Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Datenbanksysteme · Prof. Dr. Grust

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Datenbanksysteme I

WS 2019/20
Torsten Grust, Christian Duta

Assignment #9

Submission Deadline: January 7, 2020 - 10:00

Please note that the response time in our forum will likely slow down during the semester break. Assignment #10 will be published on January 07, 2020. **We wish you a merry Christmas and a happy new year!**

Exercise 1: `GROUP BY` and `HAVING` (8 Points)

1. Consider this schema

```
CREATE TABLE r(a int, b int);
```

and the following two queries[1]:

```
SELECT r.a, COUNT(*) AS c
FROM r r
WHERE r.b = 4
GROUP BY r.a;
```

```
SELECT r.a, COUNT(*) AS c
FROM r r
GROUP BY r.a
HAVING EVERY(r.b = 4);
```

Create an instance for `r` for which the different behavior of these queries can be observed in the queries' result.
Note: The `EVERY` aggregate function is synonymous to `bool_and` which was previously discussed in the lecture.

2. Consider the following query:

```
SELECT r.a, MAX(r.b) AS m
FROM r r
GROUP BY r.a
HAVING COUNT(*) > 2;
```

Write an equivalent query without `HAVING`.

[1] Various aggregate functions are explained at http://www.postgresql.org/docs/12/static/functions-aggregate.html.

3. On the university database[2], compute all names of MSc students taking more than 3 courses, together with their actual workload (number of courses taken).

```
result(name, workload)
```

4. Another task on the university database: Find the names of all staff members whose courses are attended by less than five students or whose courses are entirely unattended. Assume that staff member names are unique. One student enrolled in more than one course of the same staff member *each* counts towards the total attendance. We propose the following faulty solution:

```
SELECT s.name
FROM enrolled e, class c, staff s
WHERE e.classid = c.classid and c.staffid = s.staffid
GROUP BY s.name
HAVING COUNT(*) < 5;
```

Run this query and explain why the query is incorrect. Suggest a correct alternative.

Exercise 2: Boyce-Codd Normal Form (7 Points)

Table `delivery` stores information about the deliveries of a logistics company to their clients, executed by differing drivers using differing vehicles. Clients are assigned a unique identifier and their address is stored. For each delivery, the table stores date, time and the delivery status, as well as identifier and name of the driver and the designation of the vehicle. A driver does not change the vehicle during the course of a day. A client will receive only one delivery per day, but a driver usually delivers to multiple clients per day.

delivery

clientid	address	date	time	status	driverid	drivername	vehicle
17	Sand 13, 72076 Tü	19.08.08	17:00	OK	4711	D. Unlop	MB 104
17	Sand 13, 72076 Tü	14.10.08	09:00	keiner da	1508	M. Ichelin	FT 8
17	Sand 13, 72076 Tü	29.10.08	11:00	abgelehnt	1508	M. Ichelin	MAN 37
19	Wilhelmstr. 5, 72076 Tü	02.04.08	08:30	OK	1508	M. Ichelin	FT 8
19	Wilhelmstr. 5, 72076 Tü	10.05.08	14:30	Reklamation	4711	D. Unlop	MAN 37
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The following functional dependencies hold:

$$\begin{aligned}
\text{clientid, date} &\rightarrow \text{address, time, status, driverid, drivername, vehicle} \\
\text{date, time, vehicle} &\rightarrow \text{clientid, address, status, driverid, drivername} \\
\text{date, time, driverid} &\rightarrow \text{clientid, address, status, drivername, vehicle} \\
\text{clientid} &\rightarrow \text{address} \\
\text{driverid} &\rightarrow \text{drivername} \\
\text{driverid, date} &\rightarrow \text{vehicle}
\end{aligned}$$

This table in 1NF obviously contains quite a lot of redundancy, leading to anomalies. Construct an equivalent schema in Boyce-Codd Normal Form using the decomposition algorithm (slide 25, slide set 10 "Functional Dependencies"). Hand in the schemata of all relations R_1, R_2 generated in step 1 of function `split` as well as the final database schema.

[2] The university database schema and instance were introduced in Assignment #6 and provided to you again as `uni.zip`.