

Modern Database Concepts

Prof. Dr. Florian Heinz

florian.heinz@oth-regensburg.de

Exercise Sheet 5OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Exercise 1: Arrays

Open DBeaver and connect to the OTH PostgreSQL database instance (or any other PostgreSQL-Server with version 13+)

Create a database schema **ex5** to work with. You can do that either in the context menu of the database node (your OTH-Username) at the left side or by opening an SQL console and using the `CREATE SCHEMA ... SQL` command.

Load and import the file `people_array.sql`, which you can download from GRIPS

Perform the following queries:

1. For each person, find out the last city this person visited.
2. Find out the people, whose last visited city is Berlin.
3. Try to create an index to speed up the previous query. You can see if the index is used by prefixing the query with

```
EXPLAIN ANALYZE ...
```

. But first, prohibit the database to use a sequential scan if an index is available by stating `SET enable_seqscan = off`. Hint: You might try to use a functional index.

4. Create a query to find out all people who visited either **Wien** or **Budapest** and analyze, how long this query takes. Then, create a GIN index on the field and recheck the execution time.
5. Update the table to add **Regensburg** as the *second* visited city to all arrays (insert the element, do not overwrite an existing one)

Exercise 2: UDT

1. Create a data type (domain) **pos** (for position) that is based on integer, cannot be null and contains a value between 1 and 20
2. Create a data type (row) **waypoint** that can take a city name and a position
3. Create a table containing rows with a person's name and a waypoint.
4. Now insert rows into this table by selecting them from the **people** table (use the `INSERT INTO tablename (SELECT ...)` syntax). Try to formulate the `SELECT` from `people` first, which should yield pairs of **name** and **waypoint** (use `unnest` with `ordinality`). If that works, insert it into the new table.

Modern Database Concepts

Prof. Dr. Florian Heinz

florian.heinz@oth-regensburg.de

Exercise Sheet 5OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Exercise 3: XML

1. Load and import the file `pokemon_xml.sql` from GRIPS
2. Use an xpath expression to get the name together with the ATK stat value from the `pokemon` table
3. Use an xpath expression in the **WHERE** clause to find out, which pokemon have a sum of their baseStats greater than 400. For constructing the expression you might want to develop it on the left side of the FROM clause first before using it as a predicate on the right side.
4. Construct a query to retrieve the name, dex-number and all ability-values of all pokemon. Each ability should be in an own row, so a pokemon with two abilities occurs twice in the result. Use the `xmltable`-function for this. Hint: the first string in the `xmltable` (the row expression) has to be chosen in a way, that all path-specifications in the columns only yield a single value.
5. Now use the previous query and aggregate all abilities into an array. Use the **array_agg** aggregation function to achieve this.