University of Salzburg — Instructors: Schäler/Resmerita/Widmoser

Datenbanken 1 – PS (501.073)

Project specification — To be submitted by **01.07.2024 (23:00)** via Blackboard

In this project you are asked to write several SQL queries for a given database, which can be managed by using the provided files:

1. `create.sql` : creates the database
2. `pop.sql` : fills the tables
3. `drop.sql` : deletes the database

Your SQL queries must be written in one file, named `sql_queries.sql`, to be submitted in Blackboard as your answer to the `SQL Projekt` assignment. The queries must be separated by comment lines of the form

```
-- QUERY n
```

with $n = 1, 2, \ldots, 10$, as illustrated in the following example:

```
-- QUERY 1
select nickname
from person
where nickname='unicorn';
-- QUERY 2
select * from text;
```

Note: *The ordering of queries in the file is irrelevant*, you can place for example Query 2 before Query 1.

Your assignment is to write the required SQL statements in the file `sql_queries.sql`. These will be evaluated automatically using a checker program, so **please make sure to use exactly this filename and to number and separate the queries in the files as shown above**.

Evaluation

The output of each query will be compared with our reference output in terms of the layout of attributes, the number of tuples, their values, and their ordering. If the output is correct (in all these criteria), then it is graded with 1 (point), otherwise with 0.

The **10** SQL queries to be implemented for the given database are listed below. The **required output format** for each query is specified in the corresponding box.

1. Retrieve the names and corresponding wineries of all wines that originate from the region 'Bordeaux'.

   ```
   name | weingut
   ```

2. Determine the number of red wines (`farbe`). The resultant attribute must be called `anzahl`.

   ```
   anzahl
   ```

3. Count the number of wines per color. The numbers are determined as values of a new attribute called `anzahl`.

   ```
   anzahl | farbe
   ```

4. List the names (`name`) of all wineries which employ more than one reviewer.

   ```
   weingut
   ```

5. Find the oldest wine(s) and list their names and vintages.

   ```
   name | jahrgang
   ```

6. Retrieve the last and first names of all reviewers that have not yet issued any review.

   ```
   name | vorname
   ```

7. List every review where the reviewer is not also the producer of the reviewed wine. This is the case if the values of the attributes `Wein.weingut` and `Gutachter.weingut` are different.

   ```
   wid | gid | punkte
   ```

8. For every wine, list those reviews which differ by more than 2 points from the average value of all reviews for that wine. A result tuple consists of the wine id `wid`, the review id `gid`, the review points `punkte`, and the average of all review points for the wine as value of a new attribute called `avg`.

   ```
   wid | gid | punkte | avg
   ```

9. List the last name, first name, and title of the reviewer with the longest title. You may assume that there exists only one such reviewer.

   ```
   name | vorname | titel
   ```

10. A blended wine is wine produced from mixing several grape varieties in specific proportions, as specified in the relation `Hergestellt_aus`. Errors may have crept in the database regarding blended wines. List every blended wine whose proportions of individual grape varieties do not add to 100 (percent). Wines that do not occur in the relation `Hergestellt_aus` should not be considered here, as they are assumed to be not blended (i.e., pure). The output schema is `wid | name | anteil`, where `anteil` holds the sum of all proportions.

    ```
    wid | name | anteil
    ```