DEETC | ISEL
LEIC
Teachers: AB, GJ, JV, MP& ND

GRADE: _____

Course: SisInf

Student: _____ **Class**: _____ **No.:** _____

1st test

Duration: **75 min.** Date: 12 de april 2025

Carefully read the information on this page while you wait for the lecturer's instruction to begin the exam.

This exam contains 7 page(s), including **one** cover sheet (this one), **four** question and answer sheets, and **one** blank sheet. The exam consists of **6 questions**, some of which are divided into sub-questions, making a total of **20 points**.

Recommendations:

- It is advised that you read all the exam questions before starting, in order to better **plan your answering strategy**;
- True / False questions may have zero, one, or several correct answers;
- For True / False items, marks will only be awarded for answers that clearly indicate the intended option;
- In True / False questions:
  - **each incorrect answer deducts $\frac{1}{4}$ of the question's value**. You may choose not to answer, in which case no marks will be awarded or deducted;
  - Answers you wish to mark as **correct** should be **circled**, e.g. True $\boxed{False}$. **Illegible** answers will be marked as incorrect;
  - To correct an answer, **and only if you cannot erase it**, mark the incorrect answer with a **clearly visible cross** and then mark the correct one (True or False).
- You may use pencil and eraser, but do not forget to finalise your answers in **black ink** (preferably);
- For open-ended questions, justify and support your answers appropriately. The **simplicity** and **clarity** of your answers will be taken into account in the assessment;
- If you consider any **question ambiguous**, explain the interpretation you used when answering. **Your interpretation of the question is part of the assessment**;

By submitting the test, you agree with the conditions stated above.

Table (for the EXCLUSIVE use of the lecturer)

Question:	1	2	3	4	5	6	Total
Value:	3	3	4	4	3	3	20
Grade:							

---

1. (3 pontos) Considering the acronym ACID, which represents the desirable properties in transactional processing systems, including database management systems (DBMS), indicate what the letter I in the acronym stands for and relate it to the *Two-Phase Locking* (2PL) protocol.

2. (3 pontos) The properties of *cascadeless* and *strict* schedules are similar because both:
   (a) True False always allow conflict-serialisable schedules.
   (b) True False do not allow a transaction to write to a data item that has been written by another transaction which has not yet completed.
   (c) True False do not allow *Write/Write* conflicts.
   (d) True False protect against *dirty reads*.

3. Consider the *phantom records* anomaly, which results from the concurrent execution of transactions.
   (a) (1 ponto) Describe the anomaly, providing an example of a schedule that illustrates your explanation.

(b) (3 pontos) Considering the SQL instructions INSERT and UPDATE, indicate under what conditions they may cause the anomaly, assuming they are executed on a database that implements the 2PL protocol.

4. Consider the transactional processes below, executed concurrently on the table `conta(id integer primary key, saldo real);`. Also assume that `BL-xx` represent blocks of instructions executed sequentially without interruption from other transactions.

T1

```
-- Bl-1A:
begin transaction;
set transaction isolation level repeatable read;

-- Bl-1B:
select * from conta where Id = 1111;

-- Bl-1C:
select * from conta where id = 2222;
commit;
```

T2

```
-- Bl-2A:
update conta set saldo = saldo + 500 where id =
    1111;

-- Bl-2B:
update conta set saldo = saldo + 500 where id =
    2222;
```

(a) (2 pontos) Considering the schedule `<BL-1A, BL-2A, BL-1B, BL-2B, BL-1C>`, indicate at the end of each block of instructions the number and type of locks, assuming that the DBMS implements the 2PL protocol.

(b) (2 pontos) For each of the transactional processes T1 and T2, indicate the possible anomalies that may occur, providing a schedule that is susceptible to those anomalies.

5. (3 pontos) Considering the `account` table defined below, write a function in plpgsql that uses a cursor to return all rows from the `account` table where the `balance` field is greater than a value passed as a parameter. You should take into account that: (a) Each row is returned using `RETURN NEXT`; (b) The function must return `SETOF account`.

```sql
create table public.account
(
    id serial primary key,
    "description" varchar(100) not null,
    balance money not null,
    client int not null references client,
    updated timestamp with time zone null
);
```

```sql
CREATE FUNCTION get_accounts(v money)
RETURNS SETOF account AS
$$
-- Implementar
%$$ LANGUAGE plpgsql;
```

6. (3 pontos) Write a stored procedure in PostgreSQL that uses the `get_accounts_balance` function and processes the results internally:
   (a) The procedure must accept one input parameter: a minimum `balance` (of type `money`);
   (b) It must internally invoke the function `get_accounts_balance(v money)`;
   (c) It must iterate over the returned rows using a `FOR` loop;
   (d) It must use `raise notice` to display the values of the `description` and `balance` fields for each tuple.

   Tip: You may use a variable of type `ROWTYPE` to store each row.

```
1 CREATE OR REPLACE PROCEDURE accounts_balance(min_balance money)
2 LANGUAGE plpgsql
3 -- Implementar
```

FIM

FOLHA EM BRANCO