

## Datenbanken II B: DBMS-Implementierung — 2. Übungsblatt: Zugriffsrechte —

### Hausaufgaben

Geben Sie die Aufgaben dieses Abschnitts bis Mittwoch, 03.11.2021, 18<sup>00</sup>, über die Übungsplattform in StudIP ab. Schreiben Sie die Lösungen in eine `.sql`-Datei bzw. `.txt`-Datei (kein PDF und schon gar kein Word). Anmerkungen zu SQL-Anfragen markieren Sie bitte als Kommentar (so dass die Datei als SQL ausführbar ist).

Die Abgaben nur stichprobenartig kontrolliert. Wenn Ihre Abgabe nicht kontrolliert wurde, bekommen Sie die volle Punktzahl. Wenn Sie später wegen Plagiaten auffallen, oder bei einer Stichprobe eine fast gar nicht gelöste Aufgabe entdeckt wird, können auch alte Abgaben kontrolliert werden. Dann können auch rückwirkend Punkte abgezogen werden.

Sie benötigen 67% der Hausaufgabenpunkte und eine aktive Mitarbeit in den Übungen für die Studienleistung.

Sie haben in den Übungen Zugriff auf eine Oracle-Datenbank (Version 18c) über die Adminer-Webschnittstelle bekommen:

```
https://dbs.informatik.uni-halle.de/db2b/adminer?  
oracle=oracle-18.4-xe-db2b%2FXEPDB1&  
username=scott&db=USERS
```

Beachten Sie, dass das ursprüngliche Passwort abgelaufen war, und ersetzt werden musste. Die Änderung gegenüber dem in der Übung genannten Passwort steht in einer EMail, die über StudIP versandt wurde.

Bitte unter diesem gemeinsamen Test-Account keine Änderungen vornehmen. Die Teilaufgaben e) und f) müssen also zwingend unter dem persönlichen Account vorgenommen werden, den Sie auch bekommen haben.

- a) Schreiben Sie eine Anfrage, die Ihnen alle Tabellen listet, an denen Ihnen direkt Zugriffsrechte gegeben wurden (also nicht über eine Rolle). Drucken Sie den Besitzer der Tabelle (`OWNER`), den Tabellen-Namen (`TABLE_NAME`) und das Recht (`PRIVILEGE`).
- b) Unter diesen Tabellen müsste auch eine Tabelle des Nutzers `BRASS` sein, für die Sie sogar Einfüge-Rechte haben. Schreiben Sie eine SQL-Anfrage, die Ihnen die Spalten dieser Tabelle (`COLUMN_NAME`, `COLUMN_ID`) mit Datentyp (`DATA_TYPE`, `DATA_LENGTH`, `DATA_PRECISION`) anzeigt. Sortieren Sie die Spalten in der deklarierten Reihenfolge (also nach der `COLUMN_ID`).

- c) Sie sollen eine Zeile in diese Tabelle einfügen, bei denen die erste Spalte Ihren Nutzernamen enthält (die Pseudospalte `USER` ist immer der Name des aktuellen Nutzers). Damit die Sache nicht so einfach ist, sind auf der Tabelle Integritätsbedingungen formuliert. Schreiben Sie eine Anfrage, die die Integritätsbedingungen anzeigt.
- d) Anschließend suchen Sie passende Werte aus, die Sie einfügen können, und fügen die Zeile ein als Beleg Ihres Erfolgs. Geben Sie das `INSERT`-Statement ab.
- e) Legen Sie selbst eine Tabelle an mit gleichem Namen, Spalten, Datentypen und Primärschlüssel wie die Tabelle der obigen Aufgaben. Als `CHECK`-Constraint wählen Sie dagegen eine andere Bedingung (gerne eine ganz einfache Bedingung). Geben Sie das `CREATE TABLE`-Statement ab. Fügen Sie eine Zeile in Ihre Tabelle ein.
- f) Geben Sie den Nutzern `BRASS` und `SCOTT` Lese-Rechte und das `UPDATE`-Recht an Ihrer Tabelle (dagegen sollen Sie `INSERT` und `DELETE` nicht erlauben). Geben Sie den entsprechenden SQL-Befehl ab.

Da alle Vorlesungs-Teilnehmer sich als `SCOTT` einloggen können, ist es möglich, dass Ihre Tabelle modifiziert wird. Es ist erwünscht, dabei seinen Nutzernamen anzugeben.

