

# Advanced SQL queries

D'Ambrosi Denis

October 16, 2023

## Exercise 1

```
1 SELECT XMLELEMENT(  
2     NAME photos,  
3     XMLAGG(nester.nested)  
4 ) FROM (  
5     SELECT XMLELEMENT(  
6         NAME photo,  
7         XMLATTRIBUTES(url, titel AS title, photo.  
8             personemail AS owner),  
9         XMLAGG(  
10             XMLELEMENT(  
11                 NAME shown,  
12                 XMLATTRIBUTES(vorname AS firstname,  
13                     nachname AS lastname))  
14             )  
15         ) AS nested  
16     FROM photo, istabgebildet, person  
17     WHERE istabgebildet.photourl = photo.url AND  
18         istabgebildet.personemail = person.email  
19     GROUP BY photo.url  
20 ) AS nester;
```

## Exercise 2

```
1 SELECT JSON_BUILD_OBJECT('photos', JSON_AGG(nested))
2 FROM (
3     SELECT url, titel AS title,
4            photo.personemail AS owner,
5            JSON_AGG(
6                JSON_BUILD_OBJECT('firstname', person.
7                                   nachname, 'lastname', person.vorname)
8            ) AS shown
9     FROM photo, istabgebildet, person
10    WHERE istabgebildet.photourl = photo.url AND
11           istabgebildet.personemail = person.email
12    GROUP BY photo.url
13 ) AS nested;
```

### Exercise 3

```
1 CREATE TABLE myFiles (  
2     key varchar(255) PRIMARY KEY,  
3     content jsonb  
4 );  
5  
6 INSERT INTO myFiles (key, content)  
7 SELECT 'photos' AS key, JSONB_BUILD_OBJECT('photos',  
8     JSON_AGG(photos)) AS content  
9 FROM (  
10     SELECT url, titel AS title,  
11         photo.personemail AS owner,  
12         JSON_AGG(  
13             JSON_BUILD_OBJECT('firstname', person.  
14                 nachname, 'lastname', person.vorname)  
15         ) AS shown  
16     FROM photo, istabgebildet, person  
17     WHERE istabgebildet.photourl = photo.url AND  
18         istabgebildet.personemail = person.email  
19     GROUP BY photo.url  
20 ) AS photos;  
21  
22 SELECT photo->>'url' AS url, photo->>'owner' AS owner  
23 FROM (  
24     SELECT JSONB_ARRAY_ELEMENTS(content->'photos') AS  
25         photo  
26     FROM myFiles  
27     WHERE key = 'photos'  
28 )  
29 WHERE photo->'shown' @> '[{ "lastname": "Lea", "firstname  
30     ": "Meyer" }]'::jsonb;
```

## Exercise 4

```
1 SELECT name, "numEmpl", MAX("numEmpl") OVER (PARTITION BY  
   "parentId") AS "maxNumEmpl"  
2 FROM department;
```

## Exercise 5

```
1 WITH RECURSIVE dep AS (  
2     SELECT *  
3     FROM department  
4     WHERE "parentId" = 4 OR "deptId" = 4  
5     UNION  
6     SELECT department.*  
7     FROM department, dep  
8     WHERE department."parentId" = dep."deptId"  
9 )  
10 SELECT *  
11 FROM dep;  
12  
13 SELECT "deptId", name, MAX("totalEmpl")  
14 FROM (  
15     WITH RECURSIVE dep AS (  
16         SELECT *  
17         FROM department  
18         UNION ALL  
19         SELECT department."deptId", department.name,  
20             dep."parentId", department."numEmpl"  
21         FROM department  
22         INNER JOIN dep ON department."parentId" = dep  
23             ."deptId"  
24         WHERE department."parentId" <> department."  
25             deptId"  
26     )  
27     SELECT department."deptId",  
28         department.name,  
29         d."totalEmpl" + department."numEmpl" AS "  
30             totalEmpl"  
31     FROM department  
32     INNER JOIN (  
33         SELECT "parentId", SUM("numEmpl") AS "totalEmpl"  
34         FROM dep  
35         GROUP BY "parentId"  
36     ) AS d  
37     ON department."deptId" = d."parentId"  
38     UNION  
39     SELECT "deptId", name, "numEmpl"  
40     FROM department  
41 )  
42 GROUP BY "deptId", name;
```