Introduction to Database Systems
MSc and BSc Retake Exams

Björn Thór Jónsson

March 8, 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 11 numbered pages. Unless instructed otherwise your answers must be provided in the LearnIT quiz *Retake Exam March 2022*.

Database Description for Questions 1–3

In this exam you will work with a fictional (and poorly designed!) database of COVID test results. To start working with the database, run the commands in `idb-march-2022.sql` found in LearnIT using the PostgreSQL DBMS on your laptop. It is recommended to use `psql` for this purpose. The database has the following schema:

```
Testers (id, name)
Sites (id, location, ZIP)
Kits (id, producer)

Risks (id, level)
Variants (ID, name, riskID)

Detects (kitID, variantID, accuracy)
Tests (testerID, siteID, kitID, variantID, time)
```

Primary keys and foreign keys are defined and attributes are largely self-explanatory. You may study the DDL commands to understand the details of the tables (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. Following are some additional notes that are important for your queries:

- In the `Variants` table, a `riskID` of `NULL` indicates that the risk level has not been assessed.
- In the `Tests` table, a `variantID` of `NULL` indicates that the test was negative and hence no variant was detected. All positive tests have a diagnosed variant.
- In the `Detects` table, the `accuracy` attribute is a percentage ranging from 0 to 100. This number indicates how well a testing kit detects a variant, when an individual is infected with that variant. If the accuracy for a particular kit for a particular variant is unknown, there is no data for that combination in the table.
- To get the year from the timestamp, use `extract(year from time)`. To retrieve the latest timestamp, use `max(time)`.

