Aufgabenblatt 3
SQL

- Abgabetermin: **Dienstag, 11.06.2019, 23:59**
- Zur Prüfungszulassung muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte.
- Die Aufgaben sollen in Zweiergruppen bearbeitet werden.
- Abgabesytem unter
  `https://www.dcl.hpi.uni-potsdam.de/submit/`
  - ausschließlich pdf-Dateien *im A4-Format*
  - *eine Datei pro Aufgabe* namens Aufgabe-<aufgabenNr>.pdf
  - *jedes* Blatt beschriftet mit Namen

Vorbemerkung und Hinweise zur Bearbeitung

In dieser Übung verwenden wir eine lokale PostgreSQL Installation um Ergebnisse von SQL Anfragen direkt betrachten zu können. Hinweise zur Installation folgen auf der nächsten Seite.

Als Datensatz verwenden wir Daten aus der IMDB und Daten aus der Simulation eines Großhandels (genannt Sales), welche automatisch generiert wurden. Die Daten findet ihr unter `Lehrveranstaltungen\FG Informationssysteme\VL DBS I\Uebung 2018\imdb_database_dump.zip` als PostgreSQL dump. Eine Anleitung für die Installation von PostgreSQL und das Importieren der Daten folgt auf der nächsten Seite.
Hilfreiche/Weiterführende Links zu PostgreSQL:

- `https://www.postgresql.org/docs/9.5/static/index.html`
- `https://www.postgresql.org/docs/9.5/static/queries.html`
- `https://www.postgresql.org/docs/9.5/static/ddl.html`

Hinweise zum IMDb-Datensatz

- Die Tabellen `actor`, `actress`, `producer` und `genre` enthalten eine Spalte mit dem Namen "id". Hierbei handelt es sich um Zeilennummern und nicht um Werte, die etwa einen Film oder eine Person eindeutig identifizieren. Die Tabellen `actor` und `actress` speichern daher eher Rollen von Schauspielern in verschiedenen Filmen als Schauspieler selbst.
- Nimm an, dass Schauspieler, Schauspielerinnen und Produzenten eindeutig über ihre Namen identifiziert werden können.
- Nimm an, dass die "movie_id"-Attribute jeweils das "mid"-Attribut referenzieren. Eine Fremdschlüssel-Eigenschaft gilt allerding nicht automatisch in dem zur Verfügung gestellten IMDb-Datensample. Es gibt also z.B. Schauspieler und Genres, deren zugehörige Filme NICHT existieren!

Hinweise zum Sales-Datensatz

- Der Datensatz wurde automatisch erzeugt und ist ca 1 GB groß. Eine Abbildung des Schemas, sowie genauere Erläuterungen gibt es in den entsprechenden Aufgaben.

Hinweise zur Bearbeitung der Aufgaben

- Benenne aggregierte Spalten so um, dass sinnvolle Spaltennamen ausgegeben werden.

- Anfragen auf Schauspielerinnen *und* Schauspielern sind explizit in der Frage formuliert. D. h. falls eine Frage nur *Schauspieler* erwähnt, soll auch nur die Tabelle `actor` angefragt werden.
- Wenn nach dem Ergebnis einer SQL-Anfrage gefragt ist, dann gib maximal 10 Tupel und die Anzahl aller Tupel an.

Vorbereitungen für Aufgabenblätter 3, 4 und 5: PostgreSQL Installation

Für die Aufgabenblätter 3-5 benötigt ihr eine lokale Installation von Postgresql. Wir haben im folgenden einige Anleitungen zur Installation zusammengefasst. Sollte es bei euch zu Problemen bei der Installation kommen, versucht euch zunächst selbst zu behelfen (im Netz gibt es viele Anleitungen/Tutorials). Bei schwer lösbaren Problemen könnt ihr uns natürlich gerne kontaktieren. Für die Bearbeitung der Aufgaben braucht ihr lediglich die PSQL-Shell (Kommandozeile). Grafische Oberflächen sind nicht nötig (ihr dürft sie aber natürlich verwenden).

a) Hinweise zu Windows Setup und häufigen Problemen:

- Laden Sie den Installer z.B. von dieser Webseite runter:

  `https://www.postgresql.org/download/windows/`

  Während der Installation gibt es die Option, Stack Builder mit zu installieren. Das ist für diese Übung **nicht** notwendig.

