Introduction to Database Design / Data Management MSc and BSc Exams

Björn Thór Jónsson

May 20, 2020

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.

Instructions for SQL Queries in Question 1

Queries must work for any database instance and should avoid system-specific language 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 of the problem with the query, and it may be given partial points.

Database Description for Questions 1–3

In this exam you will work with a fictional database for a volleyball federation. To start working with the database, run the commands in `idb-may-2020-DB.sql` using the PostgreSQL DBMS on your laptop. It is recommended to use `psql` for this purpose, as the file is about 75K lines.

The database contains a variety of information in the following schema:

```
Places(ID, name, population)
Clubs(ID, name, placeID, founded)
Genders(ID, gender, description)
Teams(ID, clubID, ordinal, genderID)
Divisions(ID, name, genderID)
TeamsInDivisions(teamID, divisionID)
Matches(ID, divisionID, hometeamID, awayteamID, homesets, awaysets)
Sets(matchID, setnumber, homepoints, awaypoints)
```

Attributes are largely self-explanatory. Primary keys and foreign keys are correctly defined, and there are four secondary keys (UNIQUE) in the relations. 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 on the next page, or inspect the tables using SQL queries. Some additional notes are in order:

- The data is randomly generated based on actual cities/towns (called places in the database) and football clubs from Denmark. It is incomplete in many ways, for example not all clubs have a known place.
- Each club can have many teams of each gender, which are then identified by their ordinal, which is a single character: A, B, etc.
- The data also has some (intentional) problems, such as male teams assigned to mixed divisions, etc.
- In the table Matches, when homesets and awaysets are NULL, the match has not yet taken place and no data is found in the Sets table for that match.

