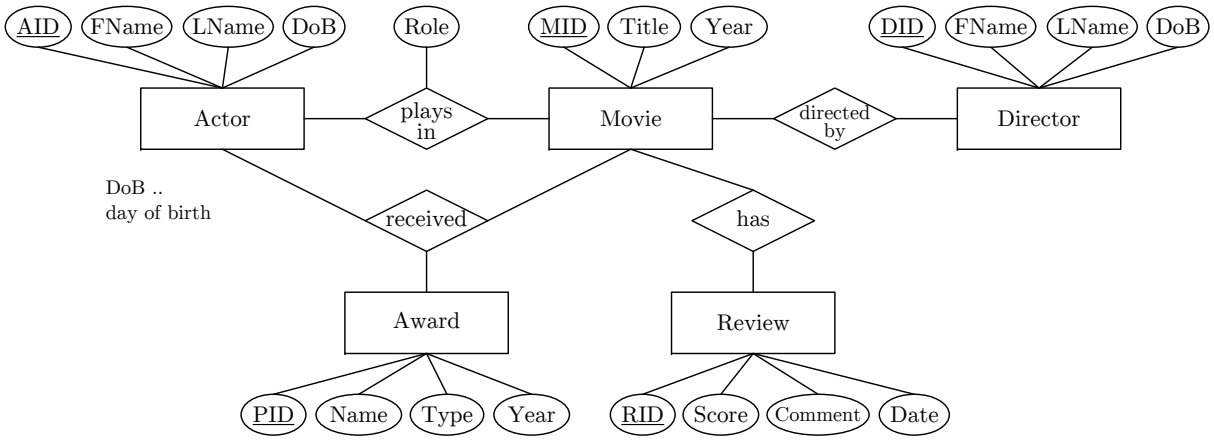
Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

May 20, 2021

Exam INF.01017UF Data Management (Winter 2020/21, V2a)

Important notes: The working time is 90min, and lecture materials or any kind of mobile devices are not allowed. Please, make sure to put your *name* and *matriculation number* on the top right of the first page of the task description, and each additional piece of paper. You may give the answers in English or German, written directly into the task description.

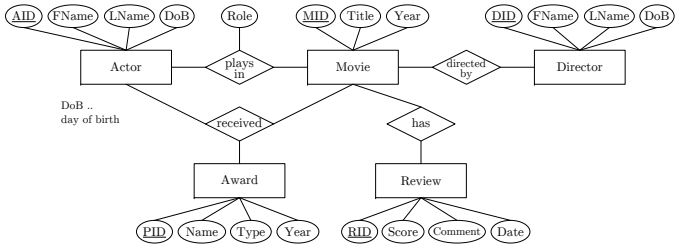
Task 1 Data Modeling (25 points)



(a) Given the Entity-Relationship (ER) diagram above, specify the cardinalities in Modified Chen notation based on the following information. (**9 points**)

- An actor may play roles in an arbitrary number of movies (including none), and every movie has a cast of at least one but potentially many actors.
- A movie is directed by exactly one director, and a single director might produce (i.e., direct) an arbitrary number of movies.
- A movie review (with score, text comment, and date) refers to exactly one movie, but there can be 0, 1, or many reviews per movie.
- Actors may receive multiple awards (e.g., best actress, best supporting access) for a specific movie. A single actor may receive multiple awards for a single movie (up to 8), but receives a specific award only for exactly one movie. A single award (e.g., best ensemble) for a single movie can be awarded to one or multiple actors.

(b) Map the given ER diagram into a relational schema in third normal form, including data types, primary keys, and foreign keys. Your schema should also ensure that each movie has an associated director, and each review refers to a movie. Note that you only need to provide the final schema and there is no need to explain the normal forms. (**12 points**)



(c) Assume an independent relation Movies(MID, Title, Year) with MID and Title each being unique and defined (not null). List below all super keys, all candidate keys, and select an appropriate primary key. Use (a,b,c) to indicate compound keys. (**4 points**)

- Super Keys:
- Candidate Keys:
- Primary Key:

Task 2 Structured Query Language (30 points)

Movies

			[min]	[Mio $]	[Mio $]	
MID	**Title**	**Year**	**Length**	**Budget**	**Revenue**	**GID**
1	The Matrix	1999	136	63	455	2
3	Hangover	2009	100	35	470	1
2	Fast and Furious	2001	106	40	210	3
7	Passengers	2016	116	130	300	2
4	Horrible Bosses	2011	98	35	210	1
5	The Hunger Games	2012	142	80	700	2
6	Draft Day	2014	110	25	30	4
8	The Post	2017	116	50	180	5

Genres

GID	**Name**
1	Comedy
6	Romance
2	Science Fiction
3	Action
4	Sports Drama
5	Historical Drama
7	Documentary

(a) Given the Movies and Genres tables above, compute the results for the following three queries: (**15 points**)

```
Q1: SELECT M.Title, M.Year, G.Name
      FROM Movies M, Genres G
      WHERE M.GID = G.GID
        AND (G.Name LIKE '% Drama'
        OR M.Length BETWEEN 130 AND 140)


Q2: SELECT Title, Year
      FROM Movies WHERE Year > 2010
    INTERSECT
    SELECT Title, Year
      FROM Movies WHERE Revenue > 250



Q3: SELECT Name, round(avg(Revenue)) --avg=sum/count
      FROM Movies M JOIN Genres G ON (M.GID=G.GID)
      GROUP BY Name
      ORDER BY avg(Revenue) DESC
```

(b) Given the Movies and Genres tables above, write SQL queries to answer the following questions (in a way that is independent of the shown data): (**15 points**)

- Q4: Which movies belong to the genre "Science Fiction" (return the Title and Year, sorted in ascending order of Title)?