  Für die Übung wird nur die SQL Shell "psql" benötigt. Nach Start der Shell muss Enter gedrückt werden, bis man die Passworteingabe erreicht. Danach sollte es folgendermaßen oder ähnlich aussehen wie in Abbildung 1

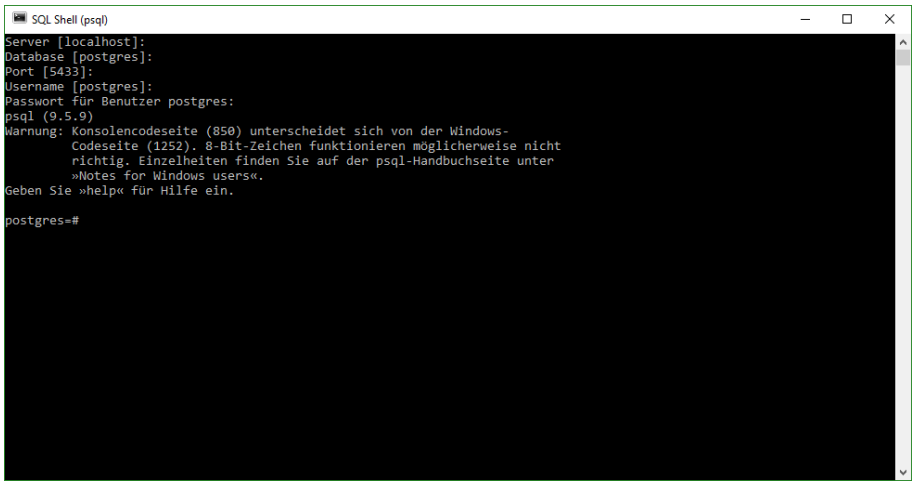

Abbildung 1: Windows SQL Shell nach erfolreicher Installation und Passworteingabe.

  Zugriffsprobleme können im Allgemeinen durch die Ausführung als Administrator gelöst werden. Da der Backslash für Kommandos reserviert ist, müssen Backslashes in Pfaden durch Slashes ersetzt werden.

b) Hinweise zu Linux Setup (Ubuntu 16.04 LTS):

- Eine Shell öffnen und *sudo apt-get install postgresql-contrib-9.5* ausführen
- Nun *sudo -u postgres psql* um PostgreSQL unter dem User postgres zu starten
- Nun sind sie mit der psql Shell verbunden. Sobald Sie eine Datenbank angelegt haben können Sie sich mit dieser direkt aus dem Terminal mit *psql <dbname>* verbinden.

c) Hinweise zu Mac Setup:

- Für die Installation wird empfohlen, den beliebten Packagemanager Homebrew zu nutzen, Ablauf:
  - Terminal öffnen
  - *brew -v* um zu überprüfen, ob Homebrew bereits installiert ist, ggf. installieren siehe `https://brew.sh/`

- *brew install postgresql@9.6* um PostgreSQL 9.6 zu installieren
- am Ende des ausgegebenen Installationslogs wird empfohlen, PostgreSQL zum Path hinzuzufügen; für die Bash-Shell sieht das dann beispielsweise so aus: *echo export PATH="/usr/local/opt/postgresql@9.6/bin:\$PATH" >> /.bashrc*
- Terminalsession beenden und eine neue starten, damit der neu hinzugefügte Path übernommen wird
- PostgreSQL-Installation ist damit abgeschlossen, ab jetzt kann man PostgreSQL folgendermaßen im Hintergrund starten und auch wieder stoppen
  - *brew services start postgresql@9.6*
  - *brew services stop postgresql@9.6* (beim Stoppen werden die angelegten Datenbanken *nicht* gelöscht)
- *psql postgres* um sich mit PostgreSQL zu verbinden
- *\q* um die Verbindung zu beenden

