# Introduction to Database Design
# BSc and MSc Exams

Björn Thór Jónsson
Johan von Tangen Sivertsen

2019-01-03

## Instructions

*Note: This file contains the questions from the January 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 `Games`. To start working with the database, import/run `idb-january-2019-DB.sql` found in LearnIT using the PostgreSQL DBMS on your laptop. The database contains information on (made-up) achievements and scores in computer games with the following schema. The data in the database is entirely fabricated except for the *names* of games.

```
Player(id,name,email)
Game(id,version,name,producer,releasedate)
Achievement(id,gameId,description)
Score(playerId,gameId,timeStamp,level,score)
PlayerAchievement(playerId,achievementId)
```

Primary and foreign key attributes are those whose names include `Id` (but no constraints are defined in the provided database). The first three relations have their first attribute as primary key. The last two relations have a composite primary key consisting of their two first attributes. An achievement is an accomplishment defined by the game, e.g., "complete the first 10 levels", or "play 100 times". The meaning of other relations and attributes should be self-explanatory.

# 1 SQL (40 points)

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

(a) Of all the players, 93 have an email address with gmail.com. How many have an email address with yahoo.dk?

(b) The Score relation contains 403 entries with a score smaller than 1000. How many entries have a score that is lower than the average score of all entries in the relation?

(c) The database does not have foreign keys defined, which can lead to data errors. How many entries in PlayerAchievement refer to an achievement that does not exist?

(d) Only 5 distinct players have both a registered score and a registered achievement in some game produced by "Electronix Arts". How many distinct players have both a registered score and a registered achievement in some game produced by "Codemasters"?

(e) The number of distinct players who have a registered score in some game, but no registered achievement in that same game, is 493. How many distinct players have a registered achievement in some game, but no registered score in that game?

(f) One player has played all the games named "Bioforge". How many players have played all the games named "Project Eden"?

(g) One game name is used by three different producers. How many game names are used by two different producers?

(h) How many pairs of games have the same name? Hint: The results of the previous problem should give you the answer, since a name with three games gives three pairs, and a name with two games gives one pair. The task here is to write a single query that counts all pairs, even in a case where a single name is used for many games.

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)

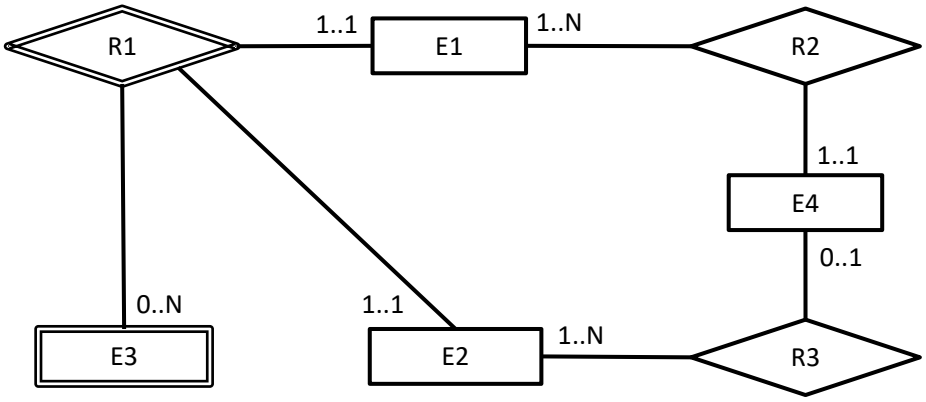


Figure 1: Generic ER Diagram. Attributes omitted.

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

(a) Every E2 entity participates in R3.

(b) An E3 is not uniquely identifiable by its own attributes.

(c) An E1 can be related through R1 to several E2.

(d) An E3 is always indirectly related to an E4 through R1 and R2.

(e) Every E4 that participates in R3 is related to exactly one E2.

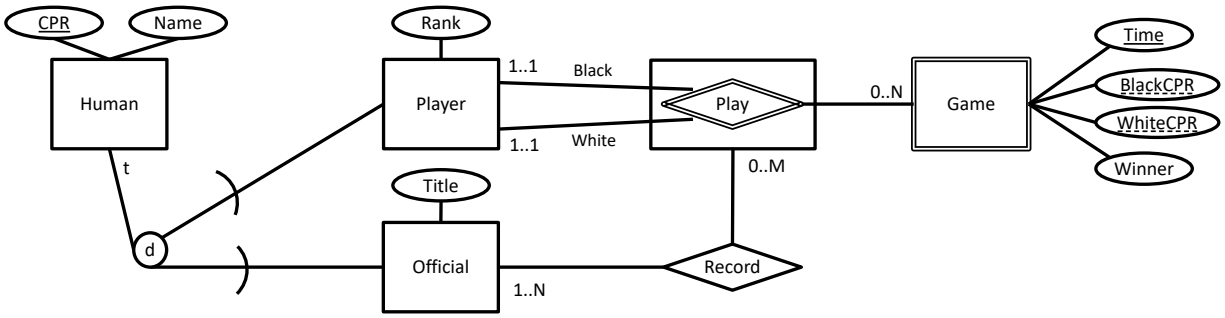


Figure 2: ER Diagram for chess database.

**b)** The ER diagram in figure 2 shows a chess tournament 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 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.

**c)** Write mysql 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 attributes in the ER diagram. Feel free to introduce additional attributes if you want to.