## Wiederholungsaufgaben

Die „Wiederholungsaufgaben“ brauchen Sie nicht abzugeben. Beschäftigen Sie sich aber bitte auch mit diesen Aufgaben. Sie müssen damit rechnen, dass Sie beim Online-Treffen gebeten werden, einen Teil des Vorlesungs-Stoffes zu wiederholen und insbesondere eine der Fragen zu beantworten. Umgekehrt können Sie natürlich auch fragen. Notieren Sie sich Fragen, die Sie gerne in der Übung geklärt haben wollen.

### Wichtiger Hinweis:

Bei mündlichen Prüfungen gibt es auch praktische Aufgaben, z.B. werden Sie wahrscheinlich SQL Anfragen an das Data Dictionary schreiben müssen. Es wäre ein schlimmes Missverständnis, wenn Sie sich nur auf Verständnisfragen vom hier gezeigten Typ vorbereiten würden. Umgekehrt können einzelne dieser Verständnisfragen auch in einer schriftlichen Prüfung vorkommen.

- g) Wie würden Sie in einer mündlichen Prüfung auf folgende Fragen antworten?
- Skizzieren Sie, wie das Schema von Tabellen inklusive Schlüssel in einem Systemkatalog repräsentiert werden kann. Sie müssen nicht das Oracle Data Dictionary auswendig lernen, sondern können auch „ad hoc“ eigene Tabellen entwickeln.
  - Was bedeuten die Datentypen `NUMERIC` und `VARCHAR2` in Oracle?
  - Die Groß-/Kleinschreibung von Tabellen- und Spaltennamen ist bei den meisten Datenbanksystemen egal (wenn nicht Delimited Identifier ". . ." verwendet wurden). Erklären Sie, wie Oracle diese „Case-Insensitivität“ implementiert hat, und was das für Data Dictionary Anfragen bedeutet.

- Warum ist es nützlich, Constraint-Namen im `CREATE TABLE`-Statement anzugeben?
- Kann man eine kurze Erklärung zu einer Tabelle oder einer Spalte in einer Oracle-Datenbank speichern?
- Nennen Sie einige Optionen von Integritätsbedingungen in Oracle, die über die einfache Sicht des ersten Datenbank-Kurses hinausgehen.
- Erläutern Sie „Object Privileges“ (das Standard-Modell für Zugriffsrechte in SQL). Geben Sie ein Beispiel für einen `GRANT`-Befehl. Erklären Sie die Bedeutung der Klausel „`WITH GRANT OPTION`“. Was sollte eine `REVOKE`-Anweisung bewirken?
- Wie könnte man Objekt-Rechte im Data Dictionary repräsentieren? (Sie können die entsprechenden Oracle-Tabellen nennen, oder eine eigene Tabelle skizzieren.)
- Welches Problem wird in Oracle mit „System Privileges“ gelöst? Was wäre eine alternative Lösung, wie sie in anderen Systemen verwendet wird?
- Was ist der Zweck von „Rollen“ in Oracle? Wie kann man sie verwenden?

### Für Interessierte

h) Informieren Sie sich über das Data Dictionary von PostgreSQL:

[<https://www.postgresql.org/docs/current/catalogs.html>]

Beachten Sie, dass Tabellen in `pg_class` gelistet werden (PostgreSQL betont manchmal seine objekt-relationalen Eigenschaften).

Sie können mit folgender URI auf eine PostgreSQL-Datenbank im Adminer zugreifen:

[[https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student\\_gast&db=postgres&ns=](https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student_gast&db=postgres&ns=)]

Das Passwort wurde in der Übung genannt.

i) Schauen Sie sich auch `information_schema.table_privileges` an:

[<https://www.postgresql.org/docs/current/information-schema.html>]

Im Information Schema gibt es u.a. auch folgende Tabellen:

- `column_privileges`,
- `applicable_roles`,
- `role_table_grants` u.s.w.

j) Die Syntax des `CREATE USER`-Befehls in Oracle ist hier dokumentiert:

[<https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-USER.html>]

Entsprechend steht die Dokumentation zum `GRANT`-Befehl hier:

[<https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/GRANT.html>]

k) Die Syntax des `CREATE ROLE`-Befehls in PostgreSQL ist hier beschrieben:

[<https://www.postgresql.org/docs/current/sql-createrole.html>]

Dieser Befehl kann auch zum Anlegen von Benutzern genutzt werden. Die Syntax des `GRANT`-Befehls in PostgreSQL ist hier erläutert:

[<https://www.postgresql.org/docs/current/sql-grant.html>]