Import der Daten: Siehe nächste Seite

Import der Daten

a) Zip Datei `Lehrveranstaltungen\FG Informationssysteme\VL DBS I\Uebung 2019\data.zip` herunterladen und entpacken

b) PostgreSQL Kommandozeile öffnen

c) Mittels *CREATE DATABASE imdb*; und *CREATE DATABASE sales*; zwei verschiedene Datenbanken erzeugen (mit $\backslash c$ $< dbname >$ könnt ihr zu der entsprechenden Datenbank wechseln)

d) Mittels $\backslash i$ <*pathToFile*> könnt ihr ein SQL-Skript ausführen. Die heruntergeladenen Skripte importiert Daten in die Datenbank mit der ihr gerade verbunden seid. Das Kommando $\backslash d$ zeigt anschließend fälschlicherweise noch keine Tabellen an. Dies wird behoben indem ihr euch mit einer beliebigen anderen Datenbank verbindet und anschließend wieder zurückwechselt. Führt die beiden Import-Skripte *imbd_import.sql* und *sales_import.sql* in den entsprechenden Datenbanken aus. Es kann sein, dass dies bei sales ein paar Minuten dauert.

e) Nun könnt ihr testen ob alles funktioniert hat, indem ihr überprüft ob die in Abbildung 2 und 3 gezeigten Queries die gleichen Resultate liefern. Lediglich der Owner wird bei euch anders sein.

```
leon=# \c imdb
You are now connected to database "imdb" as user "leon".
imdb=# \d
         List of relations
 Schema |    Name    | Type  | Owner
--------+------------+-------+-------
 public | actor      | table | leon
 public | actress    | table | leon
 public | genre      | table | leon
 public | movie      | table | leon
 public | producer   | table | leon
(5 rows)

imdb=# select count(*) from actress;
 count
-------
  3479
(1 row)

imdb=# select * from movie where title = '4 aventures de Reinette et Mirabelle';
                      mid                       |                 title                  | year
------------------------------------------------+----------------------------------------+------
 4 aventures de Reinette et Mirabelle (1987) | 4 aventures de Reinette et Mirabelle | 1987
(1 row)
```

Abbildung 2: Test Queries with Output (IMDB)

```
sales=# select * from orders LIMIT 10;
 o_orderkey | o_custkey | o_orderstatus | o_totalprice | o_orderdate | o_orderpriority |      o_clerk       | o_shippriority |                        o_comment
------------+-----------+---------------+--------------+-------------+-----------------+--------------------+----------------+-----------------------------------------------------------
          1 |     36901 | O             |    173665.47 | 1996-01-02  | 5-LOW           | Clerk#000000951    |              0 | nstructions sleep furiously among
          2 |     78002 | O             |     46929.18 | 1996-12-01  | 1-URGENT        | Clerk#000000880    |              0 |  foxes. pending accounts at the pending, silent asymptot
          3 |    123314 | F             |    193846.25 | 1993-10-14  | 5-LOW           | Clerk#000000955    |              0 | sly final accounts boost. carefully regular ideas cajole carefully. depos
          4 |    136777 | O             |     32151.78 | 1995-10-11  | 5-LOW           | Clerk#000000124    |              0 | sits. slyly regular warthogs cajole. regular, regular theodolites acro
          5 |     44485 | F             |    144659.20 | 1994-07-30  | 5-LOW           | Clerk#000000925    |              0 | quickly. bold deposits sleep slyly. packages use slyly
          6 |     55624 | F             |     58749.59 | 1992-02-21  | 4-NOT SPECIFIED | Clerk#000000058    |              0 | ggle. special, final requests are against the furiously specia
          7 |     39136 | O             |    252004.18 | 1996-01-10  | 2-HIGH          | Clerk#000000470    |              0 | ly special requests
         32 |    130057 | O             |    208660.75 | 1995-07-16  | 2-HIGH          | Clerk#000000616    |              0 | tse blithely bold, regular requests. quickly unusual dep
         33 |     66958 | F             |    163243.98 | 1993-10-27  | 3-MEDIUM        | Clerk#000000409    |              0 | uriously. furiously final request
         34 |     61001 | O             |     58949.67 | 1998-07-21  | 3-MEDIUM        | Clerk#000000223    |              0 | ly final packages. fluffily final deposits wake blithely ideas. spe
(10 rows)

sales=# select count(*) from orders;
  count
---------
 1500000
(1 row)
```

Abbildung 3: Test Queries with Output (Sales)