# 4 Normalisation (10 points)

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

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

Select the true statements:

(a) $AB$ is the only key of $R$.

(b) $AB \rightarrow C$ is an unavoidable functional dependency.

(c) Normalizing to 3NF (but not to BCNF) results in exactly two relations.

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

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

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

Select the true statements:

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

(b) $BCD \rightarrow D$ is a trivial functional dependency.

(c) Normalizing to 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:

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

Normalize $R$ to BCNF and write down the resulting relations here:

# 6 Index Selection (10 points)

Consider the following relation with information on employees:

Emp (id, name, age, sal, ...)

For each of the queries below, select the index(es) that a good query optimiser is likely to use to process the query. You may select 1 or more options (including "no index"). In the text box below the options, briefly explain your reasons. In particular, consider whether the index is covering.

(a) Emp(id)

(b) Emp(age)

(c) Emp(sal)

(d) Emp(age, sal)

(e) Emp(age, sal, id)

(f) Emp(age, sal, name)

(g) No index

The queries are:

**Query 1**

```
select id, name
from Emp;
```

**Query 2**

```
select id
from Emp
where age > (select max(sal) from Emp);
```

**Query 3**

```
select age
from Emp
where sal = 23;
```

**Query 4**

```
select sal
from Emp
where age > 35;
```

## Exam January 2019

1. **1 a). SQL**

   1 a) Write your SQL query here:

   Notes: (not included in XML)

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

2. **1 a). Numerical answer**

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

   - 87 ✓

3. **1 b). SQL**

   1 b) Write your SQL query here:

   Notes: (not included in XML)

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

4. **1 b). Numerical answer**

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

   - 967 ✓

5. **1 c). SQL**

   1 c) Write your SQL query here:

   Notes: (not included in XML)

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

6. **1 c). Numerical answer**

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

   - 2 ✓

7. **1 d). SQL**

   1 d) Write your SQL query here:

   Notes: (not included in XML)

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

8. **1 d). Numerical answer**

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

- 6 ✓

9. **1 e). SQL**

   1 e) Write your SQL query here:

   Notes: (not included in XML)

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

10. **1 e). Numerical answer**

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

    - 486 ✓

11. **1 f). SQL**

    1 f) Write your SQL query here:

    Notes: (not included in XML)

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

12. **1 f). Numerical answer**

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

    - 2 ✓

13. **1 g). SQL**

    1 g) Write your SQL query here:

    Notes: (not included in XML)

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

14. **1 g). Numerical answer**

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

    - 7 ✓

15. **1 h). SQL**

    1 h) Write your SQL query here:

    Notes: (not included in XML)

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

16. **1 h). Numerical answer**

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

- 10 ✓

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

    3a) Select the true statements:

    (a) Every E2 entity participates in R3. (0%)
    (b) An E3 is not uniquely identifiable by its own attributes. (33.33333%)
    (c) An E1 can be related through R1 to several E2. (33.33333%)
    (d) An E3 is always indirectly related to an E4 through R1 and R2. (33.33333%)
    (e) Every E4 that participates in R3 is related to exactly one E2. (0%)

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

    3b) Select the true statements:

    (a) Every player plays at least one game. (0%)
    (b) Every game has two players. (33.33333%)
    (c) An official is a player. (0%)
    (d) The same "play" relation can be recorded by multiple officials. (33.33333%)
    (e) A "play" relation must be recorded by at least one official. (33.33333%)

19. **3 c) 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-january-2019-DDL.sql`

20. **4 a) Normalisation**

    4a) Select the true statements:

    (a) $AB$ is the only key of $R$. (0%)
    (b) $AB \rightarrow C$ is an unavoidable functional dependency. (100%)
    (c) Normalizing to 3NF (but not to BCNF) results in exactly two relations. (0%)
    (d) The relation $Z(A, B, E, D)$ is in BCNF. (0%)

21. **4 b) Normalisation**

    4b) Select the true statements:

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

22. **4 c) Normalisation**

    4c) Write down the normalized relation.

    Notes: (not included in XML)

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

23. **6 a) Query 1**

    6 I) Selection for query 1:

    (a) Emp(id) (0%)
    (b) Emp(age) (0%)
    (c) Emp(sal) (0%)
    (d) Emp(age, sal) (0%)
    (e) Emp(age, sal, id) (0%)
    (f) Emp(age, sal, name) (0%)
    (g) No index (100%)

24. **6 b) Query 2**

    6 II) Selection for query 2:

    (a) Emp(id) (0%)
    (b) Emp(age) (50%)
    (c) Emp(sal) (50%)
    (d) Emp(age, sal) (0%)
    (e) Emp(age, sal, id) (0%)
    (f) Emp(age, sal, name) (0%)
    (g) No index (0%)

25. **6 c) Query 3**

    6 III) Selection for query 3:

    (a) Emp(id) (0%)
    (b) Emp(age) (0%)
    (c) Emp(sal) (100%)
    (d) Emp(age, sal) (0%)
    (e) Emp(age, sal, id) (0%)
    (f) Emp(age, sal, name) (0%)
    (g) No index (0%)

26. **6 d) Query 4**

    6 IV) Selection for query 4:

    (a) Emp(id) (0%)
    (b) Emp(age) (0%)

(c) Emp(sal) (0%)
(d) Emp(age, sal) (100%)
(e) Emp(age, sal, id) (0%)
(f) Emp(age, sal, name) (0%)
(g) No index (0%)