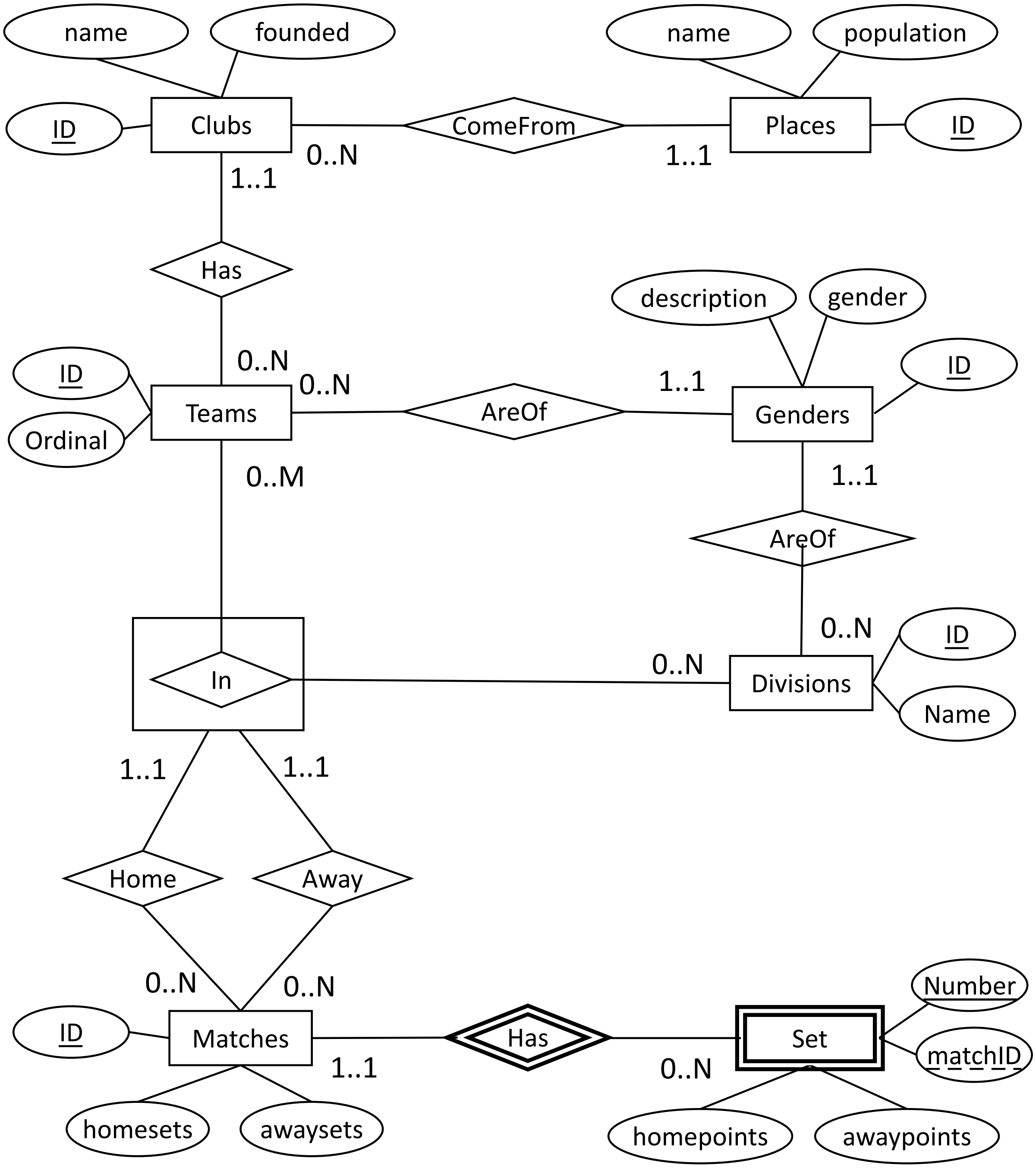


Figure 1: ER Diagram for the volleyball database.

1 SQL (40 points)

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

(a) There are 15 places with names that end with 'borg' in the database. How many places have names ending with 'lev'?

(b) The ID of the place with the smallest population is 133. What is the ID of the place with the largest population?

Note: This query returns an identifier, not a count of result rows.

(c) Officials now believe that each team should only be registered to one division. However, team with ID of 51, for example, is registered to three divisions. How many teams are registered to more than one division?

(d) Furthermore, officials have realised that teams should only be registered to divisions with the same gender as the team. How many teams are registered to divisions with a different gender from the team?

(e) The place with ID of 5 has a total of 15 teams with gender 'M' through its various clubs. What is the ID of the place that has the most teams with gender 'M'?

(f) There are 60 clubs which have teams of all genders. How many places have teams of all genders?

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

(g) How many clubs have teams registered to all divisions of the 'M' gender?

Note: This is also a (fairly complex) division query.

