DI- FCT/UNL **April 24th, 2023**

Database Systems

Test-1 2022/23

Duration: 2 hours (limited information)

Group 1 (each question is worth 2,5 values out of 20)

Consider part of a database for a bus company for a European network covering 32 countries (where the attributes constituting the primary key are underlined):

Locations({locCode,name,country,population,...}) Routes({numberR,locOrigin,locDestination})

Trips({numberR,date,driverCode,licensePlate}) Drivers(driverCode,name,sex,birthdate,...)

For each of these tables there is a B+ tree non-clustering index on the primary key attribute(s), created with the column order indicated in the tables.

The adopted DBMS uses blocks of 8KiB (8192 bytes). The records of all tables have a variable size, and, on average, a record of the locations table occupies 0.5KiB, a driver occupies 1KiB, a record of Routes and Trips occupy 128 bytes. At any given time, the locations table has 3,200 tuples, the routes table 12,000 tuples, the trips table 3,000,000 tuples and the drivers table 1,000 tuples. A B+ tree node can contain about 100 search keys, and it is known that a seek time is 10ms and the transfer time of a block t_T is 1ms, while the memory only holds 100 blocks.

Note: In this group, whenever examples are requested, these must be exclusively about this database, Additionally, all the answers must contain a brief justification.

1 a) Present two execution plans for the following SQL query, briefly justifying which of the plans has the least cost in the given database.

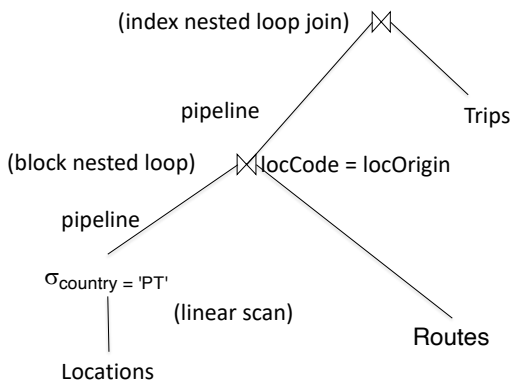
```
SELECT numberR, name FROM Drivers NATURAL INNER JOIN Trips
WHERE sex = 'F'
ORDER BY name
```

1 b) Suppose that the itinerary of each route is to be stored in a column of table Routes, where a tuple in average would occupy in average 10KiB. Can we use the mechanism of *"slotted pages"* to store the tuples of this table? If not, what are the alternatives to store such data ?

1 c) Indicate whether in your opinion the SQL query below can benefit from the existing indices.

```
SELECT * FROM Routes r, Locations l1, Locations l2
WHERE r.locOrigin= l1.locCode AND r.locDestination = l2.locCode;
```

1 d) Consider the execution plan presented in the figure. Determine the least cost of this plan knowing that there are 100 locations in Portugal with 375 routes, and each route has in average 250 trips. Assume that the height of B+-tree for the non-clustering index of Trips is 3.



1 e) Explain why the index created by CREATE INDEX idx_country ON Locations(Country) is not appropriate for obtaining efficiently the answers to the following query. Propose an alternative solution such that the query time could improve.

```
SELECT count(*) FROM  Locations
WHERE country='PT' OR country='ES'
```

Group 2 (each question is worth 2,5 values out of 20)

Note: The response to each of the items in this group cannot, under any circumstances, exceed one page.

2 a) "*The way database records are allocated to disk blocks significantly affects the performance of a database system".*

Justify why this happens.

2 b) The hash-join algorithm has two distinct phases. Explain what each one of them consists of, their purpose, limitations and if it is always necessary to execute them.

2 c) The block nested loop-join algorithm, with the pseudo-code presented below, has a worst-case complexity of $br * bs + br$ block transfers + $2 * br$ seeks, and a best-case complexity of $br + bs$ block transfers + 2 seeks. Indicate how these formulas are derived, not forgetting to distinguish the best- and the worst-case situations.

```
for each block Br of r do begin
    for each block Bs of s do begin
        for each tuple tr in Br do begin
            for each tuple ts in Bs do begin
                Check if (tr,ts) satisfy the join condition
                if they do, add tr • ts to the result.
            end
        end
    end
end
```