

**1. (5 points)**

Explain windows (partitions and frames) in the context of the SQL language. Explain the relationship between partitions and frames, and the basic differences between them.

Give an example of an SQL query which will demonstrate the use of partitions and frames. Ensure that the problem in the example could not be solved without the use of windows and functions for working with windows (eg. using GROUP BY).

Show the content of the relation(s) used in the query, and the query result.

**2. (6 points) This assignment assumes that SQL standard is used.**

The database stores information about employees and projects in an enterprise. SQL statements shown below create types **projectT** and **personT**. Also, you can assume that typed tables **projectOR** and **personOR** (based on these types) are created and populated and that their unique identifier column names are **OIDproject** or **OIDperson**. The attribute **personT.projects** stores the projects in which the person participated.

```
CREATE TYPE projectT AS (  
  idProject INTEGER,  
  projectName CHAR(50))  
INSTANTIABLE NOT FINAL  
REF IS SYSTEM GENERATED;
```

```
CREATE TYPE personT AS (  
  idPerson INTEGER,  
  fname CHAR(50),  
  lname CHAR(75),  
  projects REF(projectT) MULTISSET)  
INSTANTIABLE NOT FINAL  
REF IS SYSTEM GENERATED;
```

a) Write a SQL statement that will give the project name and number of employees who worked only on that project but only for projects that contain the text "Croatian Post" in their name. Solve the task without the use of subqueries.

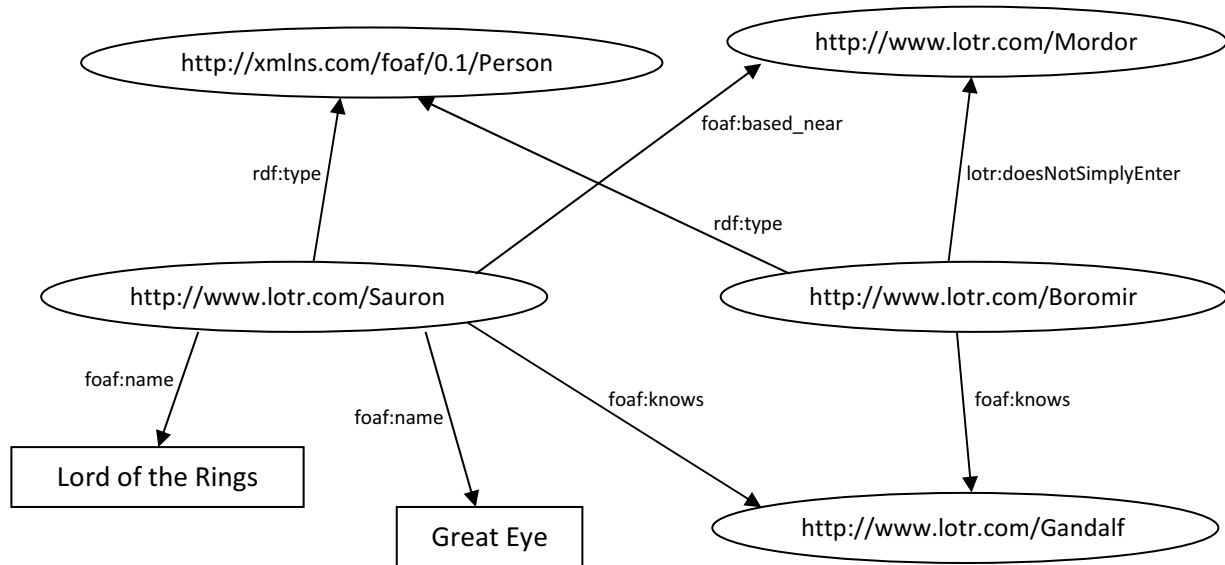
b) Table **personOR** contains a record of the employee with code 3 (**idPerson**), who has not worked on any project. Write an SQL statement that will be update the fact that the employee worked on the project with code 2 (**idProject**).

**3. (3 points)** Provide different ways in which we can define the notion of time and time-data in database management systems. What is a **Chronon**?**4. (3 points)** Explain the quorum mechanism for achieving consistency. Give examples, both for reading and writing data.**5. (5 points)** What is a **combinable reducer**? Explain using an example.

6. (4 points) Write all triplets that comprise the following graph:

Note: for the sake of simplicity, you can use Notation3 (N-Triples, Turtle) format and the following prefixes:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix lotr: <http://www.lotr.com/> .
```



7. (4 boda) Complete the following SPARQL query of semantic source DBpedia:

Assume use of the following prefixes:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
```

Fetch the English ("en") name and number of inhabitants of all the cities in Croatia whose names contain the letter "S".

Sort the results by the number of inhabitants (from larger to smaller).

```

SELECT _____
WHERE {
  ?grad a dbpedia-owl:City .
  ?grad dbpedia-owl:country <http://dbpedia.org/resource/Croatia> .
  ?grad rdfs:label ?naziv .
  ?grad dbpprop:population ?brojSt

  _____
  _____
}
_____
```

## Solutions:

### 2. a) (4 boda)

```
SELECT p.OIDproject -> projectName, COUNT(*)
  FROM personOR o, UNNEST (o.projects) AS p(OIDproject)
 WHERE p.OIDproject -> projectName LIKE '%Croatian Post%'
    AND CARDINALITY (o.projects) = 1
 GROUP BY p.OIDprojekt, p.OIDproject -> projectName;
```

### b) (2 boda)

```
UPDATE personOR SET projects = MULTiset(SELECT p.OIDproject FROM p.projectOR
                                         WHERE p.idProject = 2)
 WHERE idPerson = 3;
```

## 6. (4 boda)

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix lotr: <http://www.lotr.com/> .
<http://www.lotr.com/Sauron>
    a foaf:Person ;
    foaf:name "Lord of the Rings" ;
    foaf:name "Great Eye" ;
    foaf:knows <http://www.lotr.com/Gandalf> ;
    foaf:based_near <http://www.lotr.com/Mordor> .
<http://www.lotr.com/Boromir>
    a foaf:Person ;
    foaf:knows <http://www.lotr.com/Gandalf> ;
    lotr:doesNotSimplyEnter <http://www.lotr.com/Mordor> .
```

## 7. (4 boda)

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
SELECT ?naziv ?brojSt
WHERE {
    ?grad a dbpedia-owl:City .
    ?grad dbpedia-owl:country <http://dbpedia.org/resource/Croatia> .
    ?grad rdfs:label ?naziv .
    ?grad dbpprop:populationTotal ?brojSt
    FILTER(LANG(?naziv) = "en")
    FILTER(REGEX(?naziv, "S"))
}
ORDER BY DESC(?brojSt)
```