Institut for Matematik og Datalogi
Syddansk Universitet, Odense

10. november 2020
CW/RF

Eksaminatorier DM534 Uge 46/47

- Remember to read the relevant slides before attempting to solve the exercises
- The SQL commands to create a database with the Movies relation and all its tuples are available here: `https://www.db-fiddle.com/f/oEjzriS8QbnNTbpT4yuigf/112`
- You can use the DB fiddle website from above as a starting point to write and test your own SQL commands.

I: To be solved during exercise classes in Week 46

1. Given the following relation schema:

   Band(name: CHAR(20), formed_in: INTEGER)

   Which of the following are valid tuples of the Band relation?

   □ ('Foo Fighters', 1994)
   □ (1991, 'Incubus')
   □ ('Massive Attack')
   □ ('Disturbed', '1996')

2. The relation schema of task 1 together with the valid tuples define a relation instance. Visualize this relation instance as a table.

3. Given the following relation instance:

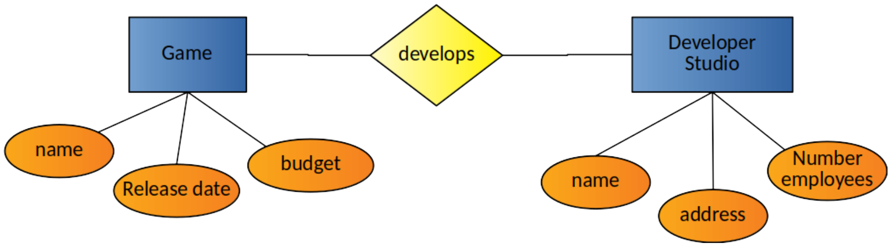
Movies

mid	**title**	**director**	**production_year**	**budget_usd**
0	'Matrix'	'The Wachowskis'	1999	63000000
1	'Raiders of the Lost Ark'	'Steven Spielberg'	1981	20000000
2	'The Shawshank Redemption'	'Frank Darabont'	1994	25000000
3	'Twilight'	'Robert Benton'	1998	20000000
4	'Dead Poets Society'	'Peter Weir'	1989	16400000
5	'Django Unchained'	'Quentin Tarantino'	2012	100000000
6	'Pulp Fiction'	'Quentin Tarantino'	1994	8500000
7	'Twilight'	'Catherine Hardwicke'	2008	37000000

Which of the following attribute sets are possible primary keys (i.e., the data instance above is consistent with choosing that set of attributes as the primary key)?

- □ {mid}
- □ {title}
- □ {director}
- □ {title, director}
- □ {director, production_year}

4. Given the following Entity-Relationship diagram:



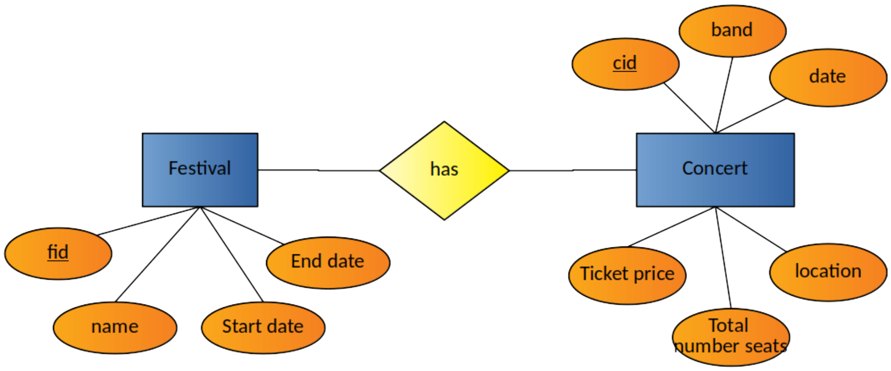
How could this ER diagram be modeled in the relational model? Provide the relation schemas. What would change if the relationship "develops" had an attribute "start date"? Note that there is no designation of keys for the entities. It is part of the task to consider what is a good choice of keys (this is a data modeling choice, for which there is no single "right answer").

5. You are given the relation instance defined in task 3.

   How many tuples do the relations resulting from the following relational operations contain? [Recall that in the relational model, relations are sets, so there can be no duplicates among (entire) tuples/rows.]

   - $\sigma_{\text{production\_year}>1994}(\text{Movies})$
   - $\pi_{\text{mid,title,director}}(\text{Movies})$
   - $\pi_{\text{director,production\_year}}(\text{Movies})$
   - $\pi_{\text{director}}(\text{Movies})$

6. You are given the following ER-diagram:



   and the corresponding relations:

   - Concert($\underline{\text{cid: INTEGER}}$, band: CHAR(20), date: CHAR(20), location: CHAR(20), total_number_seats: INTEGER, ticket_price: FLOAT)
   - Festival($\underline{\text{fid: INTEGER}}$, name: CHAR(20), start_date: CHAR(20), end_date: CHAR(20))

- FestivalHasConcert(festival: INTEGER , concert: INTEGER)

Here, underlined attributes are primary keys of the relation and dashed underlined attributes are foreign keys.