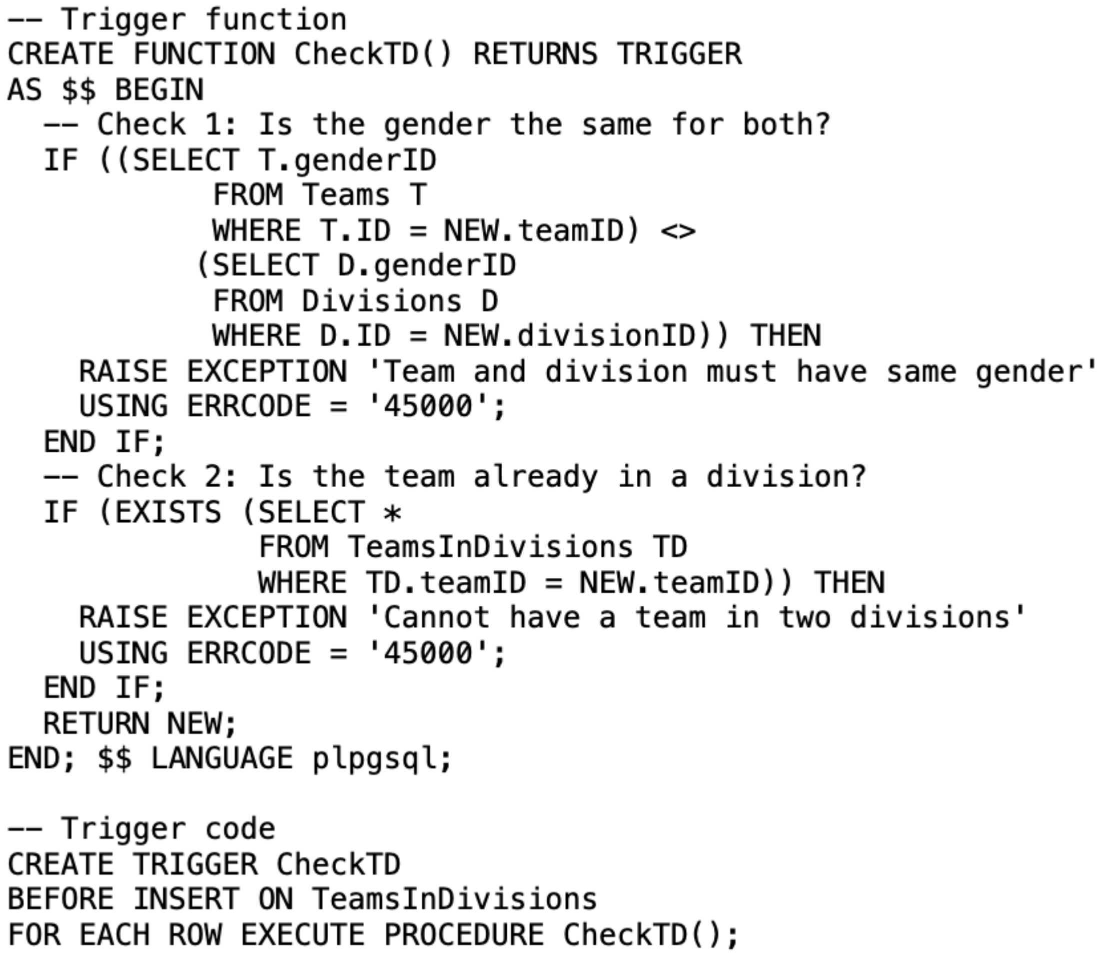
(h) Teams earn points in sets. For example, the team with ID of 0 has earned a total of 268 points. (Of those, 167 are home points, while 101 are away points; this division is not important, except to remind you to consider both home points and away points). What is the highest number of points earned by any one team?

Note: This query returns an sum of points, not a count of result rows.

Enter each query, along with its numerical answer, in LearnIT. Queries must adhere to the detailed guidelines given on Page 1.

2 (BSc ONLY) SQL programming (5 points)

Consider the SQL trigger code in Figure 2.



```
-- Trigger function
CREATE FUNCTION CheckTD() RETURNS TRIGGER
AS $$ BEGIN
  -- Check 1: Is the gender the same for both?
  IF ((SELECT T.genderID
           FROM Teams T
           WHERE T.ID = NEW.teamID) <>
          (SELECT D.genderID
           FROM Divisions D
           WHERE D.ID = NEW.divisionID)) THEN
    RAISE EXCEPTION 'Team and division must have same gender'
    USING ERRCODE = '45000';
  END IF;
  -- Check 2: Is the team already in a division?
  IF (EXISTS (SELECT *
             FROM TeamsInDivisions TD
             WHERE TD.teamID = NEW.teamID)) THEN
    RAISE EXCEPTION 'Cannot have a team in two divisions'
    USING ERRCODE = '45000';
  END IF;
  RETURN NEW;
END; $$ LANGUAGE plpgsql;

-- Trigger code
CREATE TRIGGER CheckTD
BEFORE INSERT ON TeamsInDivisions
FOR EACH ROW EXECUTE PROCEDURE CheckTD();
```

Figure 2: Insertion trigger `CheckDeptEmp` for the `dept_emp` relation.

Select the true statements:

(a) Check 1 can be replaced by a constraint on the `TeamsInDivisions` relation.

(b) Check 2 can be replaced by a constraint on the `TeamsInDivisions` relation.

