Introduction to Database Systems
BSc and MSc Final Exam
Fall 2022

Eleni Tzirita Zacharatou

December 16, 2022

Instructions

You have 4 hours to answer 6 problems described in the following. There are 7 problems in the exam, but problem 2 is only for BSc students and problem 3 is only for MSc students. The exam consists of 14 numbered pages. Your answers must be provided in the LearnIT quiz *Final Exam December 2022*.

Database Description for Questions 1–3

In this exam you will work with a fictional garments database. To start working with the database, run the commands in `idb-fall-2022.sql` found in LearnIT using the PostgreSQL DBMS on your laptop. As the database contains nearly 200K rows, it is recommended to use `psql` for this purpose. If you already have a database called `garments` on your server, you must import/run the script again because the exam version might be different.
The database has the following schema:

```
gDesigners(d_ID, d_Name, d_Country)
gGarments(g_ID, g_Price, d_ID, co_ID)

gTypes(t_ID, t_Name, t_Category)
gHasType(g_ID, t_ID, ht_Importance)

gFabrics(f_ID, f_Name)
gMadeOf(g_ID, f_ID, mo_Percentage)
gElements(f_ID, e_Element)
```

Primary and foreign keys are defined and most attributes are self-explanatory. The attribute names start with the capital letter(s) of the first table in which they appear. In the `gGarments` table, the attribute `d_ID` refers to the main designer, and the nullable attribute `co_ID` refers to the ID of a collaborating designer (i.e., a co-designer). Designers cannot collaborate with themselves.
Note that some data is incomplete. For example, the sum of `mo_Percentage` is not 100 for all garments. Furthermore, note that the data is completely fictional: while some tables are (partially) based on actual online lists, others are randomly generated.
To understand the details of the tables, you may study the DDL commands (the CREATE TABLE statements are at the top of the script), consider the ER diagram in Figure 1, or inspect the tables using SQL queries.
The number of tuples in each table is as follows:

- `gDesigners` - 2314 tuples.
- `gGarments` - 28170 tuples.
- `gTypes` - 83 tuples.
- `gHasType` - 30972 tuples.
- `gFabrics` - 56 tuples.
- `gElements` - 218 tuples.
- `gMadeOf` - 128011 tuples.

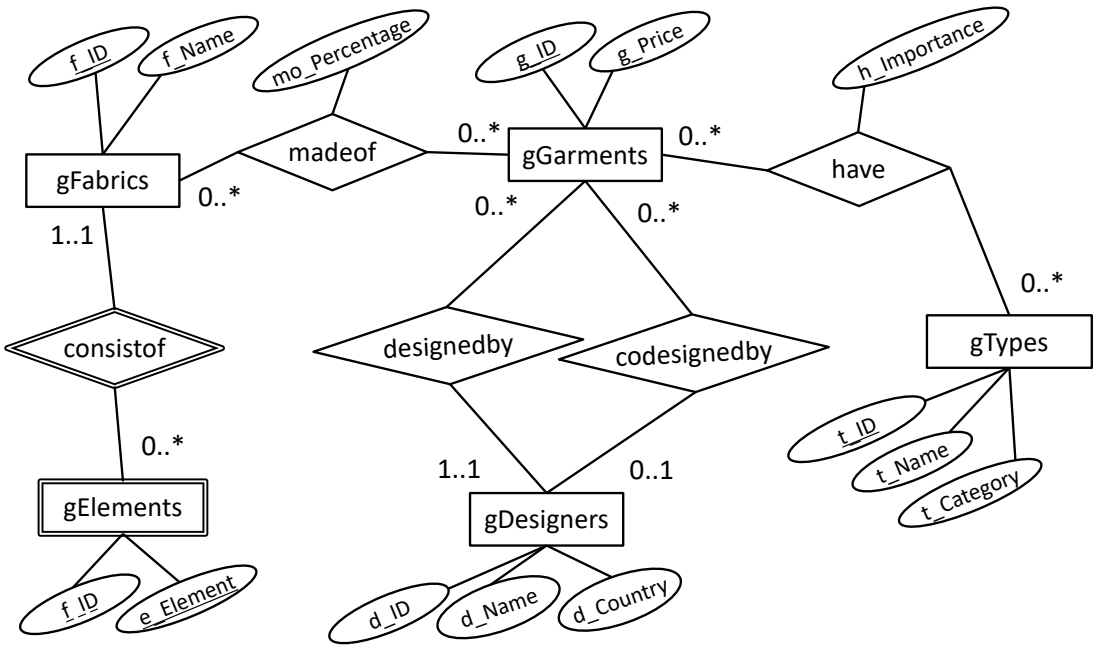

Figure 1: ER Diagram for the garments database.

