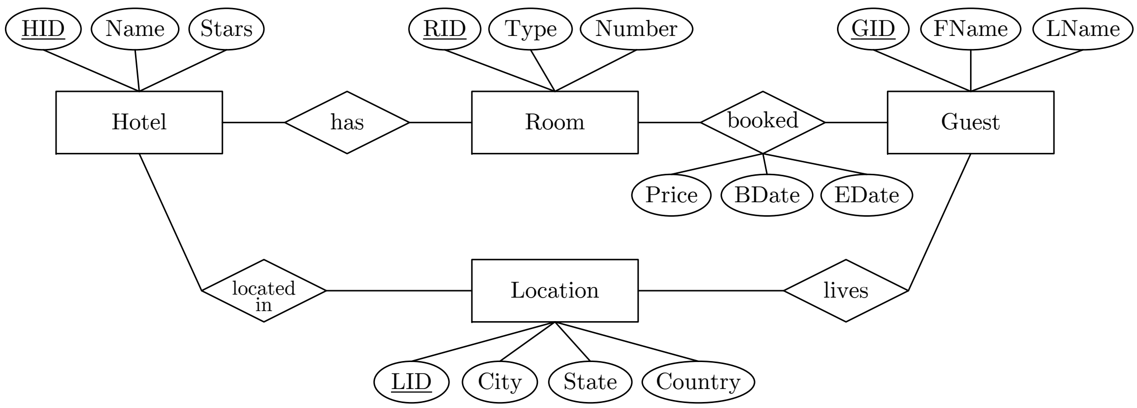
Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

February 12, 2021

Exam 706.010 Databases (Winter 2020/21, 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 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.

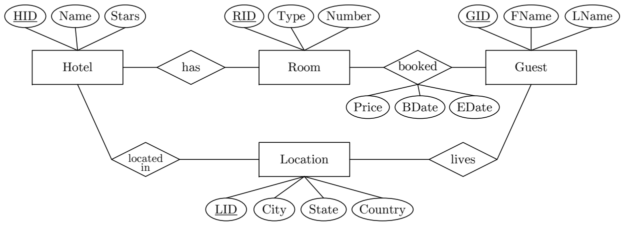
Task 1 Data Modeling (20 points)



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

- A hotel (described by a name, and a number of stars) has at least one but typically many rooms (described by a room type and room number), and every room belongs to exactly one hotel.
- Every guest (described by a first name and last name) may book one or many rooms, and multiple guests can share a single room for reduced prices or book the same room for different time periods. A booking is described by a price, a begin date (BDate) and an end date (EDate).
- Every hotel has exactly one location, and similarly guests live only in exactly one location. However, guests may not disclose this location information, and specific locations might not have any hotels or guests living there.

(b) Map the given ER diagram into a relational schema in third normal form (assuming a functional dependency State → Country, e.g., Styria implies Austria), including data types, primary keys, and foreign keys. The schema should also ensure that every room is associated with a hotel. Note that you only need to provide the final schema and there is no need to explain the normal forms. (**10 points**)



(c) Given the room booking period BDate-EDate, how could the schema be modified or augmented by constraints to enforce that a room can only be booked by a single guest at a time. You can assume a booking granularity of entire days. (**2 points**)

Task 2 Structured Query Language (30 points)

Courses

CID	Name	Type	ECTS	Students	PID
1	Data Management	VO	3	750	5
2	Data Management	KU	1	700	5
3	Machine Learning	VO	3	500	1
5	Distributed Systems	VO	4	210	3
6	Operating Systems	VU	7	600	4
4	Data Structures & Algorithms	VU	4	800	1
7	Compiler Construction	VU	7	150	2

Profs

PID	Name	Position
1	Red	Full
2	Orange	Associate
3	Yellow	Assistant
4	Green	Associate
5	Blue	Full
6	Violet	Assistant

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

```
Q1: SELECT DISTINCT C.Name, C.Type, P.Name
      FROM Courses C, Profs P
      WHERE C.PID = P.PID
        AND C.ECTS < 4
```

```
Q2: (SELECT Name, Students, ECTS
      FROM Courses WHERE Students > 400)
    INTERSECT
    (SELECT Name, Students, ECTS
      FROM Courses WHERE ECTS >= 4)
```

```
Q3: SELECT P.Position, sum(C.Students)
      FROM Courses C, Profs P
      WHERE C.PID = P.PID
      GROUP BY P.Position
```

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

- Q4: Which courses of type VO or VU are given by full professors (return course Name and Type)?

- Q5: Which professors give more than one course (return the prof Name, sorted in ascending order of Name)?

- Q6: Which professors do not give any courses (return the prof PID, Name)?

Task 3 Query Processing (20 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. (**5 points**)

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

(b) Draw the logical query trees for each of the queries Q1, Q2, and Q3 from Task 2a. (**9 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		

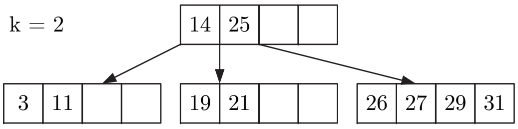
Task 4 Transaction Processing (10 points)

(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. (**5 points**)

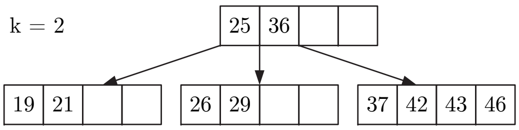
(b) Explain the concept and use of database locks. In this context, also briefly describe lock-type compatibility and multi-granularity locking. (**5 points**)

Task 5 Physical Design (20 points)

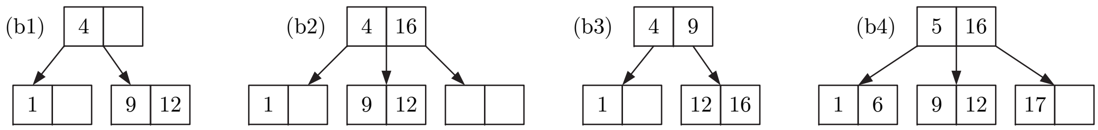
(a) Given the B-tree with k=2 below, insert the keys 7 and 37, and draw the resulting final B-tree. (**5 points**)



(b) Given the B-tree with k=2 below, delete the keys 25 and 43, and draw the resulting final 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. (**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. (**6 points**)

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

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