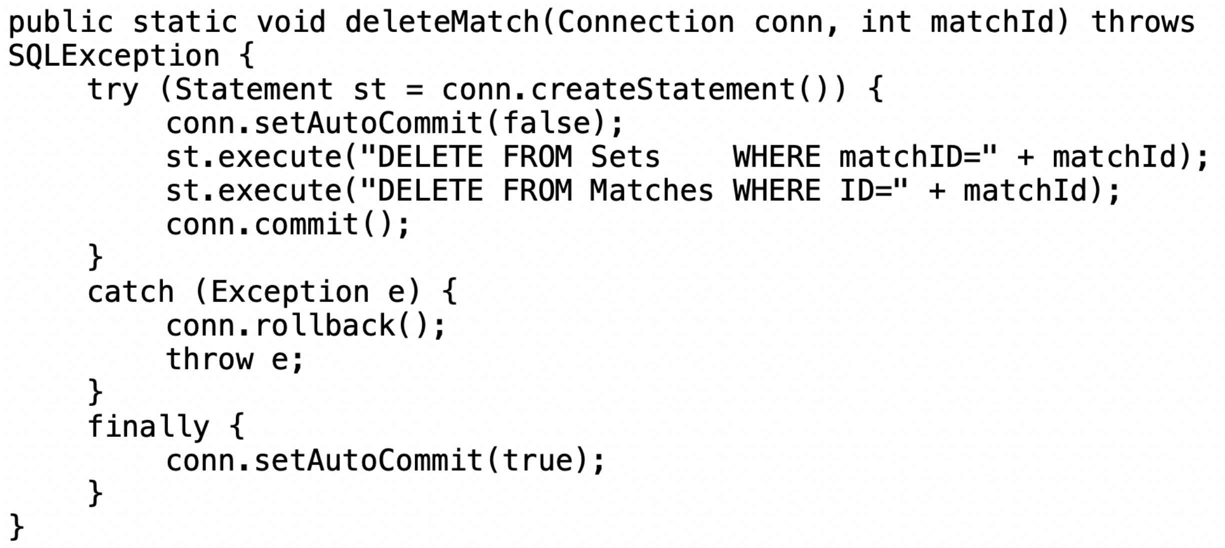
(c) Creating the trigger will fail because teams are already incorrectly registered to divisions.

(d) Creating the trigger will not correct the data in the relation.

(e) The trigger would also work correctly as an AFTER trigger.

3 (MSc ONLY) Database programming (5 points)

Consider the Java code in Figure 3.



```
public static void deleteMatch(Connection conn, int matchId) throws
SQLException {
    try (Statement st = conn.createStatement()) {
        conn.setAutoCommit(false);
        st.execute("DELETE FROM Sets    WHERE matchID=" + matchId);
        st.execute("DELETE FROM Matches WHERE ID=" + matchId);
        conn.commit();
    }
    catch (Exception e) {
        conn.rollback();
        throw e;
    }
    finally {
        conn.setAutoCommit(true);
    }
}
```

Figure 3: Code for deleting match information from the `Matches` and `Sets` relations.

Select the true statements:

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

(b) The code is using transactions correctly.

(c) The use of transactions in this code is unnecessary.

(d) The try-with block will automatically close the statement, which is why the code does not need to call st.close().

(e) The code is using Object Relational Mapping.

4 ER Diagrams and Normalization (25 points)

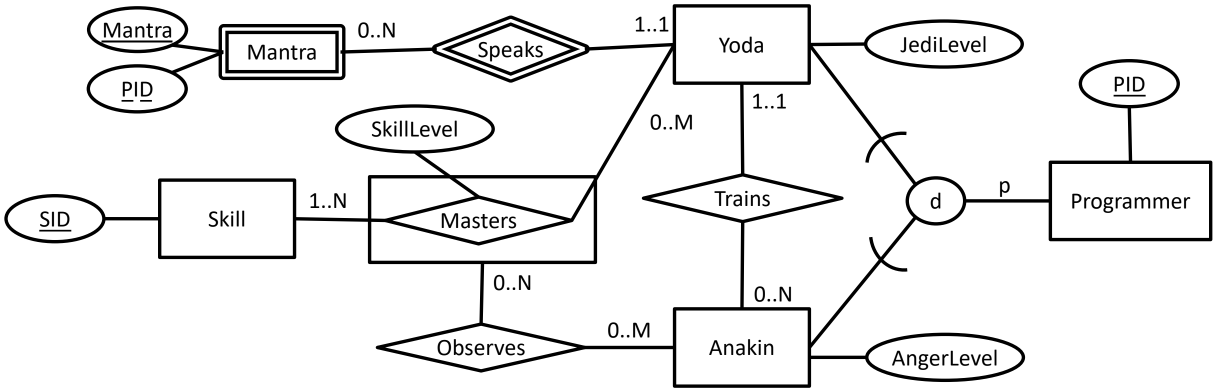


Figure 4: ER Diagram for a programmer training facility database.

