EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Database Systems 1
Summerterm 2023
Prof. Torsten Grust, Louisa Lambrecht, Tim Fischer
WSI — Database Systems Research Group

Assignment 7

Hand in this assignment until **Thursday, 21 June 2023, 12:00** at the latest.

> ⚠ **Lecture Evaluation**
>
> In the upcoming days, you will have the opportunity to **evaluate lectures** you are attending. With this in mind, we kindly ask you to keep an eye on your inbox and to provide us with **your** valuable feedback. Thank you!

> Ⓔ**xam-style Exercises**
>
> Exercises marked with Ⓔ are similar in style to those you will find in the exam. You can use these to hone your expectations and gauge your skills.

> **Running out of ideas?**
>
> Are you hitting a roadblock? Are some of the exercises unclear? Do you just need that one hint to get the ball rolling? Refer to the `#forum` channel on our Discord server and check the tag for this assignment —maybe you'll find just the help you need.

Task 1: Constraints and References

Answer **briefly**:

(a) What are **foreign key constraints**, **inclusion constraints**, and **referential integrity**?

(b) What are differences between value-based references and pointers in regular programming languages?

(c) List the kinds of database constraints that were discussed in the lecture so far.

Task 2: Cascading Deletes

> **Hint**
>
> Carefully read the entire task before getting started! There may be some nuance required to get this one right. 😉

Hand in a list of SQL commands that create two SQL tables `r` and `s` along with their constraints and instances such that deletions behave as described below. Additionally, include two example `DELETE` statements, one for each table.

Initially, the instances of `r` and `s` should exhibit the following properties:

- No value in a key column may be referenced by more than one value of any foreign key column.
- Each of the two tables must contain at least 5 rows.

Once the tables have been created, deletions in `r` or `s` should show the following behavior:

- If one arbitrary row from `s` is deleted, it must result in both `r` and `s` being empty.
- If one arbitrary row from `r` is deleted, it must result in `r` being empty (while `s` remains unchanged).

Task 3: Existential Quantification (E)

Consider two sets of integers p and q, encoded as follows:

```
CREATE TABLE p ( x int PRIMARY KEY );
CREATE TABLE q ( x int PRIMARY KEY );
```

Based on these, implement SQL queries to simulate the following **set operations**:

(a) set difference $p \setminus q$,

(b) set intersection $p \cap q$, and

(c) the subset relation $p \subseteq q$.

Hint

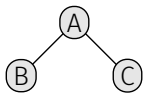
The query for Subtask (c) should ouput a single row with a single column containing either `TRUE` or `FALSE`.

Important!

- These are set operations. As such, the results should not contain duplicates.
- Formulate your queries using predicates (`NOT`) `IN` and (`NOT`) `EXISTS`, and include test cases in your answer.
- Do **not** use the corresponding SQL set operators `INTERSECT`, `EXCEPT` and, `UNION`!

Task 4: Intra-Table Foreign Keys

The lecture discussed a method for encoding tree-like data structures as relations (Slide 12, Chapter 7, "Referential Integrity"). Each node refers to its parent via a foreign key in this encoding. In this task, we will flip the picture by encoding the same relationship as foreign keys that, instead, reference the children of a node.



nodes

label	left	right
A	B	C
B	▢	▢
C	▢	▢

To get started, create the table **nodes** as depicted above; you may assume that every node will have at most two children. Be sure to add the appropriate constraints! Afterwards, encode the tree structure from Slide 12 and store it in your table (the resulting table will hold six rows).

Next, formulate SQL queries on your tree representation to answer the following questions. The resulting schemas of your queries should follow the depictions on the right.

(a) What is the label of the sibling of the node with label `'E'`?

sibling
...

Hint

The output should be empty, if the node with label `'E'` has no sibling or does not exist.

(b) What are the labels of the grandchildren of the node with label `'A'`?

grandchildren
...

Hint

The output should be empty, if the node with label `'A'` does not exist.

(c) What is the label of the root node?

root
...

Hint

The output should be empty, if the tree is empty.