Hinweise

- Dieses Skript beinhaltet evtl. Fehler, die von mir gewollt sind.
- Vermutlich gibt es in diesem Skript auch Fehler, die nicht von mir gewollt waren.
- Manche Folien / Beispiele sind unvollständig. Dies ist Absicht.
- Die Lösungen zu den Beispielen werden evtl. in der Vorlesung besprochen.

3. Zugriffsrechte

Unberechtigten Zugriff unterbinden!

Unbeabsichtigte Manipulationen verhindern!

Beispiel: SQL-Injection in Web-Anwendungen

Universität Augsburg University

```
private void writeFeedbackRecord() {
    Connection sqlCon = null;
    try {
        Class.forName(Config.DBDRIVER);
        sqlCon = DriverManager.getConnection(Config.DBURL, Config.DBUSER,
                 Config.DBPASS);
        Statement stmt = sqlCon.createStatement();
        Date d = new Date();
        String sql = "UPDATE FEEDBACK SET INTERACTION_END=" + d.getTime()
                 + ", MINEOPINION="
                 + ratingToNumber(myChoiceRating.getValue())
                 + ", CBOPINION="
                 + ratingToNumber(rickChoiceRating.getValue())
                 + ", PBOPINION="
                 + ratingToNumber(lauraChoiceRating.getValue())
                 + ", MYEXPLANATION='"
                 + sanitizeString(myChoiceReason.getValue())
                 + "', CBEXPLANATION='"
                 + sanitizeString(rickChoiceReason.getValue())
                 + "', PBEXPLANATION='"
                 + sanitizeString(lauraChoiceReason.getValue())
                 + "' WHERE PARTICIPANT='" + userName + "'";

        stmt.executeQuery(sql);

        if (stmt != null) {
            stmt.close();
        }
        if (sqlCon != null) {
            sqlCon.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Warum ist das wichtig?

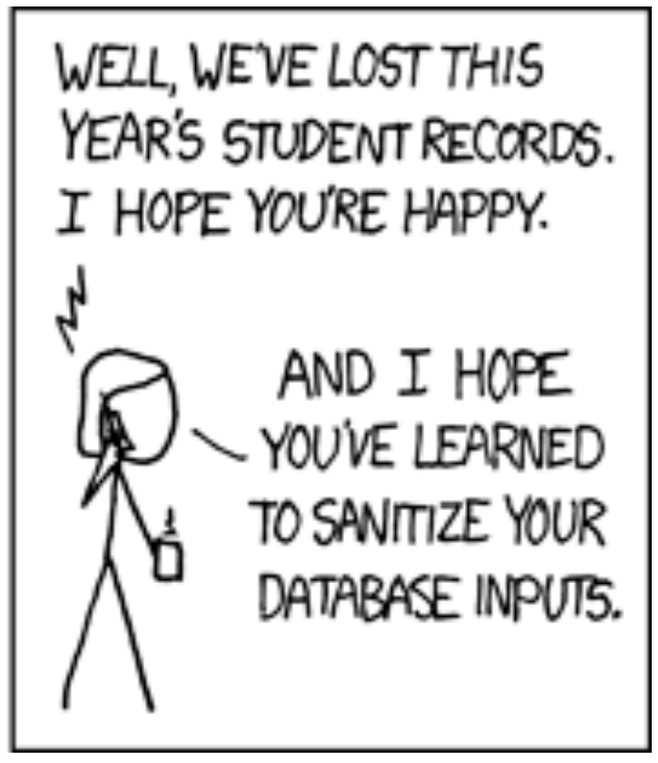
User Input geht direkt in die Query ein!

z.B.: `got you' where 1=1 --`


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.


OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY-


DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.


WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Beispiel - SQL Injection

Universität Augsburg University

Folgenden Ratschlag findet man unter
http://www.dotnetcrack.com/2009/12/default.aspx

> *“Finally, to limit the scope of a SQL injection attack, limit the permissions granted to the database user account the Web application is using. The application generally doesn't need `dbo` or `sa` permissions. The less permission granted to your database the better! Consider using a separate account for each component with data access capabilities to isolate vulnerabilities. For instance, a front-end public interface to your Web site needs more restricted DB access than an internal content management system”*

Benutzerkonten

Ein Konzept zur Modellierung von Zugriffskontrolle

Benutzerkonten

Identifikation: durch Benutzernamen

Attribute:

- Authentifizierungsmethode (intern, extern)
- Passwort zur Authentifizierung (verschlüsselt gespeichert)
- Voreingestellte Tabelspaces für temporären und dauerhaften Speicher
- Tablespace Kontingent (Quota)
- Status des Benutzerkontos (gesperrt, freigeschaltet)
- Status des Passworts (aktuell oder abgelaufen)

von Hand:

- SQL-Befehl zum Erstellen eines Users lautet:

```
CREATE USER <Benutzer> IDENTIFIED BY {<Passwort> |
                                      externally};
```

- SQL-Befehl zum ändern eines Passworts:

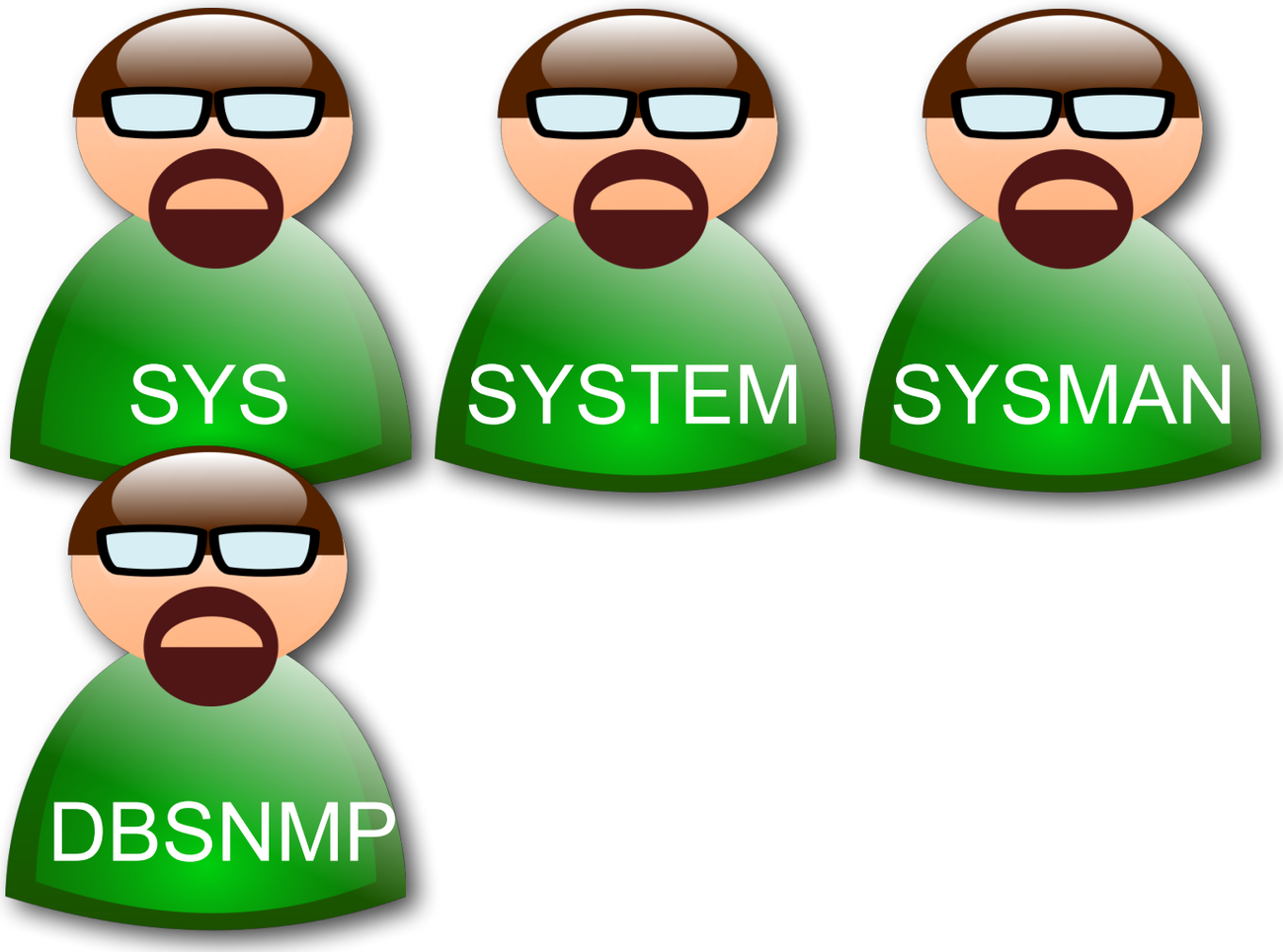
```
ALTER USER <Benutzer> IDENTIFIED BY <neues Passwort>
```

- Gelöscht werden kann ein Benutzer mit:

```
DROP USER <Benutzer> [CASCADE];
```

Ein Benutzer muss nach dem Anlegen mit geeigneten Berechtigungen ausgestattet werden!

Vordefinierte Benutzerkonten



Zur Anmeldung im System / Oracle Enterprise Manager

Management Agent of Database Control

Diese Benutzerkonten werden beim Erzeugen der Datenbank via Oracle Database Configuration Assistant angelegt.

Vorsicht, diese Benutzerkonten sind mit weitreichenden Rechten ausgestattet

Benutzerkonten und Schema

- Ein **Schema** ist ein logischer Container für Datenbankobjekte (Tabellen, Views, Trigger, ...)
- Orthogonal zu **Tablespace**: Ein Tablespace ist eine logische Sicht auf den Speicherbereich, in dem Datenbankobjekte gehalten werden
- **Datenbankobjekte** werden über das Schema adressiert
- Beim Anlegen eines **Benutzerkontos** wird implizit ein gleichnamiges **Schema** erzeugt.

Benutzerkonto und Schema (2)

- Der Schema-Name / Benutzername kann zur eindeutigen Benennung von Objekten verwendet werden:

  Beispiel: `select * from "0815".lectures;`

  Wählt alle Felder der Relation `lectures` im Schema `0815`.

- Wenn man sich als Nutzer anmeldet, ist das zugehörige Schema schon voreingestellt und muss i.A. nicht angegeben werden

In Oracle sind `user' und `schema' fast synonym.
Entsprechend in MySQL: `database' und `schema'

Weitere initiale Benutzerkonten

Universität Augsburg University

- **Interne Benutzerkonten**
  - Manche Oracle Features benötigen eigene Schemas um dort Daten zu hinterlegen
  - Dazu werden eigens Benutzerkonten angelegt, die von anbeginn gesperrt sind und deren Passwörter den Status *expired* haben
  - Diese Benutzerkonten werden nur wegen der implizit erzeugten Schemas angelegt und sind nicht zur Anmeldung gedacht
  - Ein Beispiel ist das **`WKSYS`** Benutzerkonto das von **Oracle Ultra Search** angelegt wird; Oracle Ultra Search dient zur **Volltextsuche** und legt zu diesem Zweck große Indizes an
- **Beispielkonten**
  - Typischerweise bei der Installation angelegt (**`SCOTT/TIGER`**)
  - Die Oracle Dokumentation verweist auf die Beispielkonten um bestimmte Aspekte zu demonstrieren
  - Auch die Beispielkonten müssen erst freigeschalten werden

Universität
Augsburg
University

Benutzerrechte und Rollen

Benutzerrechte

Universität Augsburg University

Benutzerrechte sind grundlegend für die Sicherheit der Datenbank

- beschränken Zugriff auf Daten und Datenbank

- Schränken die Art der ausführbaren SQL Statements ein

Systembezogene Benutzerrechte

- Erlauben die Ausführung bestimmter systemrelevanter Aktionen
  z.B.: `CREATE USER`
- Erlauben die Ausführung von Aktionen die das Schema beeinflussen
  z.B.: `CREATE TABLE`

Objektbezogene Benutzerrechte

Erlauben die Ausführung von Aktionen für bestimmte Schema-Objekte
z.B.: Zeilen zur Tabelle `STUDENT` hinzufügen oder Tabellen anderer Benutzer lesen

Einige Benutzerrechte

Eine Auswahl der verfügbaren **Systemrechte**:

- **CREATE TABLE**
- **UNLIMITED TABLESPACE**
- **CREATE USER**
- **CREATE INDEX**
- **DROP ANY VIEW**
- **ALTER USER**
- **CREATE SESSION**
- **...**

Eine Auswahl der verfügbaren **Objektrechte** für Tabellen:

- **ALTER**
- **DELETE**
- **UPDATE**
- **SELECT**
- **INDEX**
- **REFERENCES**
- **INSERT**
- **EXECUTE**
- **...**

Rollen

Wenn man alle Rechte für alle Benutzer und die gewünschten Objekte von Hand vergeben möchte verliert man leicht den Überblick.

Man möchte die Rechte mehrerer Benutzer gleichzeitig ändern

Manche Benutzerrechte möchte man je nach Anwendung aktivieren

Rollen erlauben all dies zu modellieren!

Rollen

- Rollen sind Zusammenfassungen von Rechten
- Definieren von Rollen:
  CREATE ROLE <Rolle> [**IDENTIFIED BY** <Passwort> |
  NOT IDENTIFIED];
- Rollen können so ähnlich wie Benutzer verwendet werden
- Entfernen einer Rolle:
  DROP ROLE <Rolle> [**CASCADE**];
- Rechte der Rolle **CONNECT**
  ~~**ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SYNONYM,**~~ **CREATE SESSION,** ~~**CREATE VIEW**~~

Rechte vergeben/entziehen

- In Oracle werden **Rollen** und **Systemrechte** mit dem Befehl **`GRANT`** gewährt:

```
GRANT <System- oder Rollennamen>
   [, ..., <System- oder Rollennamen>]
TO <Benutzer- oder Rollenname>
[WITH ADMIN OPTION];
```

- Ein Aufruf kann mehrere Rechte mehreren Benutzern gewähren

- Gewähren von **Objektrechten**:

```
GRANT <Privileg> [(<Spalte>, [, ..., <Spalte>])]
      [, ..., <Privileg> [(<Spalte>, [, ..., <Spalte>])]]
ON [<Besitzer>.]<Objekt>
TO <Benutzer- oder Rollenname>
   [, ..., <Benutzer- oder Rollenname>]
[WITH GRANT OPTION];
```

- Auch hier können mehrere Rechte gleichzeitig mehreren Benutzern zugeordnet werden.
- Die **`GRANT/ADMIN OPTION`** erlaubt Benutzern dieses Recht weiterzugeben (Bei Rollen nicht!)

Rollen können Rollen zugewiesen werden?

Universität Augsburg University

- Hierarchischer Aufbau von Rollen möglich!
- Wird einer Rolle A eine Rolle B zugewiesen, so enthält anschließend die Vereinigung aller Rechte aus A und B.
- Die Rollenhierarchie kann somit die Hierarchie der Rechte in einem Unternehmen nachbilden.

$$\text{HS\_ADMIN\_ROLE} \supseteq \text{HS\_ADMIN\_EXECUTE\_ROLE} \cup \text{HS\_ADMIN\_SELECT\_ROLE}$$

Wie kommt ein Nutzer zu seinen Rechten?

Universität Augsburg University

- Bei der Anmeldung: Oracle aktiviert alle Rechte, die dem Nutzer zugewiesen wurden oder in dessen **Default-Rollen** enthalten sind – Vereinigung der Rechte aller Rollen
- Die Liste der Default-Rollen kann mittels **ALTER USER** geändert werden:
  ```
  ALTER USER Alfons DEFAULT ROLE PETOMOTOMGR, PETOMOTOUSER
  ```
- Die Default-Rolle **ALL** bewirkt, dass alle nachfolgend mit **GRANT** zugewiesenen Rollen zu Default-Rollen werden
- Während einer Session kann mittels des Befehls **SET ROLE** eine Teilmenge der zur Verfügung stehenden Rollen aktiviert werden

  ```
  SET ROLE PETOMOTOUSR IDENTIFIED BY 'password'
  ```

- Alle Rollen abwählen

  ```
  SET ROLE NONE;
  ```

Vermischtes (1)

- Rechte können wieder entzogen werden:
  - **System-Rechte:**
    ```
    REVOKE <Systemprivileg> [ , ..., <Systemprivileg> ]
    FROM <Benutzer- oder Rollenname>
                    [ , ..., <Benutzer- oder Rollenname>]
    ```
  - **Objekt-Rechte:**
    ```
    REVOKE <Privileg> [ , ..., <Privileg> ]
    ON [ <Besitzer> ].<Objekt>
    FROM <Benutzer- oder Rollenname>
                [ , ..., <Benutzer- oder Rollenname>]
    ```
- **Admin** Option bei **`GRANT`**
  - Nur für **System-Berechtigungen** nicht für Objekt-Berechtigungen
  - Nur für Benutzer, nicht für Rollen
  - Rechte werden beim Entziehen nur dem genannten Benutzer entzogen:
    revoke SELECT on "DEF$_ERROR" from SYS
- **With Grant** Option bei **`GRANT`**
  - nur für **Objekt-Berechtigungen**
  - nicht für Rollen
  - Rechte werden beim Entziehen transitiv entzogen

Vermischtes (2)

- Jeder Benutzer hat die Rolle **PUBLIC**
- Vorsicht, PUBLIC wird über die Schnittstellen meist nicht explizit angezeigt
- Berechtigungen können auch für den Benutzer PUBLIC erteilt werden
- Sie stehen dann jedem Benutzer der Datenbank zur Verfügung.

Universität
Augsburg
University

Administratoren und Administratorenrechte

SYSDBA und DBA

SYSDBA und **SYSOPER** sind Berechtigungen, keine Rollen

Besonderheit: Diese können auf die DB zugreifen, auch wenn diese nicht “geöffnet” ist.

Die **Rolle** DBA hat alle Berechtigungen außer SYSDBA und SYSOPER!

DBA wird bei der Installation angelegt

Mögliche Klausuraufgaben