Specify the SQL command that creates the table corresponding to the Concert relation of task.

Also specify the SQL command that creates a corresponding table for the FestivalHasConcert relation.

[Recall that FestivalHasConcert's festival and concert attributes are foreign keys referencing the fid in the festival and cid in the concert relation, respectively.]

7. Specify an SQL command that deletes from the Movies table all movies that were not produced in 1994.

   [You can solve this task with the comparison operators that you saw in the lecture. Alternatively, use the not-equal-to operator `!=` which is also defined in SQL.]

8. You want to retrieve the titles of all movies from the Movies table of task 3 that were produced in 1994 and directed by Quentin Tarantino. Express this query in both of these two ways:

   - As an expression in relational algebra.
   - As an SQL command.

9. Consider the Movies table of task 3. Provide in SQL one INSERT and one DELETE command that can to be executed to change it into the following table.

Movies

mid	**title**	**director**	**production_year**	**budget_usd**
0	'Matrix'	'The Wachowskis'	1999	63000000
1	'Raiders of the Lost Ark'	'Steven Spielberg'	1981	20000000
5	'Django Unchained'	'Quentin Tarantino'	2012	100000000
7	'Twilight'	'Catherine Hardwicke'	2008	37000000
8	'The Lord of the Rings'	'Peter Jackson'	2001	93000000

10. Specify an SQL command without set operations (UNION or EXCEPT) that retrieves all movies from the Movie table, that have been produced before 1990 or that have a budget of at least 30 million USD. State the same query as an relational algebra expression.

11. Solve task 10 with SQL set operations (UNION or EXCEPT). State the same query as an relational algebra expression.

12. Specify an SQL command without set operations (UNION or EXCEPT) that retrieves all movies from the Movie table of task 3 that either

    - have been produced before 1990 with a budget of at least 30 million USD
    - or have been produced after 2010

    State the same query as an relational algebra expression.

13. Specify an SQL command that retrieves all pairs of movies from the Movies table of task 3 that have been directed by the same director.

    For instance, if two movies 'movie1' and 'movie2' were directed by the same director, the result relation should, amongst others, contain the following tuples:

    - (movie1, movie2)
    - (movie2, movie1)

    State the same query as an relational algebra expression.

14. Specify an SQL command that retrieves all pairs of movies (movie1, movie2) from the Movies table of task 3, with movie1's budget exceeding the budget of movie2.

    For instance, if a movie 'movie5' has a larger budget than 'movie8', the result relation should contain, amongst others, the following tuple:

    - (movie5, movie8)

    State the same query as an relational algebra expression.

II: To be solved at home before exercise classes in Week 47

1. Which of the following statements are true (multiple possible)?

   □ The result of applying a relational algebra operator to a relation instance is another relation instance.

   □ Entities of the ER-diagram can not be described by relations in the data model.

   □ A relation instance needs to contain at least one tuple.

   □ Integrity constraints are specified when querying the database.

   □ Primary keys and foreign keys are types of integrity constraints.

   □ The relational selection operator always returns a relation instance with fewer tuples.

   □ The relational projection operator may return a relation instance with fewer tuples.

   □ The SQL UNION operator can be applied to two relation instances, if they have the same number of attributes.

2. Joins are compound operators which we did not cover in the lecture. They are useful for joining (combining) the data contained in multiple relations together. In relational algebra, the *conditional join* operator $\bowtie_C$ (also called the θ-join operator) is defined by

   $$R1 \bowtie_C R2 = \sigma_C(R1 \times R2)$$

   In words: A conditional join can be calculated by first computing the cross product of the two relations $R1$ and $R2$, followed by a selection using the condition C.

   We now look at the following conditional join on relations from task I.6:

   $$\text{Festival} \bowtie_{\text{fid}=\text{festival}} \text{FestivalHasConcert}$$

   This join combines the two relations Festival and FestivalHasConcert using the attribute fid of the Festival relation, and the festival attribute of the FestivalHasConcert relation. Specifically, it combines those tuples of the two relations for which the equality condition holds.

   What is the relation schema of the result relation?

3. Specify an SQL command that calculates the conditional join given in task II.2.

4. Specify an SQL command that calculates the following nested relational operation involving two conditional joins:

$$\text{Festival} \bowtie_{\text{fid}=\text{festival}} \text{FestivalHasConcert} \bowtie_{\text{concert}=\text{cid}} \text{Concert}$$

   Hint 1: This compound operation can be expressed using basic relational operators as

$$\sigma_{\text{fid}=\text{festival}}(\sigma_{\text{concert}=\text{cid}}(\text{Festival} \times \text{FestivalHasConcert} \times \text{Concert}))$$

   Hint 2: You can start with and extend the SQL command of task II.3.