Formatmaske, die die Formatierung und verwendete Zeiteinheiten (Tag, Monat, Jahr, Stunden, Minuten, ...) beschreibt. Hier einige gängige Beispiele:

- **TO_DATE('10:45' , 'hh24:mi')**
- **TO_DATE('10:45:33' , 'hh24:mi:ss')**
- **TO_DATE('02.10.2023' , 'DD.MM.YYYY')**
- **TO_DATE('02.10.2023 10:45' , 'DD.MM.YYYY hh24:mi')**
- **TO_TIMESTAMP('02.10.2023 10:45.123000' , 'DD.MM.YYYY hh24:mi.ff')**

Für die Formatierung bei der Ausgabe von Temporalfeldern steht die **TO_CHAR**-Funktion zur Verfügung, z.B.

- **TO_CHAR(<DATE-SPALTE> , 'hh24:mi')**

- `TO_CHAR(<DATE-SPALTE>, 'hh24:mi:ss')`
- `TO_CHAR(<DATE-SPALTE>, 'DD.MM.YYYY')`
- `TO_CHAR(<DATE-SPALTE>, 'DD.MM.YYYY hh24:mi')`

Weitere Details und eine Liste aller möglichen Zeichen für die Formatmaske sind leicht über eine Internetsuche zu finden.

3.3 Anfragen auf Ihren Datensätzen

Überführen Sie die in der Datei „**eigen_query.sql**“ beschriebenen Anfragen 3.3 a) bis 3.3 f) in passende SQL-SELECT-Statements.

4 *Einreichung Ihrer Ergebnisse*

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

4.1 **Einreichung der Übungen 6 und 7**

Die Einreichung Ihrer Ergebnisse erfolgt über den Praktomaten (Links im LEA-Kurs), der während des Uploads eine formale Umfangs- und Konsistenzprüfung durchführt und Ihnen zu dieser Ebene ein unmittelbares Feedback gibt.

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

1. Rechtzeitig eine formal gültige Einreichung über den Praktomaten durchführen.
2. In der enthaltenen „**EigeneDB.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 zur Übung 6 bzw. 7 umfasst folgende Bestandteile:

- a) Spezifikation Ihres Diskursbereiches aus Übung 1 als „**EigeneDB.ods**“-Datei. Hier ist diese Datei nur für die Bestätigung relevant, ob Sie Ihre Einreichung präsentieren / diskutieren möchten.
- b) SQL-Anfrageskripte „**db_query_ue6.sql**“ bzw. „**db_query_ue7.sql**“ (siehe Kapitel 2)

4.2 **Einreichung zur Klausurzulassung Teil III**

Die Dateien „**db_query_ue6.sql**“ und „**db_query_ue7.sql**“ gehören gleichermaßen zum dritten Teil der Klausurzulassung. Zusätzlich reichen Sie bitte ein:

- a) Ihre Ergebnisse der Übungen 1-5
- b) SQL-Datei „**eigen_query.sql**“ mit Datenbankabfragen gemäß Kapitel 3 dieses Übungsblatts.