Aufgabe 1: Deutsch → SQL

Nenne für jede der folgenden natürlichsprachlichen Fragen eine geeignete SQL-Anfrage und führe sie auf den Daten der IMDb aus. Gib auf deiner Abgabe die Anfrage und deren Ergebnis an.

a) Wieviele Schauspielerinnen gibt es? **2 P**

b) Gib alle Producer aus, die keine zugehörigen Einträge in der Filmtabelle haben (nach Producernamen sortiert)! Jeder Producer soll dabei nur einmal ausgegeben werden. **3 P**

c) Gib die Titel aller Filmpaare aus, in denen mindestens ein gemeinsamer Schauspieler mitspielt! Sortiere das Ergebnis nach dem Titel des zweiten Films. **3 P**

d) Gib die Namen der Schauspieler und Produzenten an, die an der Serie "Edge of Night, The" beteiligt waren, und zwar einmal nach Mengen- und ein weiteres Mal nach Multimengensemantik. Hinweis: `UNION` benutzen. **4 P**

e) Erstelle eine Top-3 Liste der Filme mit den meisten Schauspielern und Schauspielerinnen! Sortiere entsprechend. Hinweis: Recherchiere hierzu die `FETCH FIRST` Klausel. **4 P**

f) Erstelle eine Top-3 Liste der Schauspieler und Schauspielerinnen mit den meisten Filmen! Sortiere entsprechend. **4 P**

Aufgabe 2: Deutsch → SQL

Die Aufgabenstellung aus Aufgabe 1 gilt weiter.

a) Gib alle Schauspieler an, die in mindestens einem Film des Genres „Action" spielen und deren Name mit „T" beginnt. **3 P**

b) Gib alle Schauspieler an, die *nur* in Filmen des Genres „Action" spielen und deren Name mit „T" beginnt. **4 P**

c) Gib die Namen aller Producer an, die im Jahr 2001 Filme in beliebten Genres gedreht haben. Ein „beliebtes Genre" sei ein Genre in dem mindestens 200 Filme gedreht wurden. **5 P**

d) Formuliere *eine* Anfrage, die die Jahreszahl und die Anzahl der in diesem Jahr veröffentlichten Filme abfragt für

- das höchste (späteste) vorkommende Jahr und
- das Jahr mit den meisten veröffentlichten Filmen

5 P

Aufgabe 3: Relationale Algebra → SQL

Formuliere die folgenden drei Anfragen der relationalen Algebra als SQL-Anfragen!

Verwendetes Schema:

- Stadt (StadtName, LandID, p1950, p2000, p2015)
  wobei `p1950`, `p2000` und `p2015` die Bevölkerungszahlen in diesen Jahren darstellen
- Land (LandID, Name, Kontinent, Hauptstadt, Bevoelkerung)
- Geographie (LandID, Landfläche, Wasserfläche, Küstenlänge, urbar)
  wobei `urbar` die urbare Fläche des Landes beschreibt

a) $\pi_{Name,Kontinent}(\sigma_{Bevoelkerung>200.000.000}(Land))$ **2 P**

b) $\pi_{Name}(\sigma_{(Bevoelkerung<2*p1950)\vee(Bevoelkerung<4*p2000)}(\sigma_{StadtName=Hauptstadt}(Stadt \bowtie Land)))$ **3 P**

c) $\pi_{Name}(Land \bowtie Geographie) -$
$\pi_{Name}(Land \bowtie (\sigma_{G1.urbar<G2.urbar}(\rho_{G1}(Geographie) \times \pi_{urbar}(\rho_{G2}(Geographie)))))$ **4 P**

Erklärungen zum Sales Datensatz

Für die folgenden Aufgaben betrachten wir den Sales-Datensatz. Dieser modelliert das Geschäft eines Großhandels. Das Schema der Tabellen ist in Abbildung 4 zu sehen. Es handelt sich dabei nicht um ein high-level E/R Diagramm, sondern um eine direkte Abbildung der Tabellen. Die Pfeile zeigen Schlüssel/Fremdschlüssel Beziehungen, aber der Pfeil zeigt hier in die Richtung der 1-zu-n Beziehung (anders als bei E/R Diagrammen! D.h. wie erwartet hat ein Land genau eine Region (Kontinent) aber eine Region mehrere Länder). Die Klammern hinter den Bezeichnungen der Entitätstypen spezifizieren einen Präfix, den jedes Attribut in den Tabellen besitzt (d.h. Partkey heißt in der Relation PART *P_PARTKEY* und in der Relation PARTSUPP *PS_PARTKEY*). Die Tabellen beinhalten folgende Daten:

- **PART** - Einzelteile, welche verkauft werden können. Schlüssel: P_PARTKEY
- **PARTSUPP** - Kombinationen aus Einzelteilen und Lieferanten für diese Teile. Schlüssel: (PS_PARTKEY,PS_SUPPKEY). **Anmerkung: AVAILQTY steht für available quantity also die Anzahl an verfügbaren Teilen bei diesem Lieferanten.**
- **LINEITEM** - Verkäufe einer bestimmten Anzahl eines Einzelteils (Intuitiv: eine Zeile auf der Quittung). Schlüssel: (L_ORDERKEY,L_LINENUMBER)
- **ORDERS** - Bestellungen, Schlüssel: O_ORDERKEY
- **SUPPLIER** - Lieferanten, Schlüssel: S_SUPPKEY
- **CUSTOMER** - Kunden, Schlüssel: C_CUSTKEY
- **NATION** - Länder/Staaten, Schlüssel: N_NATIONKEY
- **Region** - Kontinente, Schlüssel: N_NATIONKEY

Um den Import zu Beschleunigen sind in den euch zur Verfügung gestellten Daten die Schlüssel/Fremdschlüssel nicht als explizite Constraints festgehalten.

Aufgabe 4: Sales-Datensatz: SQL → Deutsch

Gib natürlichsprachlich wieder, wonach folgende SQL-Anfragen suchen:

a)
```
SELECT n_name, sum(l_extendedprice * (1 - l_discount)) as revenue
FROM customer, orders, lineitem, supplier, nation, region
WHERE
    c_custkey = o_custkey
    AND l_orderkey = o_orderkey
    AND l_suppkey = s_suppkey
    AND c_nationkey = s_nationkey
    AND s_nationkey = n_nationkey
    AND n_regionkey = r_regionkey
    AND r_name = 'EUROPE'
    AND o_orderdate >= date '1992-01-01'
    AND o_orderdate < date '1992-01-01' + interval '1' year
GROUP BY n_name
ORDER BY revenue DESC;
```