Instructions for SQL Queries in Question 1

Queries must return correct results for any database instance. They should avoid system-specific features, including the LIMIT keyword. Queries should not return anything except the answer; a query that returns more information will not receive full points, even if the answer is part of the returned result. A sequence of several queries that answer the question will not receive full points, but subqueries and views can be used. Queries should be as simple as possible; queries that are unnecessarily complex may not get full points, despite returning the correct answer. If you are unable to complete the query you can still submit your attempt, along with a brief description, and it may be given partial points.

1 SQL (40 points)

Answer each of the following questions using a single SQL query on the garments database:

(a) The chiffon fabric consists of 9 different elements. How many different elements does the cashmere fabric consist of?

(b) There are 84 countries that have more than one designer. How many countries have more than two designers?

(c) In the database, 12609 garments have a price that is higher than the average garment price. How many garments have a price that is lower than the average garment price?

(d) How many garments with missing price values have a type of importance equal to six?

(e) How many *main* designers have designed garments in all categories that exist in the database? Note that in this query you should only consider the main designers, not co-designers.

(f) The designer with d_ID of 200 has collaborated, either as the main designer or as the co-designer, with 11 other designers from 7 different countries. How many designers have collaborated with other designers from 14 different countries?

Enter each query and its numerical answer in the LearnIT quiz.

2 (BSc ONLY) SQL Programming (5 points)

Consider the SQL trigger code in Figure 2.
The goal of the trigger is to add some constraints on the prices of the garments. To answer this question, you need to study the `gGarments` relation of the garments database.

```
DROP TRIGGER IF EXISTS CheckPrice ON gGarments;
DROP FUNCTION IF EXISTS CheckPrice();

CREATE FUNCTION CheckPrice()
RETURNS TRIGGER
AS $$ BEGIN
        -- Check 1: Price value
        IF (NEW.g_Price <= 0) THEN
                RAISE EXCEPTION 'The price must be greater than 0'
                USING ERRCODE = '45000';
        END IF;
        -- Check 2: Price increase
        IF (NEW.g_Price > 1.8*OLD.g_Price) THEN
                RAISE EXCEPTION 'The price cannot increase by more
than 80 percent'
                USING ERRCODE = '45000';
        END IF;
  RETURN NEW;
END; $$ LANGUAGE plpgsql;

-- Trigger code
CREATE TRIGGER CheckPrice
BEFORE INSERT OR UPDATE ON gGarments
FOR EACH ROW EXECUTE PROCEDURE CheckPrice();

-- Test the trigger
INSERT INTO gGarments VALUES (1, 0, 1557);
UPDATE gGarments SET g_Price = 3000000 where g_ID = 500;
```

Figure 2: Trigger `CheckPrice` for the `gGarments` relation.

Select the true statements:

(a) The trigger is incorrectly implemented as a 'BEFORE' trigger.

(b) The INSERT statement will give an error.

(c) Since this is a 'BEFORE' trigger, it is not possible to change the inserted price in the trigger.

(d) The UPDATE statement will give an error.

3 (MSc ONLY) Database Programming (5 points)

To answer this question, you need to study the `gDesigners` relation of the garments database. Consider the Java code in Figure 3.

```
public static void insertDesigner(Connection conn, int designerId, string designerName, string countryName)
   throws SQLException {
        PreparedStatement st = conn.prepareStatement
                                ("INSERT INTO gDesigners (d_ID, d_Name, d_Country) VALUES (?,?,?)");

        st.setInt(1, designerId);
        st.setString(2, designerName);
        st.setString(3, countryName);

        st.executeUpdate();

        st.close();
}
```

Figure 3: Code to insert a new designer in the database.

