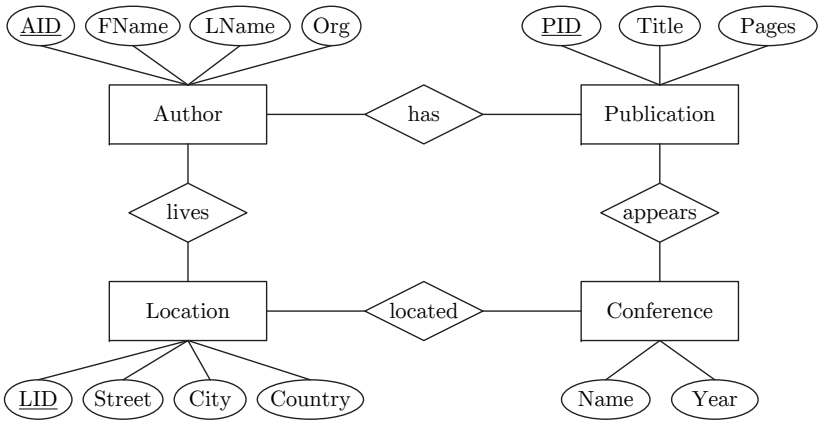
Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMVIT endowed chair for Data Management

January 30, 2020

Exam INF.01017UF Data Management (Winter 2019/20, V1a)

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 each piece of paper. You may give the answers in English or German, as well as directly write into the task description.

Task 1 Data Modeling (25 points)



(a) Given the above Entity-Relationship diagram, specify the cardinalities in Modified Chen notation based on the following information. (**8 points**)

- An author has at least one but potentially many publications, and a single publication has between one and eight authors.
- A publication might or might not appear in a conference proceedings, but due to double submission policies, the same publication cannot appear in multiple conferences. At every conference between 16 and 1024 papers are presented.
- Every conference takes place in exactly one location. Some locations are used more frequently to host conferences than others.
- Every author lives in exactly one location, but multiple authors can share a location. Due to moving authors and conference venues in selected locations, there might be locations without any authors or conferences though.

(b) Map the given Entity-Relationship diagram into a relational schema, including data types, primary keys, and foreign keys. You schema should also ensure that each author and each conference has an associated location. (**10 points**)

(c) Assume the author's organization to be multi-valued (i.e., multiple affiliations) as well as the functional dependency City $\rightarrow$ Country. Bring your schema in third normal form (schema changes are sufficient) and explain why it is in third normal form. (**7 points**)

Task 2 Structured Query Language (30 points)

Courses

CID	Name	Type	ECTS	Students	PID
1	Data Management	VO	3	500	5
2	Data Management	KU	1	450	5
3	Machine Learning	VU	3	700	4
4	Compiler Construction	VO	4	150	2
5	Distributed Systems	VO	4	210	2
6	Operating Systems	VU	7	800	1

Profs

PID	Name	Position
1	White	Assistant
2	Orange	Associate
4	Pink	Full
3	Brown	Associate
5	Blue	Full

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

```
Q1: SELECT DISTINCT C.Name
      FROM Courses C, Profs P
      WHERE C.PID = P.PID
        AND P.Position = 'Full'
```

```
Q2: SELECT Type, count(*) FROM Courses
      WHERE Students > 200
      GROUP BY Type
      ORDER BY count(*) DESC, Type ASC
```

```
Q3: SELECT Name, ECTS
      FROM Courses  WHERE Students > 400
    UNION DISTINCT
    SELECT Name, ECTS
      FROM Courses WHERE ECTS > 4
```

(b) Given the Courses and Profs tables above, write SQL queries to answer the following questions: (**15 points**)

- Q4: Which professors do not give any lectures (return the name and position of the professor)?

- Q5: Compute the professors that give more than one lecture as well as their number of lectures. (return name and count in descending order of counts and name)?

- Q6: Compute the minimum, maximum, and average number of students per course type (return course type, min, max, avg, in ascending order of course types)?

Task 3 Query Processing (14 points)

(a) Describe the conceptual idea of a nested-loop join, hash join, and sort-merge join. Furthermore, assume $R \bowtie S$ of $R(a, b, c)$ and $S(d, e, f)$ with cardinalities $N = |R|$ and $M = |S|$, and enter the space and time of these operators in the table below. (**10 points**)

Operator	**Time Complexity**	**Space Complexity**
(a1) Nested Loop Join (a2) Hash Join (a3) Sort-Merge Join (unsorted) (a4) Sort-Merge Join (sorted)		

(b) Rewrite the following four relational algebra expressions into equivalent expressions with lower costs, assuming the table schemas from task 3a. (**4 points**)

- (b1) $\sigma_{b=7}(R \bowtie S) \rightarrow$
- (b2) $(\sigma_{e>3}(S)) \cap (\sigma_{f<7}(S)) \rightarrow$
- (b3) $\pi_{a,b}(R \bowtie_{a=d} S) \rightarrow$
- (b4) $R \cup (\sigma_{d<e \,\wedge\, e<f \,\wedge\, f<d}(S)) \rightarrow$

Task 4 Transaction Processing (10 points)

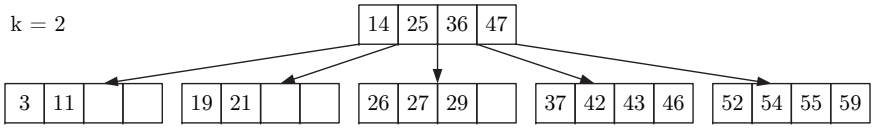
(a) Explain the concept and use of database locks. In this context, also briefly describe lock type compatibility and multi-granularity locking. (**4 points**)

Task 5 NoSQL (6 points)

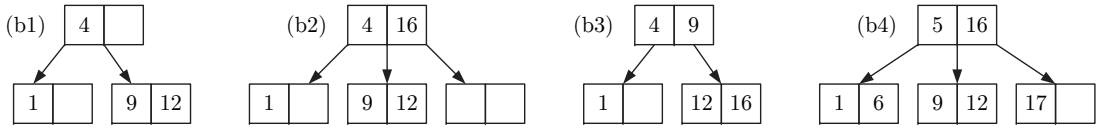
(a) Explain the architecture of a log-structured merge (LSM) tree as a commonly used data structure in Key-Value stores. Furthermore, describe in detail how `put(key, value)` and `get(key)` operations are realized on top of the LSM storage.

Task 6 Physical Design (20 points)

(a) Given the B-tree with k=2 below, insert the keys 7 and 49, and draw the resulting final B-tree. (**8 points**)



(b) 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. (**4 points**)



(c) Recall the table `Courses` from Task 2 and perform a horizontal and vertical partitioning, respectively. Specifically, provide—for both scenarios—relational algebra expressions for partitioning `Courses` into two fragments `Courses1` and `Courses2`, as well as its subsequent reconstruction from the two fragments. (**8 points**)

- Horizontal Partitioning (row partitions):
  - Courses1 :=
  - Courses2 :=
  - Courses :=

- Vertical Partitioning (column partitions):
  - Courses1 :=
  - Courses2 :=
  - Courses :=