Introduction to Database Design
MSc and BSc Exams

Björn Thór Jónsson

2019-05-29

Instructions

This file contains the questions from the May 2019 exam that remain relevant in the current version of the course. Some of today's learning outcomes are not represented in this exam.

Database description

In this exam you will work with the database `cargo`. To start working with the database, import/run `idb-may-2019-DB.sql` found in LearnIT using the PostgreSQL DBMS on your laptop. The database contains information on flights, aircrafts and airports. Note that you may already have a database called `cargo` on your server, but the exam version is different, so you must import/run the `idb-may-2019-DB.sql` script.

```
acgroup(ag, agfullname)
aircraft(actype, actypefullname, capacity, ag)
airport(airport, country)
country(country, region)
flights(id, al, dep, arr, actype, start_op, end_op, ...)
```

Most attributes are self-explanatory, and primary and foreign keys are generally defined as expected. Countries are represented by a two-letter country code; you will need to use the codes for Denmark (DK), Germany (DE) and Netherlands (NL). Regions are essentially continents, also represented by two-letter codes; you will need to use the codes for Europe (EU) and Asia (AS).

The `flights` relation is the most complicated one and warrants further explanation:

- There are more attributes in the relation, but these are the ones you will need in the exam.
- The `al` attribute is a two-letter code for the airline in question. The actual names of the airlines are not relevant.
- The `dep` and `arr` attributes are the departure and arrival airports, respectively. Both attributes have foreign key constraints to the airport relation.
- The `actype` is the type of the aircraft, with a foreign key constraint to the aircraft relation.
- The `start_op` and `end_op` attributes indicate how long the flight has been (or was) running. In these queries we make no distinction between flight routes that are currently running and flight routes that have stopped.
- You can use the following expression to compute the total running time of a flight route in days: `DATEDIFF(end_op, start_op)`.

1 SQL (40 points)

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

(a) In Denmark, there are 2 registered airports. How many airports are registered in Germany?

(b) In Asia, there are 57 airports that have both departing and arriving flights. How many airports are in Europe have both departing and arriving flights?

(c) The average number of days that a flight route has been running is 42.77. For how many days has the longest running flight route been running?

(d) There are 6126 flights that a) depart from an airport within Europe and b) have an aircraft capacity of more than 300 passengers. How many flights with more capacity than 300 passengers depart from an airport within Asia?

(e) Each aircraft has a registered aircraft group (aircraft.ag). The smallest such aircraft group has 2 members. How many members does the largest group have?

Hint: Using a view can simplify the query significantly. If you do, include the view creation statement in your answer.

(f) According to the flights relation, there are 124 airports with more departing flights than arriving flights. How many airports have more arriving flights than departing flights?

(g) How many freight flights (`ag = 'F'`) land in a different country from where they departed, but in the same region?

(h) Only 1 airline has flights departing from every registered airport in Denmark. How many airlines have flights departing from every registered airport in the Netherlands?

Enter each query, as well as the numerical answer to each question, in LearnIT. The query must work for any instance of the schema. The query 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. In particular, a sequence of several queries that allow you to answer the question will not receive full points (but subqueries are fine). If you are unable to write a working query you can still submit your attempt, and it may be given partial points; in this case include a brief description of the problem with the query.

3 ER Diagrams and DDL (25 points)

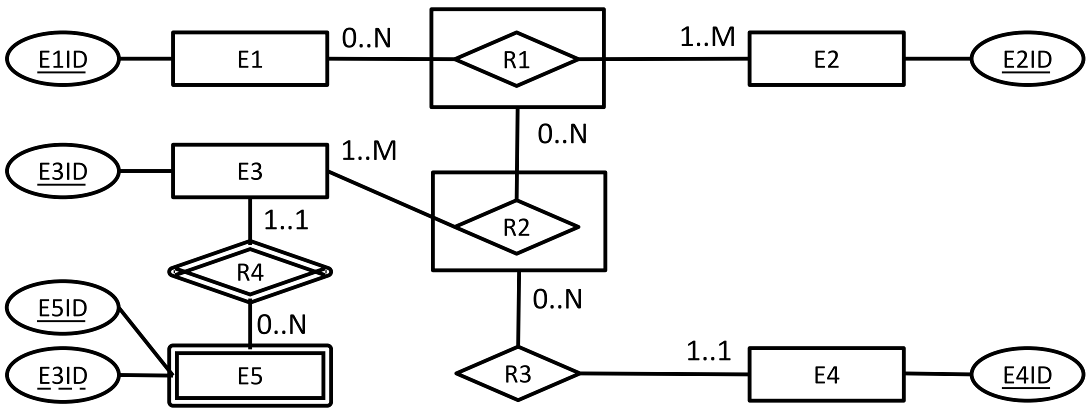

Figure 1: Generic ER Diagram. Attributes omitted.

a) Figure 1 shows a generic ER diagram. Select the true statements:

(a) Every E2 entity must participate in R1.