Select the true statements:

(a) The code is safe against SQL injection attacks.

(b) The code should use transactions to preserve the integrity of the database.

(c) The method call 'insertDesigner(1,"Gustavo Cadile","Argentina")' will handle freeing the resources associated with the statement.

(d) By default, the designer will be inserted into the database without calling conn.commit().

4 ER Diagrams and Normalization (25 points)

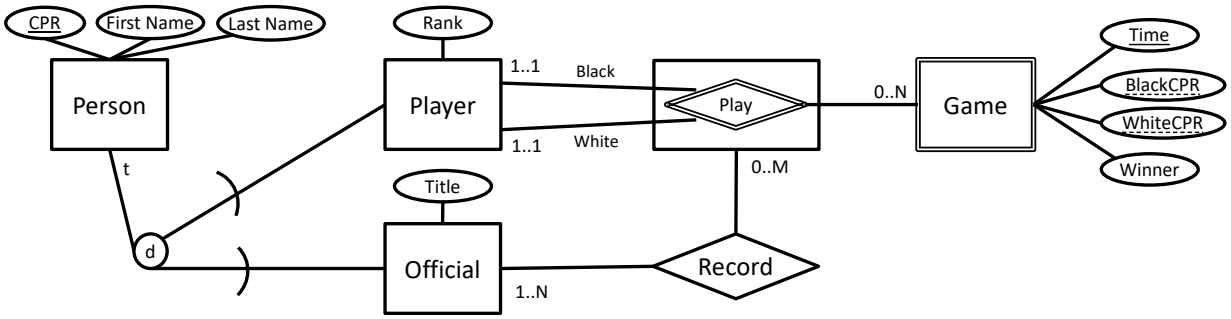

Figure 4: ER Diagram for a chess database.

a) The ER diagram in Figure 4 shows a chess tournament database. Select the true statements. You should base your answers **only** on the ER diagram:

(a) Every player plays at least one game.

(b) Every game has two players.

(c) An official is a player.

(d) The same "play" relation can be recorded by multiple officials.

(e) A "play" relation must be recorded by at least one official.

(f) Every official records at least one "play" relation.

(g) There can exist a person who is neither a player nor an official.

b) Write SQL DDL commands to create the chess database based on the ER diagram in Figure 4. The DDL script must run in PostgreSQL. The relations must include all primary key, canditate key, foreign key, and NOT NULL constraints. Constraints that cannot be enforced with standard primary key and foreign key constraints can be omitted. Make reasonable assumptions on the attribute types.

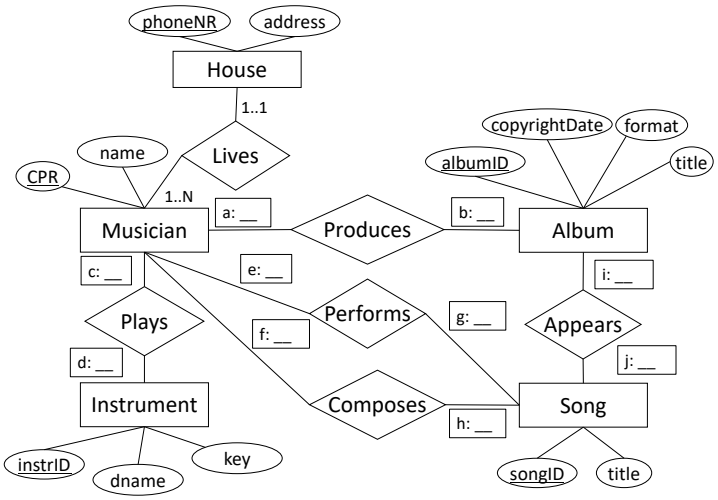

Figure 5: ER Diagram for the record label database (**Version A**).

