1 – Some SQL

1.1 Triggers

Assume that we have a modern people´s registry

preg (personID, name, lastname)

and a modern marriage registry

mreg (mID, pID, prevLastname, newLastname, mDate)

where

- the **mDate** is a date/time attribute showing the marriage date and when the last name is changed (if it is changed at all), and
- the **pID** in **mreg** is the ID of the person for which the marriage and the change is registered, referring to the **personID** in **preg**.

A person can be married and re-married after a divorce many times, and the modern rules accept last name change for both male and female (so you don´t have to worry about the gender, which is not in the relations here).

Create a trigger that checks whether the new and the old last names match before an update to the **preg**, registers (logs) the old last name and the new last name for that specific marriage and person in the marriage register **mreg** and updates the people´s registry with the new last name.

1.2 Arrays

How can you remove the additional **mreg** in part 1.1 and create instead a **preg** that has a list of last names and their corresponding marriage or change dates for each new last name?

Write the new relation and explain briefly how it would function.

2 – Indices

This is a short question with a long answer but try to keep it as brief as possible and use your own words in the explanations:

List up the major categories and types of indices that we have seen and explain their respective advantages and disadvantages, as well as their uses.

3 – Relational Algebra & Query Compilation

3.1 Query Processing Overview

We went through the steps of query processing as an introduction to query compilation. Sketch the steps with the inputs/outputs for each step and the relations between them, and describe their role/function briefly in your own words.

3.2 Relational Algebra Expressions

Assume two relations **R1(A, B, C, D, E)** and **R2(F, G, H)**. Assume that attributes E and F are of the same type, that attribute A is a character or string (variable or fixed) and that attribute H is a numeric value.

Write the equivalent of the following SELECT sentence in relational algebraic terms.

```
SELECT B, C, G
FROM R1, R2
WHERE E = F AND A = ´a´ AND H > 1
```

3.3 Estimating Size (Logical Query Plans)

3.3.a Draw the relational algebraic tree directly from the relational algebraic expression you wrote in 3.2.

3.3.b Draw a second alternative relational algebraic tree of the same relational algebraic expression in 3.2, improving (optimizing) upon the first one by re-ordering the hierarchy of operations on the tree.

Note that we are looking for your understanding of basic optimization rules, so make sure that the trees in 3.3.a and 3.3.b are different and that there is room for improvement from one to the other.

3.3.c Compare the two, explaining the improvement criteria, what you did and why, and explaining or demonstrating the improvement. You may assume that both relations have 10 tuples (rows). You can sketch for the purpose of explaining if you need to.

4 – DBMS Architecture

Sketch the typical DBMS architecture we saw a few times during the beginning of the course and explain the role and job of each path and component of the architecture.

5 – Logs and their Uses

5.1 Log Functions

For what purposes do we use database logs? Use your own words to explain.

5.2 Log Types

List the log types we have seen during the semester, explaining their uses and the differences between them.

5.3 Understanding the Workings of a Log

You listed and explained the components of a typical DBMS architecture. Think of that architecture.

5.3.a Where does the log reside?

5.3.b Which components of a DBMS are involved in managing a log?

6 – Concurrency Control & Serialization

6.1 The Concepts

Define and explain the concepts serial, serializable and conflict serializable.

6.2 Conflicts in Execution Plans

List, explain and exemplify the three types of conflicts in execution plans. Note that exemplify means give examples (at least one for each).

6.3 Conflict Serializable vs. Serializable

We have seen and discussed the following execution plan in the lectures:

S = $r_1(A)$; $w_1(A)$; $r_2(A)$; $w_2(A)$; $r_2(B)$; $w_2(B)$; $r_1(B)$; $w_1(B)$

6.3.a What does the serializability of this execution plan (or other similar execution plans) depend upon? Explain and demonstrate briefly.

6.3.b Pick the non-serializable case. Show explicitly what you have picked and why. Hint: refer to the previous question in 6.3.a.

Show all the conflicts in the execution plan also stating what kind of conflict the conflict is.

7 – Isolation Levels

7.1 Snapshot Isolation

Snapshot Isolation is a very popular technique that we have used some time on.

Define and then describe in your own words what Snapshot Isolation is and what it is used for. Why is it so popular? Does it guarantee serializability?

7.2 SI Plans

7.2.a What is an SI plan? What does it mean that an SI plan is strict?

For the next two questions: We have not talked about these explicitly, but it is implicit in the definition of an SI plan, and you should be able to answer these using simple reasoning.

7.2.b Does an SI Plan help manage concurrency? How?

7.2.b Would you say that an SI Plan opens up for possible deadlocks, or that it helps avoid deadlocks?

8 – Map-Reduce

The following table T1 stores information about shops, the products they sell and the available amount for each product. The table has three columns (shop, product, amount).

T1

shop	product	amount
s1	p1	3
s1	p2	2
s1	p3	1
s1	p4	0
s2	p1	4
s2	p2	5
s2	p3	6
s2	p4	7
s3	p1	0
s3	p2	0
s3	p3	1
s3	p4	2

We wish to transform table T1 into a new table T2 where data from the “product” column in T1 is used to determine the columns in the new table T2, and the data from the “amount” column in T1 is used as the data for that new columns in T2. That is, p1, p2, p3, p4 (i.e., the data from the “product” column in T1) become columns in the new table T2, and 1, 2, 3, …, 7 (i.e., data from the “amount” column in T1) become data for that new columns in T2, as follows:

T2

shop	p1	p2	p3	p4
s1	3	2	1	0
s2	4	5	6	7
s3	0	0	1	2

For large amounts of data such a transformation can become a time-consuming and resource-intensive process. To make the transformation as efficient as possible we can make use of the Map-Reduce framework.

8.1 Express the transformation of T1 into T2 in the Map-Reduce framework.

8.2 Extend the Map-Reduce solution from exercise 8.1 to calculate the total amount of available products in each shop, i.e.:

T3

shop	total amount of available products
S1	6
S2	22
S3	3

9 – NoSQL Data Modeling

The following relations are given:

Knows(personid_a, personid_b, since) records information about which persons know which other persons and the year since they know them (for simplicity we assume that the "knows" is not symmetric, i.e. if p1 knows p2 it doesn't mean that p2 knows p1).

Knows

personid_a	personid_b	since
p1	p2	2010
p1	p3	2011
p2	p3	2010
p4	p5	2012
p4	p1	2011
p5	p3	2011

Person(personid, name, age) records information about the id, name, and age of persons.

Person

personid	name	age
p1	Joe	22
p2	Ann	22
p3	Bill	21
p4	Pat	23
p5	John	24

Assume *personid_a* and *personid_b* from *Knows* are foreign keys pointing to *personid* in *Person*.

9.1 Represent the data entities and the relationship between the data entities from above tables in a document structure – to be stored and manipulated in a document database (e.g., MongoDB). Provide the JSON documents and explain your modeling choices.

9.2 Represent the data entities and the relationship between the data entities from above tables in a labeled property graph structure – to be stored and manipulated in a labeled property graph database (e.g., Neo4j). Draw the graph and discuss the elements of the graph (nodes, properties, relationships, labels, etc.) (Alternatively, you can provide the Neo4j (Cypher) CREATE statements to construct the graph). Explain your modeling choices.

9.3 Discuss the advantages/disadvantages of representing data in the relational, document, graph models in the context of the example in this exercise.

> END OF SPRING 2020 IN3020 EXAM <

> and **GOOD LUCK** again <