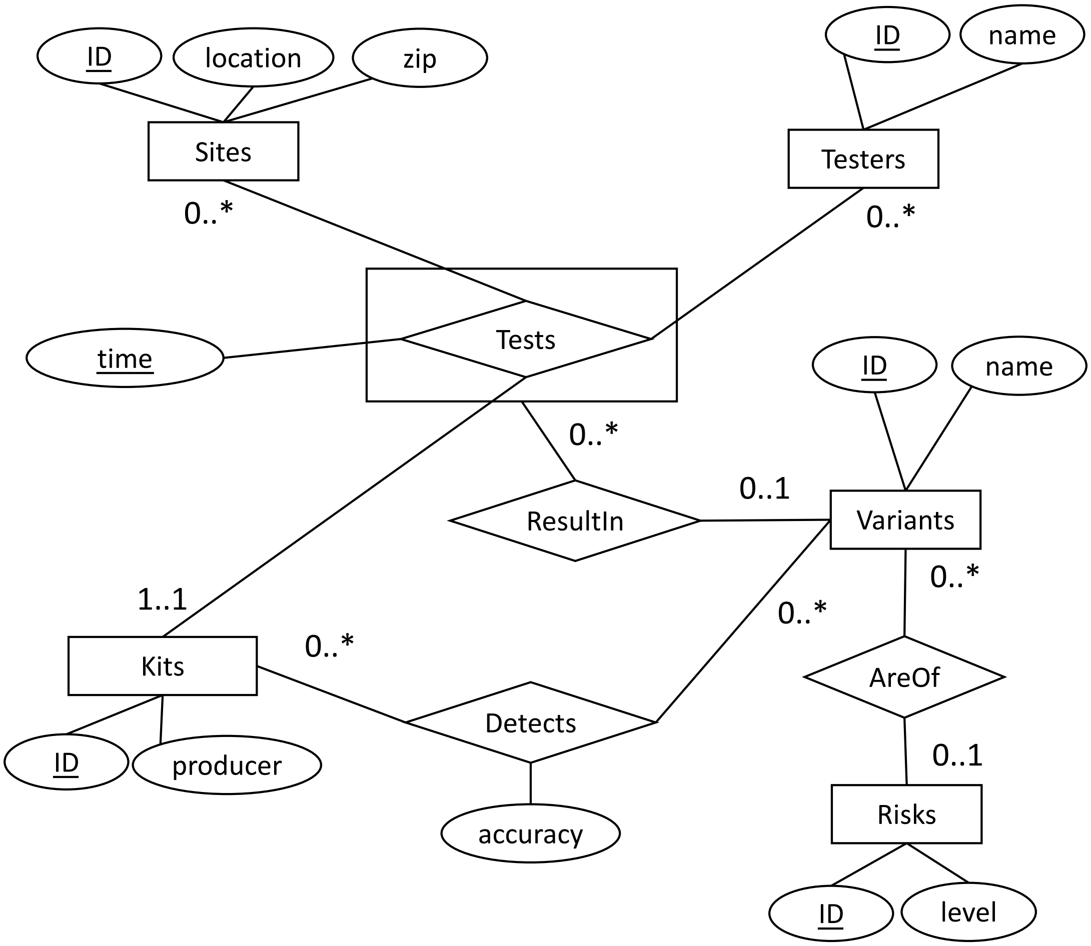


Figure 1: ER Diagram for the COVID testing database.

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 marks, 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 examination database. Enter each query, along with its numerical answer, in LearnIT. Queries must adhere to the detailed guidelines given on Page 3.

(a) There are 2,586 tests that have detected the 'delta' variant. How many tests have detected the 'omicron' variant?

(b) There are 6 variants with an unknown risk level. How many variants have never been detected with a test?

(c) How many different variants of the 'extreme' risk level have been detected in 2021 by a tester called 'Kent Lauridsen' using a kit produced by 'JJ'?

(d) There are 7 variants that can be detected by some kit with at least 80% accuracy. How many variants cannot be detected by any kit with more than 50% accuracy?

(e) The best average accuracy for any kit is about 59.89%. What is the lowest average accuracy for any variant which is detected by more than one kit?

(f) The tester with ID = 68 has performed 287 tests. What is the ID of the tester that has performed the most tests with the kit produced by 'JJ'?

Note: The output of this query is a single identifier. The query must be written to return the correct results, however, even if there were two testers tied with the most tests.

(g) There are 3 testers that have detected all variants with a known risk level. How many testers have detected all variants with risk level 'mild'?

Note: This is a division query; points will only be awarded if division is attempted.

(h) Write a query that returns the timestamp for the last time a detection was made of a variant with risk level 'extreme' using a kit that has $<$ 10% accuracy for that variant.

Note: The output of this query is a single timestamp. It may be copied directly with or without quotes.

2 (BSc ONLY) SQL programming (5 points)

Consider the SQL trigger code in Figure 2. The goal of the trigger is to log any tests that result in a variant at the 'extreme' risk level. To answer this question, you will need to study the database instance.

```
CREATE TABLE LogExtremeTests (
    testerID INT,
    siteID INT,
    time TIMESTAMP,
    PRIMARY KEY (testerID, siteID, time),
    FOREIGN KEY (testerID, siteID, time) REFERENCES Tests(testerID, siteID, time)
);

DROP TRIGGER IF EXISTS FlagExtremeTests ON Tests;
DROP FUNCTION IF EXISTS FlagExtremeTests();

CREATE FUNCTION FlagExtremeTests() RETURNS TRIGGER
AS $$ BEGIN
  IF EXISTS (
      SELECT *
      FROM Variants V
        JOIN Risks R on V.riskID = R.ID
      WHERE V.id = NEW.variantID
        AND R.level = 'extreme') THEN
    INSERT INTO LogExtremeTests VALUES (NEW.testerID, NEW.siteID, NEW.time);
  END IF;
  RETURN NEW;
END; $$ LANGUAGE plpgsql;

CREATE TRIGGER FlagExtremeTests
AFTER INSERT ON Tests
FOR EACH ROW EXECUTE PROCEDURE FlagExtremeTests();

INSERT INTO Tests VALUES (1, 1, 1, 1, CURRENT_TIMESTAMP);
```

Figure 2: Insertion trigger `FlagExtremeTests` for the `Tests` relation.

Select the true statements:

(a) The trigger is incorrectly implemented as an 'AFTER' trigger.