- Q5: Which movie from the years 2005-2015 (both inclusive) yielded the maximum Revenue (return the Title of this movie and its Revenue)?

- Q6: Compute the number of movies associated with each genre, including genres without any movies (return the genre Name, and count)?

Task 3 Query Processing (15 points)

(a) Assume relations $R(a, b, c)$ and $S(d, e)$, and indicate in the table below whether or not the two relational algebra expressions per row are equivalent in bag semantics (✓for equivalent, $\times$ for non-equivalent). For non-equivalent expressions, briefly explain why. (**3 points**)

Expression 1	**Expression 2**	**Equivalent? Why Not?**
$(\sigma_{b<7}(R)) \cup (\sigma_{b>7}(R))$	$\sigma_{b=7}(R)$	
$\sigma_{c=3}(\sigma_{a=e}(R \times S))$	$(\sigma_{c=3}(R)) \bowtie_{a=e} S$	
$\pi_{b,d}(R \bowtie_{a=e} S)$	$(\pi_{a,b}(R)) \bowtie_{a=e} (\pi_{d,e}(S))$	

(b) Draw two logical query trees for queries Q2 and Q3 from Task 2(a). (**6 points**)

(c) Describe the conceptual ideas of a nested-loop join, and a hash join. Furthermore, assume $R \bowtie S$ with cardinalities $N = |R|$ and $M = |S|$, and enter the space and time complexity of these operators (in the open-next-close iterator model) in the table below. (**6 points**)

Operator	Time Complexity	Space Complexity
Nested Loop Join		
Hash Join		

Task 4 Transaction Processing (10 points)

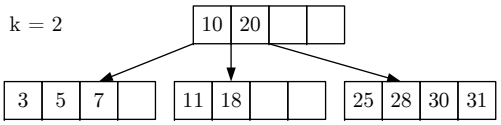
(a) Explain the concept of a database transaction log, and how it helps to ensure Atomicity and Durability of changes made by uncommitted and committed transactions on failures. (**6 points**)

(b) Indicate in the table below, which operation schedules are equivalent (✓for equivalent, × for non-equivalent). The notation $r_1(a)$ and $w_2(b)$ refers to the read of object a by transaction T_1 and the write of object b by T_2. (**4 points**)

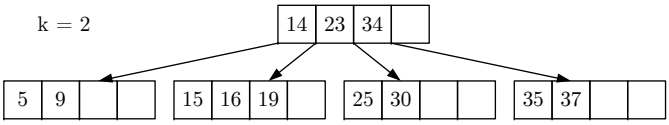
Schedule 1	Schedule 2	Equivalent?
$\{r_1(a), w_1(a), r_2(b), w_2(b)\}$ $\{r_1(c), w_1(c), r_2(c), r_2(d), w_2(d)\}$ $\{r_1(e), w_1(e), w_2(e), w_2(f)\}$ $\{r_1(g), r_1(h), r_2(g), r_2(h), w_2(h)\}$	$\{r_1(a), r_2(b), w_1(a), w_2(b)\}$ $\{r_1(c), r_2(c), r_2(d), w_1(c), w_2(d)\}$ $\{r_1(e), w_2(e), w_1(e), w_2(f)\}$ $\{r_2(g), r_2(h), r_1(h), r_1(g), w_2(h)\}$	

Task 5 Physical Design (15 points)

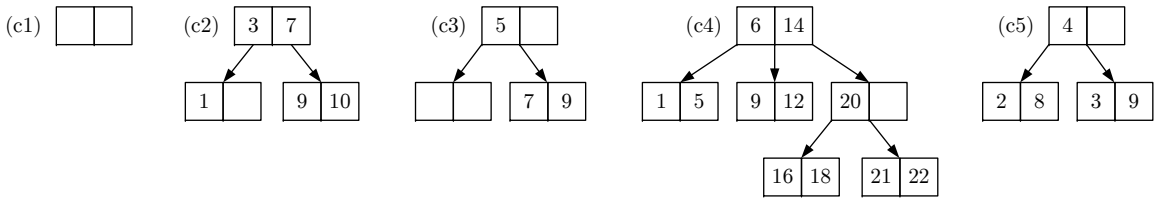
(a) Given the B-tree (k=2) below, insert key 16, then insert 26, and draw the resulting B-tree. (**5 points**)



(b) Given the B-tree (k=2) below, delete key 14, then delete 37, and draw the resulting B-tree. (**5 points**)



(c) Which of the following trees are valid—i.e., satisfy the constraints of—B-trees with k=1. Mark each tree as valid (✓), or invalid (×) and name the violations. (**5 points**)



Task 6 Distributed Graph Processing (5 points)

Explain how Apache Spark's abstraction of Resilient Distributed Datasets (RDDs) can be leveraged to compute the connected components of a graph in a distributed manner.