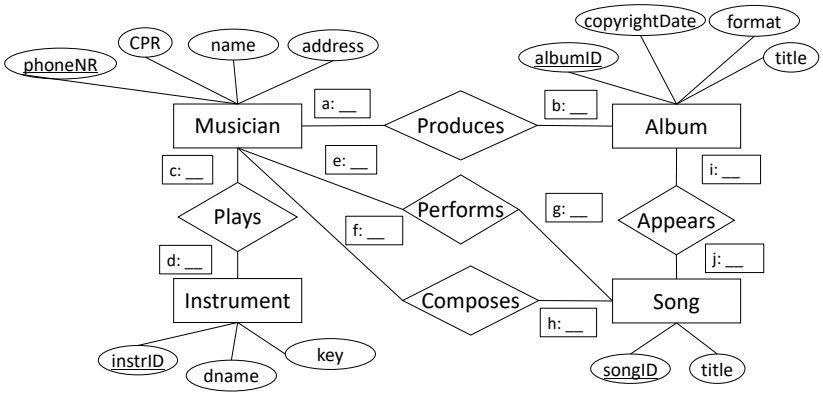

Figure 6: ER Diagram for the record label database (**Version B**).

c) Consider Figures 5 and 6 as well as the following information for a record label database called ITU Records.

(a) Each musician who records with ITU Records has a CPR number, a name, an address, and a phone number. Poorly paid musicians often share the same address. Each address corresponds to a unique phone number.

(b) Each instrument used in songs recorded at ITU Records has a unique identification number, a name (e.g., guitar, synthesizer, flute), and a musical key (e.g., C, B-flat, E-flat).

(c) Each album recorded at ITU Records has a unique album identifier, a title, a copyright date, and a format (e.g., CD or Vinyl).

(d) Each song recorded at ITU Records has a unique identification number and a title.

(e) Each musician may play several instruments and a given instrument may be played by several musicians.

(f) Each album has a number of songs on it, but no song may appear on more than one album.

(g) Each song is performed by one or more musicians, and a musician may perform a number of songs.

(h) Each song is composed by one or more musicians, and a musician may compose a number of songs.

(i) Each album has exactly one musician who acts as its producer. A musician may produce several albums.

Figures 5 and 6 show two different ER diagrams for the ITU Records database. Select the ER diagram that best fits the requirements and explain your choice. Additionally, write down the cardinality for each blank box (from a to j).

d) Consider a table $R(A, B, C, D)$ with the following dependencies:

$$\begin{aligned} B &\rightarrow C \\ AB &\rightarrow AC \\ D &\rightarrow A \end{aligned}$$

Select the true statements:

(a) B is a candidate key of R.

(b) $AD \rightarrow A$ is a trivial functional dependency.

(c) Normalizing to BCNF results in exactly three relations.

(d) The relation $Z(B, C)$ is in BCNF.

(e) The relation $W(B, D)$ is in BCNF.

e) Consider a table $R(L, M, N, O)$ with the following dependencies:

$$\begin{aligned} L &\rightarrow M \\ LN &\rightarrow MN \\ MN &\rightarrow O \\ L &\rightarrow N \end{aligned}$$

Normalize R to the highest possible normal form (3NF or BCNF), based on functional dependencies, while allowing all functional dependencies (excluding trivial, unavoidable, and redundant dependencies) to be checked within a single relation. For each resulting relation, write its columns and clearly indicate whether it is in BCNF.

5 Index Selection (10 points)

Consider the following large relation with information on taxi rides:

Ride(id, pickup.x, pickup.y, price, <many long attributes>)

The four given attributes are integer values, and none of them is nullable. The pickup.x and pickup.y attributes denote the location of the taxi pickup. Assume that both pickup.x and pickup.y range uniformly from -5000 to 5000. The price attribute denotes the price of the taxi ride. Now consider the following three SQL queries:

Query 1

```
select pickup.x, pickup.y
from Rides
where id = 2016;
```

Query 2

```
select id
from Rides
where price = (select max(price) from Rides);
```

Query 3

```
select *
from Rides
where pickup.x > 0 and pickup.y < 0;
```

Answer each of the following questions:

a) Select the correct statements in the following:

(a) Query 1 will benefit more from a clustered index on id rather than an unclustered index on id.

(b) Query 1 will perform best without any index.

(c) Query 2 will benefit from an unclustered index on price.

