EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Advanced SQL
Winter Term 2025/2026
Prof. Torsten Grust, Tim Fischer, Björn Bamberg
WSI — Database Systems Research Group

Assignment 3

Hand in this assignment until **Friday, November 14th 2025, 12:00 pm** at the latest.

> 🤔 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—maybe you'll find just the help you need.

> Ⓔ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.

Task 1: So similar, yet so different Ⓔ

Create an instance for table `r` with schema `r(a int, b int)` such that the two queries `Q1` and `Q2` compute **different** results.

Query `Q1`: `WHERE`-clause

```
SELECT   r.a, count(*) AS c
FROM     r AS r
WHERE    r.b <> 3
GROUP BY r.a;
```

Query `Q2`: `HAVING`-clause

```
SELECT   r.a, count(*) AS c
FROM     r AS r
GROUP BY r.a
HAVING   bool_and(r.b <> 3);
```

> Note
> Query `Q2` uses the aggregate function `bool_and`. Read about it in the DuckDB documentation 🌐.

Task 2: Production Steps Ⓔ

Consider table `production` which tracks the progress of items currently in production. The production process of these items occurs in multiple steps. If a step has been *completed* for an item, we store the completion time and date. If the step is still *in progress*, we write `NULL` into column `completion` holds `NULL`.

```
CREATE TABLE production (
  item       char(20) NOT NULL,
  step       int      NOT NULL,
  completion timestamp, -- NULL ≡ incomplete
  PRIMARY KEY (item, step)
);
```

To the right, you can see an example instance of `production`. Notice that step 2 of `AT-AT` production is not complete yet. Similarly for `DS II` production, step 1 is still pending, but steps 2 and 3 are already completed.

item	step	completion
TIE	1	1977-03-02 04:12:00
TIE	2	1977-12-29 05:55:00
AT-AT	1	1978-01-03 14:12:00
AT-AT	2	NULL
DSII	1	NULL
DSII	2	1979-05-26 20:05:00
DSII	3	1979-04-04 17:12:00

A **Formulate a SQL query** that lists the names of all items for which all production steps are complete. Avoid duplicate item names. For the example instance to the right, we expect the following result:

complete
TIE

Task 3: Measurements

You are handed a note by a physicist, who kindly asks you to take a look at the measurements $m(t)$ (taken at time t) they intend to store inside a RDBMS of your choice (DuckDB, of course). They explain that at each timestamp t, one or more measurements $m(t)$ have been recorded.

Measurements Note

t	$m(t)$
1.0	1.0, 3.0, 5.0, 5.0
2.5	0.8, 2.0
4.0	0.5
5.5	3.0
8.0	2.0, 6.0, 8.0
10.5	1.0, 3.0, 8.0

Write SQL queries based on the measurements table (provided in `measurements.sql`) for the following tasks. Assume that, in the future, more measurements may be recorded.

A **Compute a *one-column* table** with the global maximum of the measurements $m(t)$.

B **Compute a *two-column* table** that lists each time t and the average of its measurements $m(t)$.

C **Compute a *two-column* table** that lists the average of all measurements $m(t)$ in each timeframe $[0.0 - 5.0)$, $[5.0 - 10.0)$, $[10.0 - 15.0)$, … . Be aware that there might be more measurements in different timeframes in the future.

D **Find all times t (there may be more than one)** at which the global maximum was recorded. The result is a *two-column* table that lists time t and the global maximum.