(b) An E1 entity is always indirectly related to some E4 entity through R1, R2 and R3.

(c) Every E3 has one corresponding E5 entity.

(d) The key for a table created for relationship R3 would contains the attributes E1ID, E2ID, E3ID and E4ID.

(e) The key for a table created for entity E5 would contain the attributes E3ID and E5ID.

(f) A table created for relationship R2 would have a foreign key to the table for entity E1.

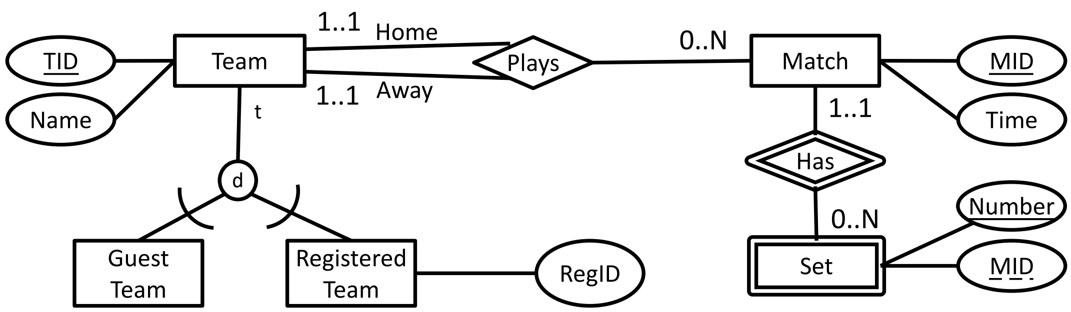

Figure 2: ER Diagram for volleyball database.

b) The ER diagram in figure 2 shows a simple volleyball database. Select the true statements.You should base your answer **only** on what data is allowed and not allowed by the ER diagram:

(a) Every team can play every other team only once.

(b) Each match has exactly one home team.

(c) Only registered teams can play in a match.

(d) A match has exactly one set.

(e) A guest team can have the same name as a registered team.

(f) No information indicates which team wins each match.

c) Write SQL DDL to create the chess database based on the ER diagram in Figure 2. The relations must include all key and foreign key constraints. Make reasonable assumptions on the attribute types.

d) Write an ER diagram for an auction database. The diagram should clearly show the entities, relationships and participation constrains described below. Use the notation presented in the textbook and lectures. Attributes are not important. If you need to make additional assumptions put them in the box below.

- Every item sold is either an artwork or an object, never both.
- Objects are produced by one producer.
- Artworks are created by at least one artist, but artists can collaborate on artworks.
- Experts may verify whether an artist took part in creating an artwork.

4 Normalisation (10 points)

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

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

Select the true statements:

(a) AB is the only key of R.

(b) $B \rightarrow D$ is an unavoidable functional dependency.

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

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

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

$$\begin{aligned} CD &\rightarrow AB \\ ADE &\rightarrow E \\ A &\rightarrow E \\ AB &\rightarrow ABCDE \end{aligned}$$

Select the true statements:

(a) AB is the only key of R.

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

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

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

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

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

Normalize R to 3NF or BCNF and write down the resulting relations here:

6 Index Selection (10 points)

Consider the following relation with information on employees:

Person(id, name, birthyear, children, <many long attributes>)

For each of the queries below, select the index(es) that a good query optimiser is most likely to use to process the query. You should assume that id, birthyear and children are integer values, while name is a string. Also, assume that attribute values correspond to reality and that the query optimizer has basic (approximate) statistics, such as smallest and largest value of each attribute. You may select 1 or more options (including "no index"). In particular, consider whether the index is covering.

(a) Person(id)

(b) Person(birthyear)

(c) Person(children)

(d) Person(birthyear, children)

(e) Person(birthyear, children, name)

(f) No index

The queries are:

Query 1

```
select id, name
from Person
where children > 5;
```

Query 2

```
select *
from Person
where birthyear > (select max(children) from Person);
```

Query 3

```
select id
from Person
where children < 50;
```

Query 4

```
select name
from Person
where children > 5 and birthyear = 1935;
```

Final Exam May 2019

1. **1a) SQL**

   1a) Write your SQL query here:

   Notes: (not included in XML)

   - See `idb-may-2019-SQL.sql`

2. **1a) Numerical answer**

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

   - 17 ✓

3. **1b) SQL**

   1b) Write your SQL query here:

   Notes: (not included in XML)