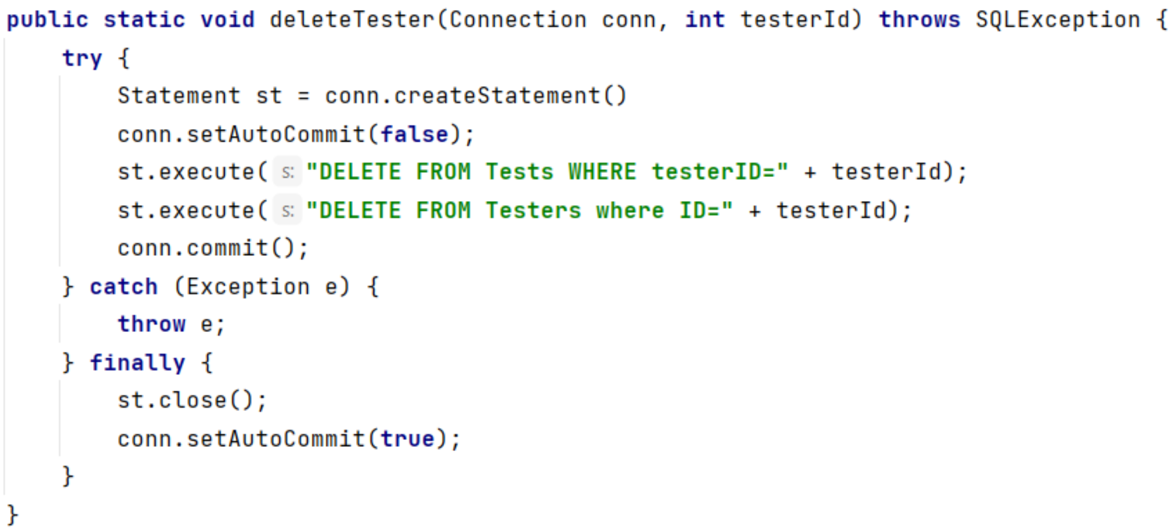
(b) The INSERT statement will succeed in adding a row to the `LogExtremeTests` relation.

(c) The data in the `LogExtremeTests` relation is sufficient to allow a competent SQL programmer to find the variant detected and the kit used.

(d) As this is an AFTER trigger, adding code that rejects tests based on data values (e.g., because they are older than one week) would not work.

3 (MSc ONLY) Database programming (5 points)

To answer this question, you will need to study the relations of the exam database. Consider the Java code in Figure 3.



```
public static void deleteTester(Connection conn, int testerId) throws SQLException {
    try {
        Statement st = conn.createStatement()
        conn.setAutoCommit(false);
        st.execute( s: "DELETE FROM Tests WHERE testerID=" + testerId);
        st.execute( s: "DELETE FROM Testers where ID=" + testerId);
        conn.commit();
    } catch (Exception e) {
        throw e;
    } finally {
        st.close();
        conn.setAutoCommit(true);
    }
}
```

Figure 3: Code to remove tester information from the database.

Select the true statements:

(a) The code is not safe against SQL injection attacks.

(b) The code is not using transactions correctly.

(c) The connection must be to a PostgreSQL server for the code to work correctly.

(d) The connection is closed automatically when a statement is closed.

(e) The code frees the resources associated to the statement when no longer needed.