(d) For Query 3, a clustered index on pickup.x will perform better than an unclustered index on pickup.x.

(e) Query 3 will benefit more from a clustered index on pickup.x rather than a clustered index on pickup.y.

b) Indicate for each query whether a covering index would be preferable to a clustered index. Explain your answer and define the indexes you consider.

c) Considering all three queries, which single clustered index would you define on the relation? Explain your answer.

6 Hardware and DBMS Design (10 points)

a) Select the correct statements below:

(a) If the buffer manager applies a NO STEAL policy, there is no need to perform REDO logging.

(b) It is possible to accelerate a nested loop join by creating an index on the outer relation.

(c) Each internal node in a B+-tree contains hundreds of search keys.

(d) With consistent hashing, when a server crashes, there is no need to redistribute all the keys from scratch.

b) Assume a *hybrid* storage layout that combines the characteristics of a row-store and a column-store. For a given relation, this hybrid layout stores the same data on each page as a row-store. Within each page, however, the hybrid layout groups all values of a particular attribute together on a minipage. That is, instead of storing the records contiguously in the page like a row-store, within each page, it first stores all values for the first attribute, followed by all values for the second attribute, and so on. Figure 7 shows an example with four records of the relation `Person(id, name, age)`. The row-store page stores the four records (for Jane, John, Jim, and Susan) one after the other. RH1-RH4 are the record headers. The hybrid page first stores all the values for the id, then all the values for the name, and finally all the values for the age. Note that this is different from a column-store, which uses a separate file for each column.

Discuss the pros and cons of the hybrid layout compared to row-stores and column-stores.

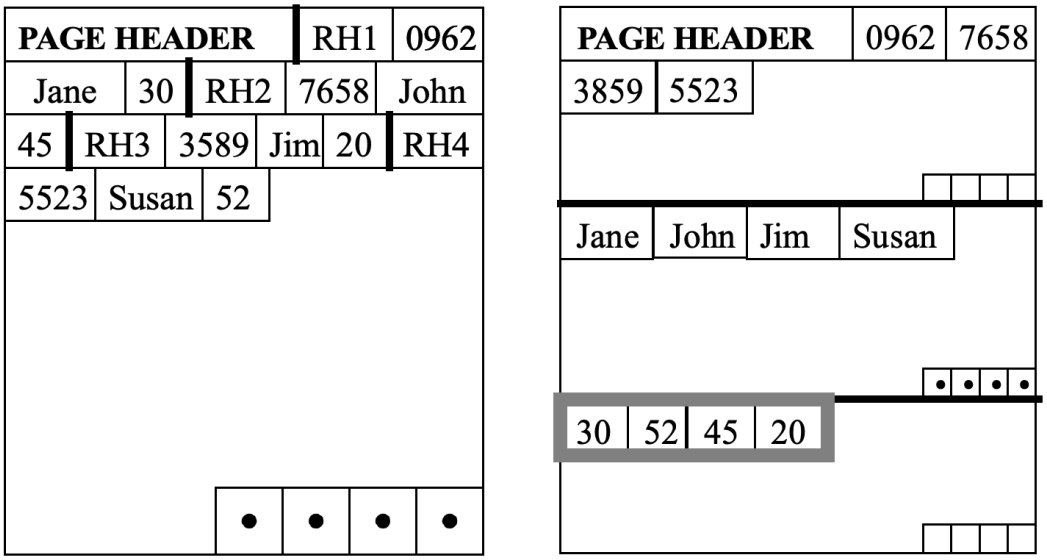

Figure 7: Row-store page layout (left) vs. Hybrid page layout (right).

7 Data Systems for Analytics (10 points)

a) Select the correct statements below:

(a) Big data analytics applications perform many random disk accesses since they process data in random order.

(b) Spark applies transformations to RDDs as soon as possible and stores the intermediate results in memory.

(c) Big data originates from a big variety of sources and can be structured, semi-structured, or unstructured.

(d) Spark can be used to process structured data.

b) Assume that you need to process a massively large dataset with a limited budget. Which framework would you choose as a more cost-effective solution, Hadoop or Spark? Explain your choice.