5 P

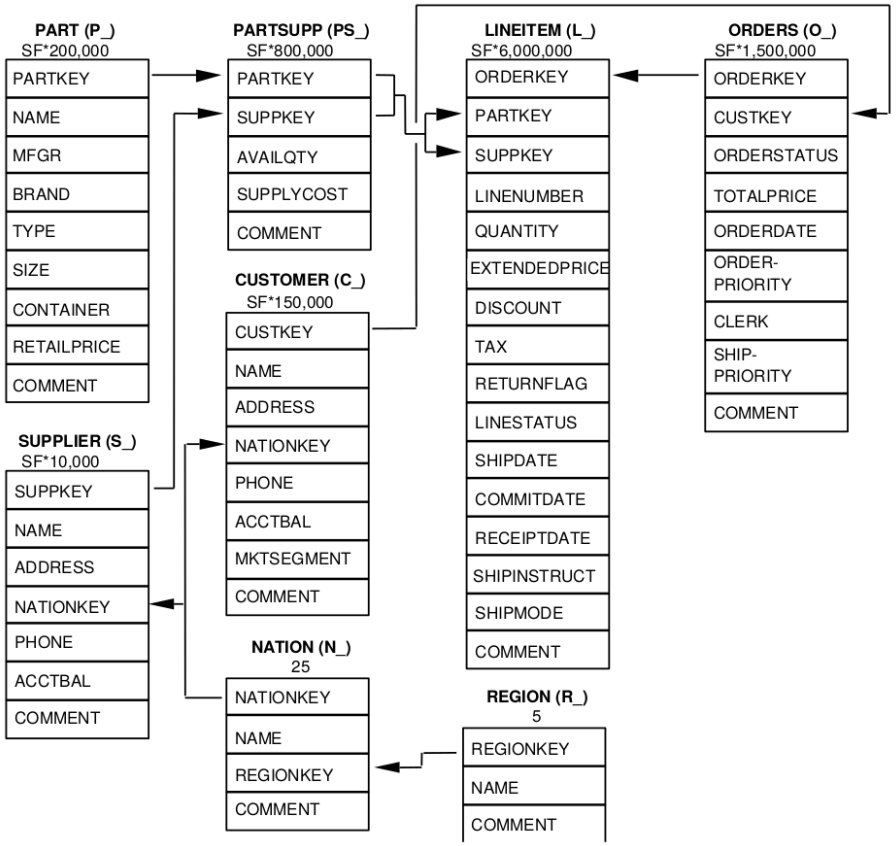

Abbildung 4: Sales Schema

b)
```
SELECT supp_nation, cust_nation, l_year, sum(volume) as revenue
FROM (
    SELECT
        n1.n_name as supp_nation,
        n2.n_name as cust_nation,
        EXTRACT(YEAR FROM l_shipdate) as l_year,
        l_extendedprice * (1 - l_discount) as volume
    FROM supplier, lineitem, orders, customer, nation n1, nation n2
    WHERE
        s_suppkey = l_suppkey
        AND o_orderkey = l_orderkey
        AND c_custkey = o_custkey
        AND s_nationkey = n1.n_nationkey
        AND c_nationkey = n2.n_nationkey
        AND (
            (n1.n_name = 'GERMANY' AND n2.n_name = 'RUSSIA')
            OR (n1.n_name = 'RUSSIA' AND n2.n_name = 'GERMANY')
        )
        AND l_shipdate between date '1995-01-01' and date '1996-12-31'
    ) as shipping
GROUP BY supp_nation, cust_nation,
    l_year
ORDER BY supp_nation, cust_nation, l_year;
```

6 P

Aufgabe 5: Sales-Datensatz: Deutsch → SQL

Wir befassen uns weiterhin mit dem Datensatz aus der vorherigen Aufgabe. Nenne für jede der folgenden natürlichsprachlichen Fragen eine geeignete SQL-Anfrage und führe sie auf den Sales-Daten aus. Gib auf deiner Abgabe die Anfrage, die ersten 10 Zeilen des Ergebnisses, sowie die Anzahl an Tupeln des Ergebnisses an.

a) Schreiben Sie eine Query, die für jedes Land *L* und jedes Jahr den Gesamt-Profit aus dem Verkauf aller in diesem Jahr bestellten Teile (Part), dessen Name den String *'chocolate'* enthält, auflistet, wobei die Teile von einem Lieferanten (Supplier) im Land *L* geliefert worden sein müssen. Der Profit eines Teils in einem Verkauf (Lineitem) ist definiert als: (l_extendedprice*(1-l_discount)) - (ps_supplycost * l_quantity). Die Ausgabe soll aufsteigend nach dem Land und innerhalb eines Landes absteigend nach dem Jahr sortiert werden. Zur Selbstkontrole sind hier die ersten 5 Zeilen des Ergebnisses angegeben:

```
ALGERIA,1998,28755250.6829
ALGERIA,1997,51619211.9021
ALGERIA,1996,50971117.0264
ALGERIA,1995,50178326.0903
ALGERIA,1994,50354843.0773
```

8 P

b) Gib für Deutschland ('GERMANY') für alle dort verfügbaren Teile (Part) den Schlüssel, den Namen, sowie die Gesamtanzahl der verfügbaren Teile aus (egal von welchem Supplier, solange er zu Deutschland gehört). Gib jedoch nur die Teile aus, deren Gesamtanzahl mehr als 0.001% der Anzahl aller überhaupt verfügbaren Teile in Deutschland ausmachen. Das Ergebnis soll nach der Gesamtanzahl absteigend sortiert werden. Zur Selbstkontrole sind hier die ersten 5 Zeilen des Ergebnisses angegeben:

```
85606,dodger khaki honeydew lawn mint,26531
60932,peru goldenrod ghost magenta white,25274
80958,tomato white tan drab thistle,22290
139035,salmon navajo cornflower grey maroon,21778
164254,spring ghost orchid saddle beige,21116
```

8 P