

DI- FCT/UNL

April 26<sup>th</sup>, 2024**Database Systems****Test-1 2023/24****Duration: 2 hours (limited information)****Group 1 (each question is worth 2,5 values out of 20)**

Consider part of a database for a national library (the attributes forming the primary key are underlined):

Authors({ <u>authorID</u> ,name,birthdate,country,...})	AuthorsBooks({ <u>authorID</u> , <u>bookID</u> ,authorNR})
---	--

Books({ <u>bookID</u> ,title,year,summary})	BooksGenres( <u>bookID</u> ,genreID)
---	--------------------------------------

Copies( <u>bookID</u> ,copyNR,status,lent)	Genres( <u>genreID</u> ,description)
--	--------------------------------------

The authors are identified by an integer primary key, and have a name, birth date and birth country. The books are also identified by an integer primary key, and have a title, edition year and summary of content. Each book may have several authors, stored in AuthorBooks table, where the authorNR column is an integer attribute capturing the author order (1, for the first author, etc...). Each book can have several copies, distinguished by the copyNR attribute (numbered sequentially in each book), and the copy status is a string describing if the book is damaged, and the lent attribute is a boolean indicating if the book has been borrowed by some reader. The Genres table simply has a textual description of the existing Genres (e.g. History, Science Fiction, etc...), and the BookGenres table associates each Book with the corresponding Genres. For each of these tables there is a secondary B+ tree (non-clustering) index on the primary key attribute(s), created with the column order indicated in the tables. Additionally, there is a primary (clustered) index for Books(title).

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 Authors table occupies 1KiB, a book record occupies 2KiB, a record of Copies, Genres, and AuthorsBooks occupy 128 bytes, and BookGenres records' occupy 32 bytes. At any given time, the Authors table has 100,000 tuples, the Books table 1,000,000 tuples, the Copies table has 5,000,000 tuples, the Genres table has 1,000 tuples. The AuthorsBooks has 4,000,000 tuples, and the BooksGenres has 2,000,000 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 title, year, summary
FROM Authors NATURAL INNER JOIN AuthorsBooks NATURAL INNER JOIN Books
WHERE name = 'Fernando Pessoa' AND year >= 1930
```

**1 b)** Present a query for which the use of multitable clustered file organization would be helpful.

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

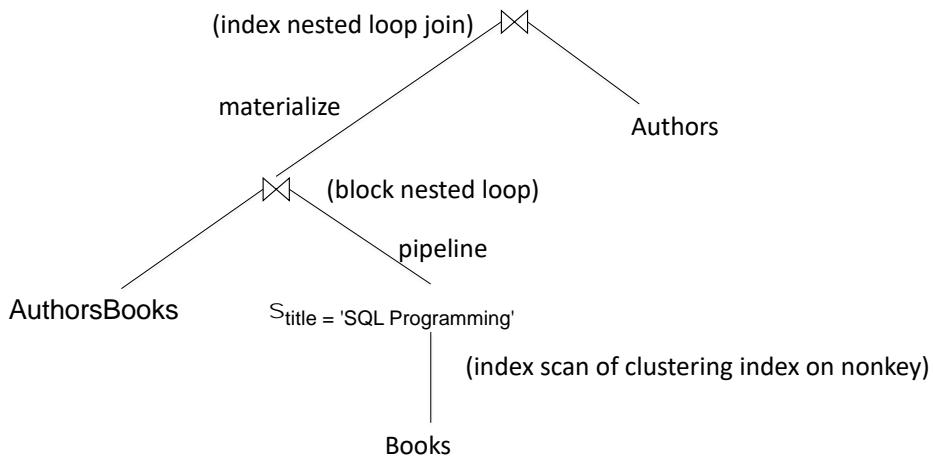
```
SELECT *
FROM Authors NATURAL INNER JOIN AuthorsBooks NATURAL INNER JOIN Books
WHERE authorNR = 1
```

**1 d)** Explain why the existing index for BooksGenres 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 Genres NATURAL INNER JOIN GenresBooks
WHERE description='Science Fiction' OR description='Horror'
```

(empty page – any content will not be graded)

**1 e)** Consider the execution plan presented in the figure. Determine the least cost of this plan knowing that there are 10 books with the title 'SQL Programming', and each book has in average 4 authors. Assume that the height of B+-tree for the non-clustering index of Authors is 3, and for the clustering index of Books is 4.



## 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)** Some RAID levels use block-striping for organizing files in disk systems. Describe this technique and why it is used.
- 2 b)** Explain the advantages and disadvantages of using of LSM trees to ordinary B<sup>+</sup>-tree index in write intensive workloads.
- 2 c)** The block nested loop-join algorithm, with the pseudo-code presented below, has a worst-case complexity of  $\lceil b_r / (M-2) \rceil * b_s + b_r$  block transfers  $+ 2 * \lceil b_r / (M-2) \rceil$  seeks when  $M$  — 2 disk blocks are used as blocking unit for outer relations with  $M$  = memory size in blocks. Indicate how this formula is derived and what for are the other two disk blocks used.

```

for each block  $B_r$  of  $r$  do begin
  for each block  $B_s$  of  $s$  do begin
    for each tuple  $t_r$  in  $B_r$  do begin
      for each tuple  $t_s$  in  $B_s$  do begin
        Check if  $(t_r, t_s)$  satisfy the join condition
        if they do, add  $t_r \bullet t_s$  to the result.
      end
    end
  end
end
  
```

**THE END**