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

(a) Every programmer has mastered at least one skill.

(b) Every Yoda has mastered at least one skill.

(c) Every Anakin is connected to at least one skill via some relationship(s).

(d) Every Anakin is connected to at least on mantra via some relationship(s).

(e) Anakins can only observe their Yoda trainer mastering skills.

(f) No two Yodas can have the same mantra.

b) Write SQL DDL commands to create a programmer training facility database based on the ER diagram in Figure 4 using the methodology given in the textbook and lectures. The DDL script *must run* in PostgreSQL. The relations must include all relevant primary key, candidate key, foreign key and NOT NULL constraints. Constraints that cannot be enforced with these standard constraints should be omitted. All attributes should be of type INTEGER except Mantra, which should be of type VARCHAR.

c) Write an ER diagram for a programming language database. The diagram should clearly show the entities, relationships and participation constrains described below. Attributes are only important if they are mentioned in this description; you should not add other attributes. Follow precisely the notation presented in the textbook and lectures.

- A programming language has a unique name.
- Programming languages are divided into functional, OO and query languages. Some may be of more than one sub-type and some may be of none.
- Programming languages may have multiple predecessors.
- Authors may have contributed to many languages, but languages must have at least one contributing author.
- Author sometimes criticise the contributions of authors to a particular programming language (sometimes they even criticise themselves, although this is not relevant for the ER diagram).

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

$$\begin{aligned} A &\rightarrow BCDE \\ ABC &\rightarrow E \\ E &\rightarrow B \\ CD &\rightarrow E \end{aligned}$$

Select the true statements:

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

(b) $ABC \rightarrow E$ is a trivial functional dependency.

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

(d) The relation can be normalized to BCNF without losing functional dependencies (excluding trivial, unavoidable, and derivable dependencies) .

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

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

Normalize R to the highest possible normal form (3NF or BCNF), based on functional dependencies, while allowing all functional dependencies (excluding trivial, unavoidable, and derivable 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 phone calls:

Calls(callID, callerID, receiverID, date, duration, <detailed logging info>)

For each of the queries below, select the index that a good query optimiser is most likely to use for the Sales table to process the query. Assume that all indexes are *unclustered* B+-trees. Also, assume that the relation has call data for the last 30 years, that each day has roughly the same number of calls, that there are 10 million possible callers/receivers, and that most calls last minutes (duration measures milliseconds). For each case, select the best index (or indexes) for the query, or select "no index" if a full table scan would yield better performance than any of the possible indexes.

(a) Calls(callID)

(b) Calls(callerID)

(c) Calls(receiverID)

(d) Calls(date)

(e) Calls(date, duration)

(f) No index

The queries are:

Query 1

```
select *
from Calls
where date = ’12-12-2010’;
```

Query 2

```
select avg(duration)
from Calls
group by date;
```

Query 3

```
select callID, date
from Calls;
```

Query 4

```
select avg(callID)
from Calls
where callerID = 453543 and receiverID = 1654332;
```

6 Hardware and DBMS Design (10 points)

a) Select the correct statements below:

(a) Magnetic tapes are useful for archival storage (e.g., database backups).

(b) Utilisation of the L2 cache is not important in a disk-based database management system.

(c) Traditional relational systems can scale-up infinitely.

(d) By definition, a NoSQL system can not implement the SQL query language.

b) Discuss briefly the suitability of eventual consistency for a typical banking application.

7 Data Systems for Analytics (10 points)

a) Select the correct statements below:

(a) Big data is always correct data.

(b) Most machine learning algorithms will learn the biases present in the training data.

(c) Document stores are the best tool for big data analytics applications.

(d) Although Hadoop is fairly recent technology, the MapReduce concept is very old.

b) Contrast briefly the different types of *Value* in Big Data applications.

