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

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

# Datenbanksysteme I

WS 2021/22
Torsten Grust, Christian Duta, Tim Fischer

# Assignment #8

Submission Deadline: December 22, 2021 - 10:00

> Please note that students currently have the opportunity to **evaluate lectures**. Please help us to improve **your** courses by providing precious feedback. Check your Mailbox now and participate **today**.

## Exercise 1: Aggregate Queries (6 Points)

Again, we use the now familiar university schema and data from assignment #6. In case you miss any schemata or instances, we provided you with an archive `uni.zip` containing the university data from assignment #6 again.

Please solve the following tasks by formulating SQL queries. Make sure that your result tables match the **result** schemas.

1. Calculate the average age of students for each type of degree, i.e. BSc and MSc students.
   **Output:** a table with two columns `pursued_degree` of type `text` and `average_age` of type `numeric`.
2. List the number of different major subjects for which students are registered.
   **Output:** a table with one column `subject_count` of type `bigint`.
3. Find the oldest student(s).
   **Output:** a table with two columns `student_name` of type `text` and `age` of type `int`.
4. Determine the number of classes in which each student is enrolled. Hand in **two queries**. The result of the first query includes students not enrolled in any classes and one where these students are not included.
   **Output:** a table with two columns `student_name` of type `text` and `classes_count` of type `bigint`.

## Exercise 2: From $NF^2$ to 1NF (12 Points)

The table in Figure 1 contains information about the departments of a company, including each department's contacts as well as its employees and their tasks.[1] Boolean column `client` indicates whether a contact is a client (as opposed to a staff member, for example). The table is given in Non-First Normal Form ($NF^2$).

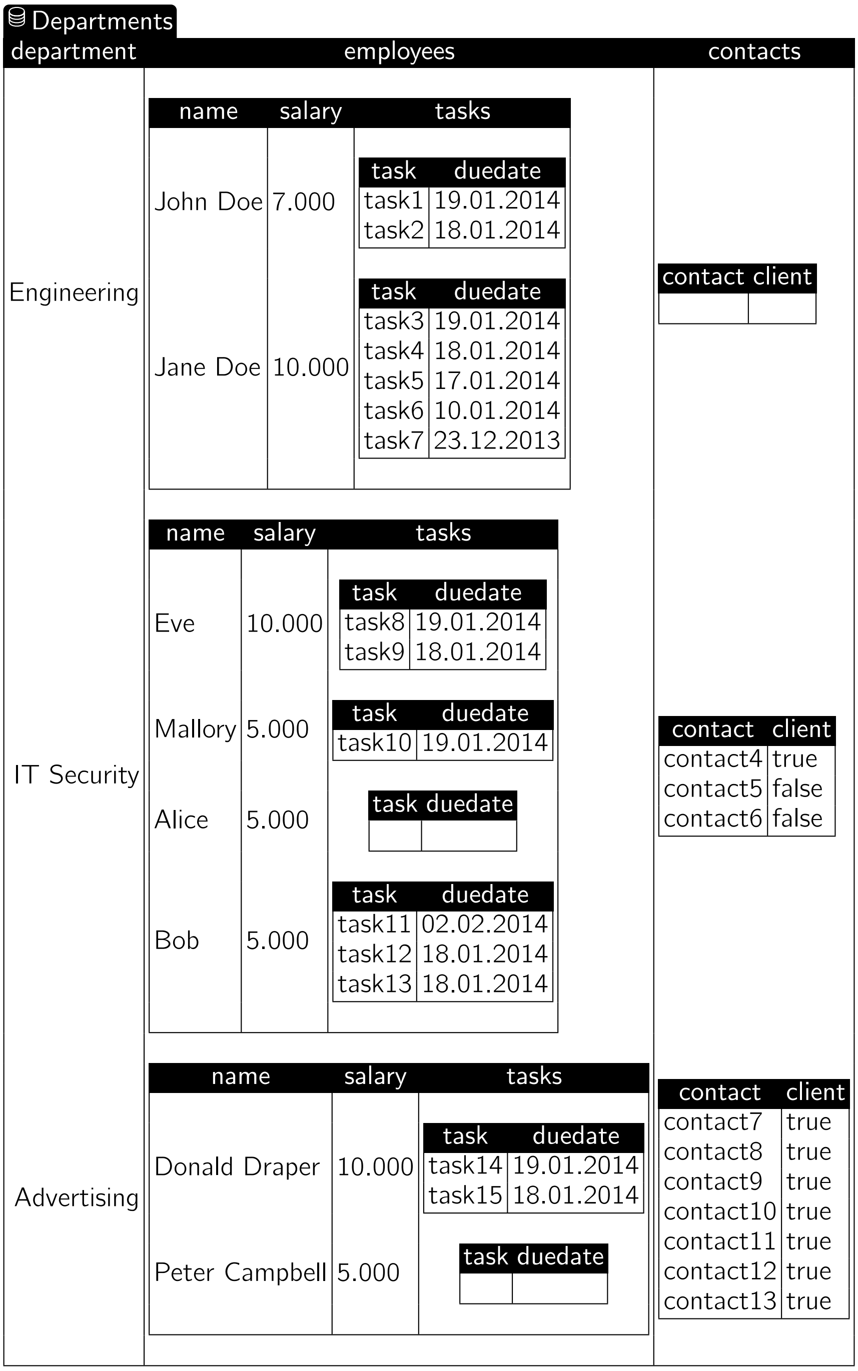
1. Transform the $NF^2$ schema into an equivalent 1NF database schema using algorithm `nf2to1nf()` from slide 10, slide set 8 "Database Design". Then, formulate suitable SQLDDL statements to create the resulting flat tables.
2. Provide SQL DML statements that populate the flat tables such that they provide the same information as the $NF^2$ table in Figure 1.
3. For each of the following tasks, formulate a single SQL query on your 1NF representation:
   (a) Compute the number of employees per department.
   **Output:** a table with two columns `department` of type `text` and `employees_count` of type `bigint`.
   (b) Find departments without contacts.
   **Output:** a table with one column `department` of type `text`.
   (c) Return the names and department of employees without tasks.
   **Output:** a table with two columns `name` and `department` both of type `text`.
   (d) Compute the number of clients per department.
   **Output:** a table with two columns `department` of type `text` and `clients_count` of type `bigint`.
   **Note**: Obviously, some departments don't have any clients. Nevertheless, they should be included in the result with their correct number of clients. Think about using a *correlated subquery*.

