EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

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

Assignment 2

Hand in this assignment until **Friday, November 7th 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: Matrix Multiplication Ⓔ

You are given two tables representing two matrices A and B of sizes $m \times n$ and $n \times p$, respectively. We create those tables with three columns, where `row` represents the row index and `col` represents the column index of the matrix. The column `val` represents the value of a matrix element.

```sql
CREATE TABLE A (
  row int,
  col int,
  val int,
  PRIMARY KEY(row, col));
```

```sql
-- does not copy keys, contraints etc.
CREATE TABLE B AS TABLE A LIMIT 0;
-- add keys/constraints manually:
ALTER TABLE B
ADD PRIMARY KEY (row, col);
```

A **Formulate a SQL query**, which performs matrix multiplication $A \cdot B$.

🔢 Example: *Relational* Matrices

Consider the following two matrices and their product:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 5 & 4 & 5 \\ 11 & 10 & 11 \end{pmatrix}$$

To the right, you see the equivalent computation based on our tabular matrix representation.

A

row	col	val
1	1	1
1	2	2
2	1	3
2	2	4

·

B

row	col	val
1	1	1
1	2	2
1	3	1
2	1	2
2	2	1
2	3	2

=

$A \cdot B$

row	col	val
1	1	5
1	2	4
1	3	5
2	1	11
2	2	10
2	3	11

B We can also use a *sparse encoding* for our matrices, *i.e.*, we skip all entries with a value of 0. **Does your SQL query of Subtask A also work on these sparse matrices?** Explain your answer, briefly.

🔢 Example: *Sparse* Relational Matrices

Consider the following two matrices and their product:

$$A = \begin{pmatrix} 1 & 3 & 0 \\ 0 & 0 & 7 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 8 \\ 1 & 1 & 10 \\ 3 & 6 & 0 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 4 & 3 & 38 \\ 21 & 42 & 0 \end{pmatrix}$$

To the right, you see the equivalent computation based on our tabular *sparse* matrix representation.

A

row	col	val
1	1	1
1	2	3
2	3	7

·

B

row	col	val
1	1	1
1	3	8
2	1	1
2	2	1
2	3	10
3	1	3
3	2	6

=

$A \cdot B$

row	col	val
1	1	4
1	2	3
1	3	38
2	1	21
2	2	42

Task 2: King and Knight

Familiarize yourself with the *incomplete* SQL file `chess.sql`. In it, we define a chess board (see table `board`) on which we can only place kings (♚, ♔) and knights (♞, ♘). Recall the valid chess moves from the last assignment[1].

In this task you will be completing this SQL file and once you've done so running it should yield the output shown to the right. The output depicts the current state of the chess board and all possible moves (marked with `*`) for all pieces on the board. Try playing around with the `board` table to check out different arrangements of pieces.

```
 |A|B|C|D|E|F|G|H
8| | | | | | | | 
7| | | |*|*|*| | 
6| | |*|*|♔|*| | 
5| |*| |*|*|*| | 
4| | | |♞| | | | 
3| |*| | | |*| | 
2| | |*| |*| | | 
1| | | | | | | | 
```

⚠ Don't be lazy!
Your completed SQL file has to work for *all possible arrangements*! So be sure to poke around and see if your code fails. 😉

A Complete the `INSERT` statement for table `board_and_pieces` by replacing the `YOUR QUERY HERE` comment with your solution.

💡 Hint
If done correctly, this table should contain the one entry for each legal board position.

B Complete the `INSERT` statement for table `possible_movement` by replacing the `YOUR QUERY HERE` comment with your solution.

💡 Hint
If done correctly, this table should contain the one entry for each legal move of any piece given the current arrangements in `board_and_pieces`.

🔠 Having character encoding issues?
As with the last assignment: if you run into problem with the unicode characters for chess pieces, simply replace them as follows: ♔ with `'k'`, ♚ with `'K'`, ♘ with `'n'` and ♞ with `'N'`.

[1]For details about chess, see https://en.wikipedia.org/wiki/Chess 🌐.