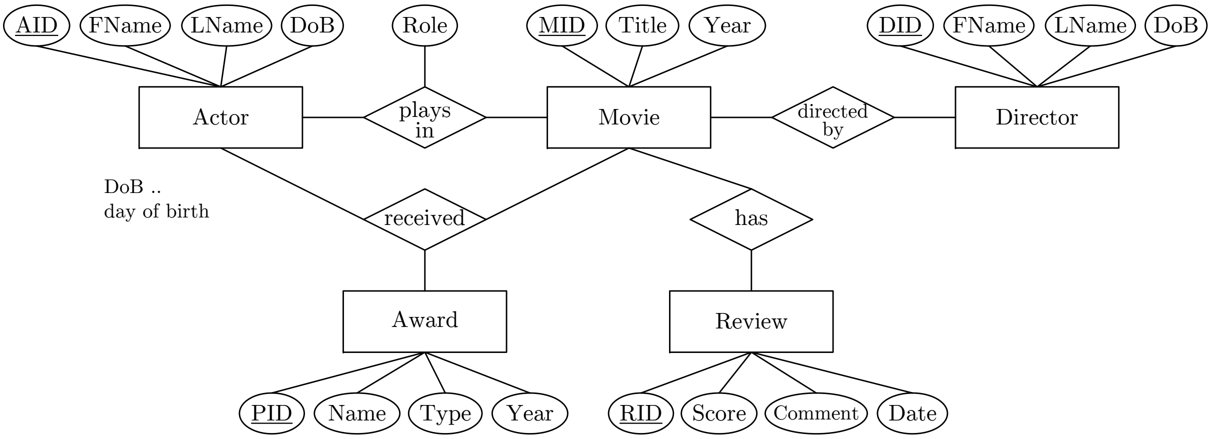
Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

July 29, 2020

Exam INF.01017UF Data Management (Summer 2020, V4a)

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 this exam sheet (first page), and on each additional piece of paper. You may give the answers in English or German, and you should directly write into the task description.

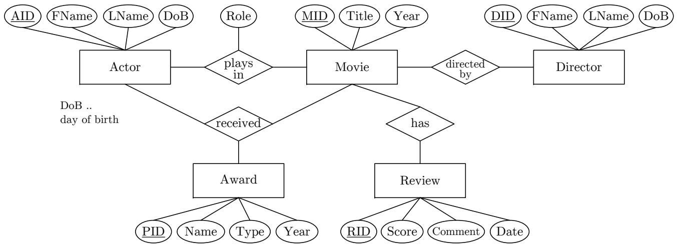
Task 1 Data Modeling (25 points)



(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) Given the relations in the table below, indicate the normal forms (None, or 1NF/2NF/3NF) they satisfy. Attribute $\underline{a}$ refers to a candidate key, b^M to a multi-valued attribute (e.g., a list of items), and $a \to b$ is a functional dependency from a to b (a implies b). (**4 points**)

Relation	**None**	**1NF**	**2NF**	**3NF**
$R(\underline{a}, b, c)$ with $a \to b$, $a \to c$				
$R(\underline{a}, b^M, c)$ with $b \to c$				
$R(\underline{a}, b, c)$ with $a \to b$, $b \to c$				
$R(\underline{a}, b, c)$ with $a \to b$, $a \to c$, $b \to c$				

Task 2 Structured Query Language (30 points)

Movies

MID	Title	Year	[min] Length	[Mio $] Budget	[Mio $] 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, × for non-equivalent). For non-equivalent expressions, briefly explain why. (**3 points**)

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

(b) Draw two logical query trees for query Q2 from Task 2(a): once in unoptimized form (with intersection), and once in optimized form (without intersection). (**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)

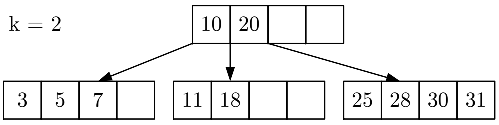
(a) Explain the ACID property Isolation (1 point), the concept of transaction isolation levels (2 points), and how these isolation levels relate to the anomalies dirty read, unrepeatable read, and phantom read (3 points). (**6 points**)

(b) Given an existing lock of type none, shared (S), or exclusive (X), indicate in the table below which additional locks can be successfully acquired (i.e., are compatible with the existing locks). Use ✓to indicate compatible locks, and × for incompatible locks. (**4 points**)

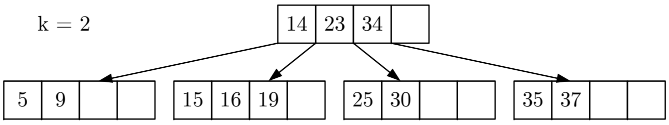
	Requested Lock			
Existing Lock	**S**	**X**	**IS**	**IX**
None				
S				
X				

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**)

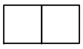


(b) Given the B-tree (k=2) below, delete key 14, then delete 37, and draw the resulting B-tree. (**5 points**)

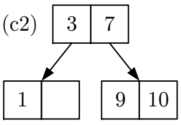


(c) Which of the following trees are valid—i.e., satisfy the constraints of—B-trees with k=1. Mark each tree as valid (✓), or invalid (×) and name the violations. (**5 points**)

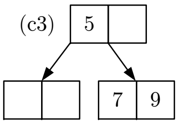
(c1)



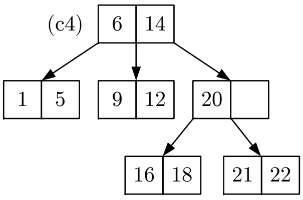
(c2)



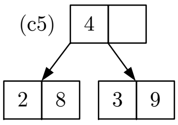
(c3)



(c4)



(c5)



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.