## Exercise 3: LEGO Data Warehouse (12 Points)

We provided you with a dataset that you already know from the lecture: The LEGO data warehouse `legodw.sql`. Load the file into a PostgreSQL database. Then, formulate queries for the following tasks:

1. For each store in Germany, compute the turnover per day of the week.
   **Output:** a table with four columns `store`, `city`, `dow` and `turnover` with each of type `int`, `text`, `int` and `money` respectively.
2. For each store in Germany, compute the day(s) with the highest turnover.
   **Output:** a table with four columns `store`, `city`, `dow` and `turnover` with each of type `int`, `text`, `int` and `money` respectively.
3. Compute the most popular set(s) per country, i.e. those sets that have had the most sold *items*.
   **Output:** a table with three columns `country`, `set` and `items` with each of type `text`, `id` and `bigint` respectively.

[1] Due to the widespread use of employee-department schemas in database literature, it has been called the *Drosophila melanogaster* of database systems research.



Departments

| department | employees | | | | contacts | |
|---|---|---|---|---|---|---|
| | name | salary | tasks | | contact | client |
| | | | task | duedate | | |
| Engineering | John Doe | 7.000 | task1 | 19.01.2014 | | |
| | | | task2 | 18.01.2014 | | |
| | Jane Doe | 10.000 | task3 | 19.01.2014 | | |
| | | | task4 | 18.01.2014 | | |
| | | | task5 | 17.01.2014 | | |
| | | | task6 | 10.01.2014 | | |
| | | | task7 | 23.12.2013 | | |
| IT Security | Eve | 10.000 | task8 | 19.01.2014 | contact4 | true |
| | | | task9 | 18.01.2014 | contact5 | false |
| | Mallory | 5.000 | task10 | 19.01.2014 | contact6 | false |
| | Alice | 5.000 | | | | |
| | Bob | 5.000 | task11 | 02.02.2014 | | |
| | | | task12 | 18.01.2014 | | |
| | | | task13 | 18.01.2014 | | |
| Advertising | Donald Draper | 10.000 | task14 | 19.01.2014 | contact7 | true |
| | | | task15 | 18.01.2014 | contact8 | true |
| | Peter Campbell | 5.000 | | | contact9 | true |
| | | | | | contact10 | true |
| | | | | | contact11 | true |
| | | | | | contact12 | true |
| | | | | | contact13 | true |

Figure 1: Table `Departments` in NF$^2$