Final Exam May 2020

1. **1a) SQL**

   1a) Write your SQL query here:

   Notes: (not included in XML)

   - `See idb-may-2020-SQL.sql`

2. **1a) Numerical answer**

   1a) Run the query of the previous question and paste the result here (an integer):

   - 12 ✓

3. **1b) SQL**

   1b) Write your SQL query here:

   Notes: (not included in XML)

   - `See idb-may-2020-SQL.sql`

4. **1b) Numerical answer**

   1b) Run the query of the previous question and paste the result here (an integer):

   - 34 ✓

5. **1c) SQL**

   1c) Write your SQL query here:

   Notes: (not included in XML)

   - `See idb-may-2020-SQL.sql`

6. **1c) Numerical answer**

   1c) Run the query of the previous question and paste the result here (an integer):

   - 62 ✓

7. **1d) SQL**

   1d) Write your SQL query here:

   Notes: (not included in XML)

   - `See idb-may-2020-SQL.sql`

8. **1d) Numerical answer**

   1d) Run the query of the previous question and paste the result here (an integer):

- 54 ✓

9. **1e) SQL**

   1e) Write your SQL query here:

   Notes: (not included in XML)

   - `See idb-may-2020-SQL.sql`

10. **1e) Numerical answer**

    1e) Run the query of the previous question and paste the result here (an integer):

    - 34 ✓

11. **1f) SQL**

    1f) Write your SQL query here:

    Notes: (not included in XML)

    - `See idb-may-2020-SQL.sql`

12. **1f) Numerical answer**

    1f) Run the query of the previous question and paste the result here (an integer):

    - 30 ✓

13. **1g) SQL**

    1g) Write your SQL query here:

    Notes: (not included in XML)

    - `See idb-may-2020-SQL.sql`

14. **1g) Numerical answer**

    1g) Run the query of the previous question and paste the result here (an integer):

    - 1 ✓

15. **1h) SQL**

    1h) Write your SQL query here:

    Notes: (not included in XML)

    - `See idb-may-2020-SQL.sql`

16. **1h) Numerical answer**

    1h) Run the query of the previous question and paste the result here (an integer):

- 1637 ✓

17. **2) [BSc only] SQL programming**

    2) [BSc only] Select the true statements:

    (a) Check 1 can be replaced by a constraint on the `TeamsInDivisions` relation. (0%)
    (b) Check 2 can be replaced by a constraint on the `TeamsInDivisions` relation. (50%)
    (c) Creating the trigger will fail because teams are already incorrectly registered to divisions. (0%)
    (d) Creating the trigger will not correct the data in the relation. (50%)
    (e) The trigger would also work correctly as an AFTER trigger. (0%)

18. **3) [MSc only] Java programming**

    3) [MSc only] Select the true statements:

    (a) The code is safe against SQL injection attacks. (33.33333%)
    (b) The code is using transactions correctly. (33.33333%)
    (c) The use of transactions in this code is unnecessary. (0%)
    (d) The try-with block will automatically close the statement, which is why the code does not need to call st.close(). (33.33333%)
    (e) The code is using Object Relational Mapping. (0%)

19. **4a) ER Diagram Interpretation**

    4a) Select the true statements:

    (a) Every programmer has mastered at least one skill. (0%)
    (b) Every Yoda has mastered at least one skill. (50%)
    (c) Every Anakin is connected to at least one skill via some relationship(s). (50%)
    (d) Every Anakin is connected to at least on mantra via some relationship(s). (0%)
    (e) Anakins can only observe their Yoda trainer mastering skills. (0%)
    (f) No two Yodas can have the same mantra. (0%)

    Notes: In (e) it was unclear whether this question only applied to the Observes relationship, or to Anakins in general. In either case, however, the correct answer would be FALSE. In (f), the question can either be understood to refer to the actual mantra (text), in which case the answer would be FALSE, or it can be understood to refer to the weak entity, in which case the answer would be TRUE. As a result, item (f) was omitted from grading.

20. **4b) SQL DDL**

    4b) Write your DDL for creating the database. You can also write any extra assumptions, attributes or explanations you feel are necessary.

