EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

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

Assignment 5

Hand in this assignment until **Thursday, 8 June 2023, 12:00** at the latest.

Ⓔ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 (1 credit)

Consider the following SQL DDL statement which creates a table to hold a company's employees:

```
CREATE TABLE employees (
  employee_id   int,
  lastname      text,
  firstname     text,
  address       text,
  hire_date     date,
  salary        salary,   -- monthly salary (in €)
  emp_role      emp_role, -- employee role
  department_id int       -- identifier of employee's department
);
```

Assume the mini-world abides by the following rules:

1. No employee detail may be left unspecified.
2. The `salary` of all employees must not be less than 1,473.33 € a month.
3. An employee's role is either 'Manager', 'Developer', 'Accountant' or 'Secretary'.
4. Managers hired after November 24, 2013 have a monthly `salary` of at most 17,679.96 €.
5. No two employees must share the same identifier (`employee_id`).

Subtasks:

(a) For each rule, use SQL DDL statements to define constraints to enforce the rule as well as one `INSERT`-statement that abides by the rule and one example that violates the rule. Create types for `salary` and `emp_role`.

(b) Please explain why it is impossible to enforce...

i. ... rule 4 using a domain constraint (`CREATE DOMAIN`),

ii. ... rule 5 in terms of a `CHECK` constaint.

Task 2: Defining Keys (1 credit)

Please define the terms “candidate key”, “superkey” and “primary key” in your own words as precisely as possible.

Task 3: Identifying Keys Ⓔ (1 credit)

Based on the instance of a table `T(A, B, C, D)` depicted to the right:

(a) … list all possible *candidate keys*.

(b) … list all superkeys of this table.

T

A	B	C	D
1	1	9	11
1	2	8	12
2	2	7	13
3	4	4	11
4	4	5	12
5	5	6	13

Task 4: Using Keys (1 credit)

Consider the following schema definition and constraints for table **r**:

```
CREATE TABLE r (a int, b varchar(9999), c int, d int, e text);

ALTER TABLE r ALTER COLUMN a SET NOT NULL;
ALTER TABLE r ALTER COLUMN b SET NOT NULL;
ALTER TABLE r ALTER COLUMN c SET NOT NULL;

ALTER TABLE r ADD UNIQUE (a, c);
ALTER TABLE r ADD UNIQUE (b);
ALTER TABLE r ADD UNIQUE (d);
```

(a) Ⓔ Your task is to choose a primary key for **r** under the assumption that neither **a** nor **c** are unique on their own. Which (combinations of) columns are eligible for primary key? Which primary key would you choose? Why?

(b) Provide the required *SQL* DDL commands to declare the primary key you chose.

(c) For one reason or another you are not satisfied with your choice of primary key in (a) above. Your co-worker thus proposes to add an **artificial primary key column id** to table **r**.

```
-- PostgreSQL assigns <table name>_pkey as the primary key constraint name,
   thus:
ALTER TABLE r DROP CONSTRAINT r_pkey;

ALTER TABLE r ADD COLUMN id int;
ALTER TABLE r ADD PRIMARY KEY (id);
```

What advantages and disadvantages does the new column **id** bring with it? Discuss briefly.

(d) Ⓔ Consider the following query:

```
SELECT DISTINCT v.a, v.b, v.e FROM r AS v;
```

Is it necessary to use `DISTINCT` here? Justify your answer.