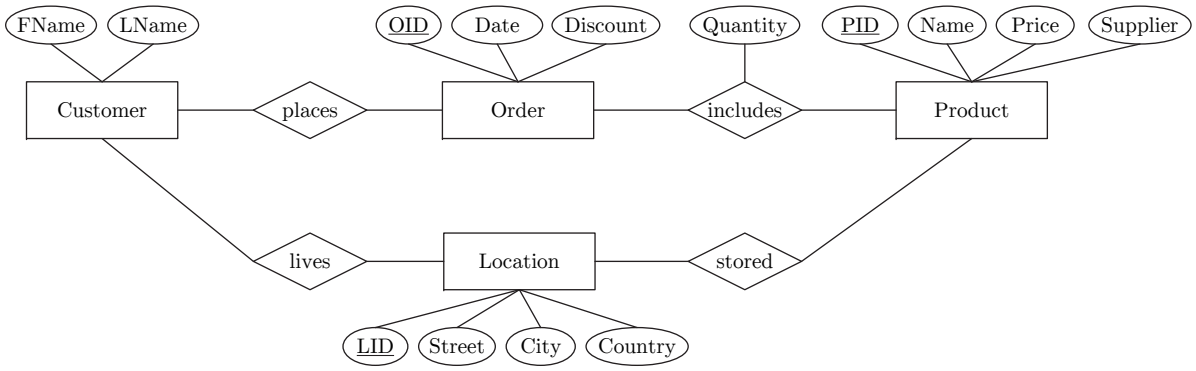
Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMVIT endowed chair for Data Management

February 06, 2020

Exam INF.01017UF Data Management (Winter 2019/20, V3a)

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 handout and each additional 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**)

- A customer places at least one but potentially many orders, and every order is placed by exactly one customer.
- An order may include between one and 128 line items (different products, each of a certain quantity).
- A product may be included in an arbitrary number of orders, and is stored in exactly one warehouse location. A warehouse might store many different products though.
- Every customer lives in exactly one location, but multiple customers might share a location.
- Due to moving customers and products stored in selected locations, there might be locations without any customers or products.

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

(c) Assume the product's supplier to be a multi-valued attribute (i.e., same product bought from potentially many suppliers) 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 are not associated with any courses (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 top-2 course types by average number of students (return course type, average number of students, in ascending order of number of students)?

Task 3 Query Processing (15 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 complexity 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. (**5 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) $R \cup (\sigma_{d<e \,\wedge\, e<f \,\wedge\, f<d}(S)) \rightarrow$
- (b5) $\sigma_{b=3}(\gamma_{b,max(c)}(R)) \rightarrow$

Task 4 Transaction Processing (5 points)

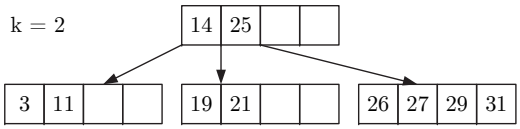
(a) Describe the concept of a database transaction log, and explain how it relates to the ACID properties Atomicity and Durability.

Task 5 Stream Processing (5 points)

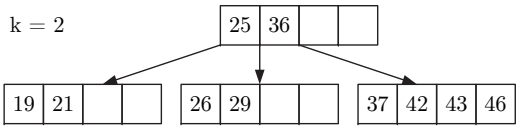
(a) Describe the basic system architecture and execution model of stream processing engines with a special focus on continuous queries and back pressure.

Task 6 Physical Design (20 points)

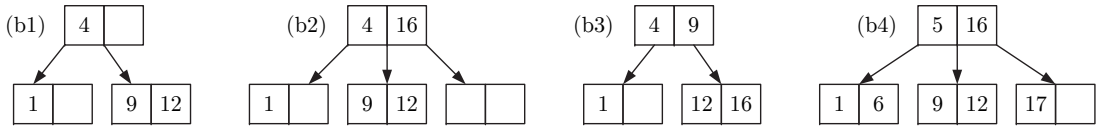
(a) Given the B-tree with k=2 below, insert the keys 7 and 37, and draw the resulting final B-tree. (**4 points**)



(b) Given the B-tree with k=2 below, delete the keys 21 and 43, and draw the resulting final B-tree. (**4 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. (**4 points**)



(d) 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 :=