Notes: (not included in XML)

- `See idb-may-2020-DDL.sql; to be written but will have 7 tables.`

21. **4c) ER Diagram Creation**

    4c) Upload the ER diagram or deliver a hand drawing at the exam.

    Notes: (not included in XML)

    - `See idb-may-2020-ER.jpg.`

    Note: The solution given, which was the best solution in the exam, was written by Alexandra Waldau nd posted with permission. It models the predecessor relation correctly as a relationship from programming languages to themselves. However, making a new predecessor entity and a many-to-many relationship to programming languages was also accepted as correct.

22. **4d) Normalisation**

    4d) Select the true statements:

    (a) A is the only (candidate) key of R. (50%)
    (b) $ABC \to E$ is a trivial functional dependency. (0%)
    (c) Normalizing to 3NF/BCNF results in exactly two relations. (0%)
    (d) The relation can be normalized to BCNF without losing functional dependencies (excluding trivial, unavoidable, and derivable dependencies) . (50%)

23. **4e) Normalisation**

    4e) Write down the normalized relations.

    Notes: (not included in XML)

    - `No decomposition, the relation is already in BCNF.`

24. **5a) Query 1**

    5a) Selection for Query 1:

    (a) Calls(callID) (0%)
    (b) Calls(callerID) (0%)
    (c) Calls(receiverID) (0%)
    (d) Calls(date) (100%)
    (e) Calls(date, duration) (0%)
    (f) No index (0%)

25. **5b) Query 2**

    5b) Selection for Query 2:

(a) Calls(callID) (0%)
(b) Calls(callerID) (0%)
(c) Calls(receiverID) (0%)
(d) Calls(date) (0%)
(e) Calls(date, duration) (100%)
(f) No index (0%)

26. **5c) Query 3**

    5c) Selection for Query 3:

    (a) Calls(callID) (0%)
    (b) Calls(callerID) (0%)
    (c) Calls(receiverID) (0%)
    (d) Calls(date) (0%)
    (e) Calls(date, duration) (0%)
    (f) No index (100%)

    Note: Using an index on Calls(callID) or and index on Calls(date) is incorrect, as using the index would only increase (significantly) the work needed to be done. Using both indexes together could work, as then the two indexes could be joined, reducing the total IO cost at the expense of CPU work for a hash-join. Full credit was given for the latter solution.

27. **5d) Query 4**

    5d) Selection for Query 4:

    (a) Calls(callID) (0%)
    (b) Calls(callerID) (50%)
    (c) Calls(receiverID) (50%)
    (d) Calls(date) (0%)
    (e) Calls(date, duration) (0%)
    (f) No index (0%)

28. **6a) Hardware and DBMS Design**

    6a) Select the true statements:

    (a) Magnetic tapes are useful for archival storage (e.g., database backups). (100%)
    (b) Utilisation of the L2 cache is not important in a disk-based database management system. (0%)
    (c) Traditional relational systems can scale-up infinitely. (0%)
    (d) By definition, a NoSQL system can not implement the SQL query language. (0%)

29. **6b) Hardware and DBMS Design**

    6b) Write your reflections here:

Notes: (not included in XML)

- `Banking applications usually work with structured data and require ACID consistency, which is not only about individual values but relationships between values, including auditable counters and such. Eventual consistency is a much simpler concept, (a) focusing solely on individual records as discussed in the CAP theorem, and (b) even allowing inconsistencies in the values of those records. As such, Eventual Consistency is not at all suitable in banking.`

30. **7a) Data Systems for Analytics**

7a) Select the true statements:

(a) Big data is always correct data. (0%)
(b) Most machine learning algorithms will learn the biases present in the training data. (50%)
(c) Document stores are the best tool for big data analytics applications. (0%)
(d) Although Hadoop is fairly recent technology, the MapReduce concept is very old. (50%)

31. **7b) Data Systems for Analytics**

7b) Write your reflections here:

Notes: (not included in XML)

- `We discussed two types of values: financial and societal. Most companies would desire financial value from their big data applications, for example in the form of improved operations and profits. Governmental and scientific institutions are more likely to go for societal value, for example in the form of improved public health or scientific knowledge. Of course, these may mix in arbitrary proportions, and societal value often results in financial value.`