4 ER Diagrams and Normalization (25 points)

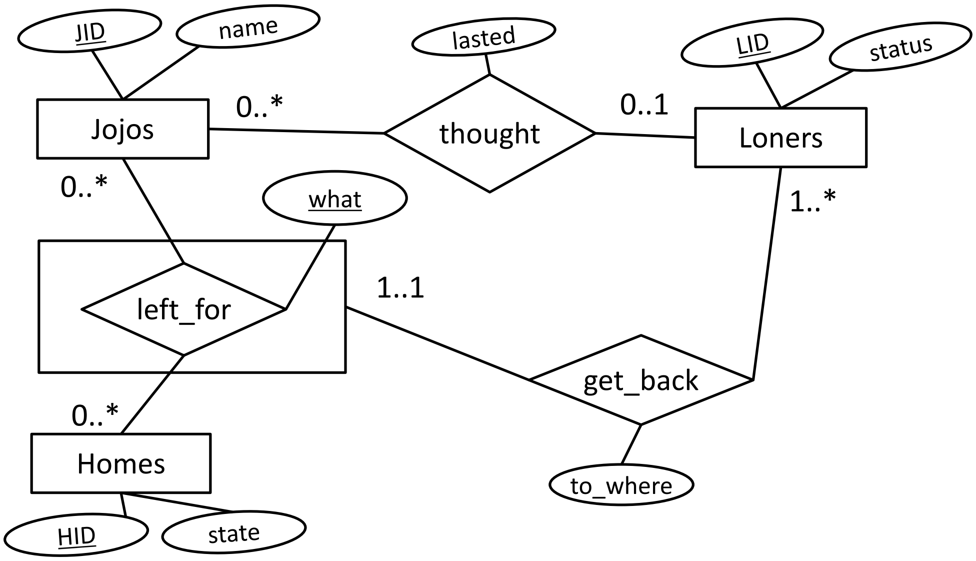


Figure 4: ER Diagram for a database of Jojos.

a) The ER diagram in Figure 4 shows a database for Jojos who thought the were Loners. (If you are wondering what a Jojo is, the ER diagram is very loosely based on lyrics of the Beatles' song Get Back. It is recommended to not worry about what a Jojo is, however, and focus instead on the relationships and their participation constraints.) Select the true statements. You should base your answers **only** on the ER diagram:

(a) All Jojos connect to at least one Loner through some relationships.

(b) All Loners connect to at least one Home through some relationships.

(c) All Homes connect to at least one Loner through some relationships.

(d) Some Loner may be connected to itself through some relationships.

(e) Some Home may be connected to itself through some relationships.

(f) All Jojos must be connected to themselves through some relationships.

b) Write SQL DDL commands to create a database based on the ER diagram in Figure 4. The DDL script *must run* in PostgreSQL as a whole. The relations must include all primary key and foreign key constraints. Constraints that cannot be enforced with standard primary key and foreign key constraints should be omitted. The type of key attributes should be INT, all other attributes may be of type VARCHAR.

c) Write an ER diagram for a database of thesis writing based on the following requirements. The diagram should clearly show the entities, attributes, relationships and participation constrains described below. Use the notation presented in the textbook and lectures. Attributes are only important if mentioned. If you need to make additional assumptions put them in the box below.

- Faculty supervise students.
- Each student has a least one and at most 3 supervisors.
- A student writes one thesis.
- Each thesis has at least one topic.
- Faculty can validate theses.
- Faculty may also evaluate the supervision of students.

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

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

Select the true statements:

(a) M is the only (candidate) key of R.

(b) $O \rightarrow N$ is a redundant functional dependency.

(c) Normalizing to 3NF or BCNF results in exactly two relations.

(d) The relation can be normalized to BCNF without losing dependencies.

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

$$\begin{aligned} LMN &\rightarrow O \\ O &\rightarrow MP \end{aligned}$$

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 canals:

Canals(id, x, y, depth, <many long attributes>)

The four given attributes are integer values, and none of these are nullable. Assume that the x and y attributes range uniformly from 0 to 10000. Now consider the following three SQL queries:

Query 1

```
select x, y
from Canals
where depth = (select max(depth) from Canals);
```

Query 2

```
select avg(depth)
from Canals;
```

Query 3

```
select *
from Canals
where x > 5000;
```

Answer each of the following questions:

(a) Indicate for each query whether a clustered index could be defined (i.e., would be preferable to a non-clustered index or no index at all). Explain your answer and define the indexes you consider.

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

(c) Considering all three queries, which 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 only old values are logged during updates, durability can still be implemented.

(b) Key-value stores excel at joining large relations.

(c) SSDs are faster than HDDs, partly because they have no moving parts.

(d) Support for ACID transaction properties is important for workloads with many small update transactions.

b) Discuss pros and cons of using a RAM-based relational database system, compared to an SSD-based system. Be sure to mention both pros and cons. For pros, it is not enough to state that it will be faster, as that is a given, but rather how it could be simpler.

7 Data Systems for Analytics (10 points)

a) Select the correct statements below:

(a) A well-designed relational database can never have incorrect data.

(b) Social value can be a significant reason for working with big data.

(c) Clustering data is generally achieved with one scan of the data collection.

(d) Support for ACID transaction properties is generally useless for big data analysis applications.

b) Some large companies have recently stopped using SSDs for big data applications and reverted to using HDDs. Give arguments for and against such a transition back to HDDs for big data applications.