   - See `idb-may-2019-SQL.sql`

4. **1b) Numerical answer**

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

   - 185 ✓

5. **1c) SQL**

   1c) Write your SQL query here:

   Notes: (not included in XML)

   - See `idb-may-2019-SQL.sql`

6. **1c) Numerical answer**

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

   - 1572 ✓

7. **1d) SQL**

   1d) Write your SQL query here:

   Notes: (not included in XML)

   - See `idb-may-2019-SQL.sql`

8. **1d) Numerical answer**

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

- 185 ✓

9. **1e) SQL**

   1e) Write your SQL query here:

   Notes: (not included in XML)

   - See `idb-may-2019-SQL.sql`

10. **1e) Numerical answer**

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

    - 24 ✓

11. **1f) SQL**

    1f) Write your SQL query here:

    Notes: (not included in XML)

    - See `idb-may-2019-SQL.sql`

12. **1f) Numerical answer**

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

    - 182 ✓

13. **1g) SQL**

    1g) Write your SQL query here:

    Notes: (not included in XML)

    - See `idb-may-2019-SQL.sql`

14. **1g) Numerical answer**

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

    - 518 ✓

15. **1h) SQL**

    1h) Write your SQL query here:

    Notes: (not included in XML)

    - See `idb-may-2019-SQL.sql`

16. **1h) Numerical answer**

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

- 1 ✓

17. **3a) Generic ER diagram**

    3a) Select the true statements:

    (a) Every E2 entity must participate in R1. (0%)
    (b) An E1 entity is always indirectly related to some E4 entity through R1, R2 and R3. (50%)
    (c) Every E3 has one corresponding E5 entity. (0%)
    (d) The key for a table created for relationship R3 would contain the attributes E1ID, E2ID, E3ID and E4ID. (0%)
    (e) The key for a table created for entity E5 would contain the attributes E3ID and E5ID. (50%)
    (f) A table created for relationship R2 would have a foreign key to the table for entity E1. (0%)

18. **3b) Volleyball ER diagram**

    3b) Select the true statements:

    (a) Every team can play every other team only once. (0%)
    (b) Each match has exactly one home team. (33.33333%)
    (c) Only registered teams can play in a match. (0%)
    (d) A match has exactly one set. (0%)
    (e) A guest team can have the same name as a registered team. (33.33333%)
    (f) No information indicates which team wins each match. (33.33333%)

19. **3c) DDL**

    3c) 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-2019-DDL.sql`

20. **3d) ER-diagram creation**

    3d) Upload the ER diagram or deliver a hand drawing at the exam.

    Notes: (not included in XML)

    - See `idb-may-2019-ER.pdf`

21. **4a) Normalisation**

    4a) Select the true statements:

    (a) AB is the only key of R. (50%)
    (b) $B \rightarrow D$ is an unavoidable functional dependency. (0%)

(c) Normalizing to BCNF results in exactly two relations. (0%)
(d) The relation $Z(A, B, C)$ is in BCNF. (50%)

22. **4b) Normalisation**

    4b) Select the true statements:

    (a) AB is the only key of R. (0%)
    (b) $ADE \to E$ is a trivial functional dependency. (33.33333%)
    (c) Normalizing to 3NF or BCNF results in exactly two relations. (33.33333%)
    (d) The relation $Z(A, B, C, D)$ is in BCNF. (33.33333%)

23. **4c) Normalisation**

    4c) Write down the normalized relation.

    Notes: (not included in XML)

    - `R1(C, B) R2(A, C, D, E)`

24. **6a) Query 1**

    6a) Selection for query 1:

    (a) Person(id) (0%)
    (b) Person(birthyear) (0%)
    (c) Person(children) (100%)
    (d) Person(birthyear, children) (0%)
    (e) Person(birthyear, children, name) (0%)
    (f) No index (0%)

25. **6b) Query 2**

    6b) Selection for query 2:

    (a) Person(id) (0%)
    (b) Person(birthyear) (0%)
    (c) Person(children) (100%)
    (d) Person(birthyear, children) (0%)
    (e) Person(birthyear, children, name) (0%)
    (f) No index (0%)

26. **6c) Query 3**

    6c) Selection for query 3:

    (a) Person(id) (0%)
    (b) Person(birthyear) (0%)
    (c) Person(children) (0%)
    (d) Person(birthyear, children) (0%)

(e) Person(birthyear, children, name) (0%)
(f) No index (100%)

27. **6d) Query 4**

6d) Selection for query 4:

(a) Person(id) (0%)
(b) Person(birthyear) (0%)
(c) Person(children) (0%)
(d) Person(birthyear, children) (0%)
(e) Person(birthyear, children, name) (100%)
(f) No index (0%)