

- [0. Vorbereitung](#)
  - [1. Rechteverwaltung](#)
  - [2. Stored Procedures, Functions, Trigger](#)
  - [3. Transaktionen](#)
  - [4. Views](#)

## 0. Vorbereitung

Verwenden Sie das folgende SQL-Skript um das Schema `company` zu erstellen.

```
CREATE SCHEMA IF NOT EXISTS company;
SET SCHEMA 'company';
DROP TABLE IF EXISTS emp_employee;
DROP TABLE IF EXISTS pro_project;
DROP TABLE IF EXISTS fao_favorite_color;

CREATE TABLE pro_project (
  pro_id          SERIAL,
  pro_name        VARCHAR(255),

  PRIMARY KEY (pro_id)
);

CREATE TABLE fao_favorite_color (
  fao_id          SERIAL,
  fao_color_name  VARCHAR(50),

  PRIMARY KEY (fao_id)
);

CREATE TABLE emp_employee (
  emp_id          SERIAL,
  emp_pro_id      INTEGER,
  emp_email       VARCHAR(50) NOT NULL UNIQUE,
  emp_first_name  VARCHAR(50),
  emp_last_name   VARCHAR(50),
  emp_fao_id      INTEGER,

  PRIMARY KEY (emp_id),
  FOREIGN KEY (emp_pro_id) REFERENCES pro_project(pro_id),
  FOREIGN KEY (emp_fao_id) REFERENCES fao_favorite_color(fao_id)
);

INSERT INTO fao_favorite_color(fao_color_name) VALUES
('red'),
('green'),
('yellow'),
('blue');
```

```

INSERT INTO pro_project (pro_name) VALUES
('tree_planting'),
('car_repair'),
('weather_forecast');

INSERT INTO emp_employee (emp_pro_id, emp_email, emp_first_name, emp_last_name,
emp_fao_id)
VALUES (
  (SELECT pro_id FROM pro_project WHERE pro_name LIKE 'tree_planting'),
  'pete@email',
  'Pete', 'Eat',
  (SELECT fao_id FROM fao_favorite_color WHERE fao_color_name LIKE 'blue')
);
INSERT INTO emp_employee (emp_pro_id, emp_email, emp_first_name, emp_last_name,
emp_fao_id)
VALUES (
  (SELECT pro_id FROM pro_project WHERE pro_name LIKE 'car_repair'),
  'foo@email',
  'Foo', 'Bar',
  (SELECT fao_id FROM fao_favorite_color WHERE fao_color_name LIKE 'yellow')
);
INSERT INTO emp_employee (emp_pro_id, emp_email, emp_first_name, emp_last_name,
emp_fao_id)
VALUES (
  null,
  'fred@email',
  'Fred', 'Bar',
  (SELECT fao_id FROM fao_favorite_color WHERE fao_color_name LIKE 'blue')
);

```

## 1. Rechteverwaltung

1. Legen Sie eine neue Rolle `finance` ohne Login an
2. Geben Sie der Rolle **Leseberichtigung** auf den Projektnamen der Projekttafel
3. Legen Sie einen neuen Nutzer an, welcher Mitglied dieser Rolle ist
4. Loggen Sie sich mit dem Nutzer ein
5. Fragen Sie die Daten der Tabelle ab, auf welche Sie Zugriff gegeben haben
6. Ueberpruefen Sie den Zugriff auf andere Tabellen

## 2. Stored Procedures, Functions, Trigger

1. Erstellen Sie eine Funktion, welche nur die ersten beiden Buchstaben der Email, gefolgt von drei `***` und der Emailendung "@comp.de" zusammenfügt, um die Einträge zu anonymisieren.  
Die Funktion soll einen VARCHAR(100) übernehmen und diesen entsprechend verändern.  
Verwenden Sie die Funktion `SUBSTRING` und `CONCAT` - schlagen Sie nach wie diese Funktionen verwendet werden.
2. Verwenden Sie die Funktion in einer SQL-Abfrage.
3. Verwenden Sie die Funktion in einer Abfrage durch JDBC

## 3. Transaktionen

Realisieren Sie ein Beispiel wie in der Vorlesung beschrieben

1. Legen Sie ein neues Schema mit der Tabelle `konto` an
2. Legen Sie geeignete Testdaten an
3. Führen Sie eine Ueberweisung durch. Verwenden Sie Transaktionen und keine Transaktionen.
4. Verwenden Sie `ROLLBACK` , um eine laufende Transaktion zu unterbrechen
5. Vergleichen Sie die Ergebnisse

## 4. Views

1. Erstellen Sie einen SELECT Befehl, um alle Projekte ohne Mitarbeiter zu finden
2. Erstellen Sie nun damit eine View
3. Erstellen Sie eine neue View, welche alle Mitarbeiter mit ihrer Lieblingsfarbe ausgibt
4. Verwenden Sie diese View in einer Abfrage und geben sie nur Mitarbeiter zurueck, welche mit einem `f` starten