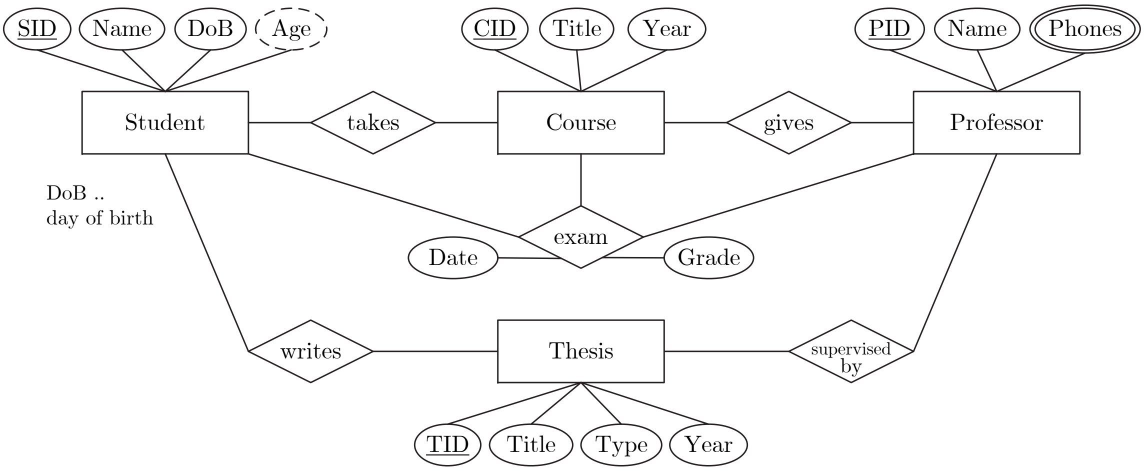
**Univ.-Prof. Dr.-Ing. Matthias Boehm**
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

July 28, 2022

# ☐ Exam INF.01017UF Data Management (Summer 2022, V3a)
# ☐ Exam 706.010 Databases (Summer 2022, V3b)

**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.

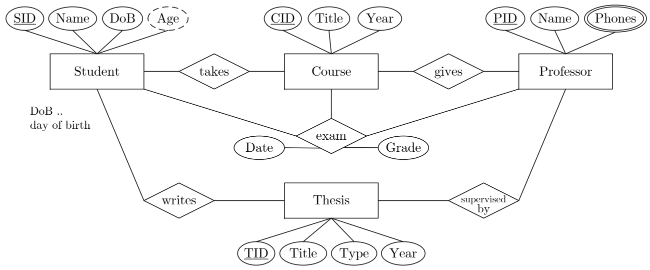
## Task 1 Data Modeling (25 points)




(a) Given the above Entity-Relationship diagram, specify the cardinalities in Modified Chen notation based on the following information. (**11 points**)

- A student can take up to 64 courses, and a single course can be simultaneously taken by up to 1000 students.
- A professor can give an arbitrary number of courses (including none), but every course it taught by exactly one professor. The phones attribute of a professor is a multi-valued attribute containing a list of phone numbers (e.g., {111-222-3333, office}; {444-555-6666, mobile}; {777-888-9999, private}).
- A student might have written multiple theses (e.g., types BS, MS, PhD), and every thesis is written by exactly one student. A thesis is also supervised by exactly one professor, and professors can supervise an arbitrary number of theses (including none).
- A professor can take exams for many (an arbitrary number of) students of a course, and for many courses of a single student, but the exam of a specific student-course combination is taken by exactly one professor.

(b) Map the given Entity-Relationship diagram into a relational schema in *third normal form*, including data types, primary keys, and foreign keys. Your schema should also ensure that each course has an associated professor, and each thesis has a student (author) and professor (supervisor). (**14 points**)



## Task 2 Structured Query Language (30 points)

**Employees**

| <u>EID</u> | FName | LName | Age | Country | PID |
|---|---|---|---|---|---|
| 4 | Isabella | Brown | 30 | AT | 2 |
| 2 | Olivia | Johnson | 30 | FR | 1 |
| 1 | Emma | Smith | 35 | DE | 3 |
| 3 | Ava | Williams | 20 | DK | 1 |
| 5 | Sophie | Jones | 35 | AT | 2 |
| 6 | Taylor | Miller | 55 | DE | 5 |
| 7 | Charlotte | Davis | 40 | DE | 2 |

**Projects**

| <u>PID</u> | Name | Customer |
|---|---|---|
| 1 | UX Design | B |
| 2 | App Backend | B |
| 3 | Data Storage | C |
| 4 | ML Pipeline | A |
| 5 | UX Design | A |
| 6 | HW Accelerator | D |

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

```
Q1: SELECT DISTINCT P.Customer, P.Name
      FROM Employees E, Projects P
      WHERE E.PID = P.PID
        AND E.LName IN('Williams','Jones','Miller')
```

```
Q2: SELECT FName, LName
      FROM Employees WHERE Country = 'DE'
    UNION DISTINCT
    SELECT FName, LName
      FROM Employees WHERE Age >= 35
```

```
Q3: SELECT P.Name, round(avg(E.Age)) --avg=sum/count
      FROM Employees E, Projects P
      WHERE E.PID = P.PID
      GROUP BY P.Name
```

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

- Q4: Which employees work on projects for customer B (return the FName and LName, sorted in ascending order of LName)?

- Q5: Which customers have more than one project (return the Customer, and number of projects per Customer)?

- Q6: Which projects are not worked on by any employee (return the project PID, Name, and Customer)?

## Task 3 Query Processing (16 points)

(a) Assume relations $R(a, b, c)$, $S(d, e)$, $T(f, g)$ 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, explain why. (**4 points**)

| Expression 1 | Expression 2 | Equivalent? Why Not? |
|---|---|---|
| $\sigma_{b=3 \wedge d<b}(R \bowtie_{a=e} S)$ | $(\sigma_{b=3}(R)) \bowtie_{a=e} (\sigma_{d<3}(S))$ | |
| $(R \bowtie_{a=e} S) \bowtie_{d=f} T$ | $R \bowtie_{a=e} (S \bowtie_{d=f} T)$ | |
| $\pi_{b,d}(R \bowtie_{a=e} S)$ | $(\pi_{a,b}(R)) \bowtie_{a=e} (\pi_{d,e}(S))$ | |
| $\sigma_{b>7}(R) \cap \sigma_{a<b \wedge b<a}(R)$ | $\sigma_{b>7}(R)$ | |

(b) Draw two logical query trees for query Q2 from Task 2(a): once in unoptimized form (with union), and once in optimized form (without union). (**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 (9 points)

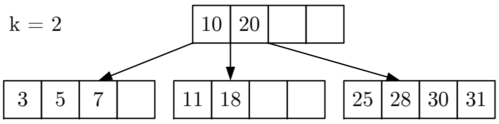
(a) Explain the concept of a database transaction log, and how it helps to ensure Atomicity and Durability of changes made by uncommitted and committed transactions on failures. (**6 points**)

(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 $T_2$. (**3 points**)

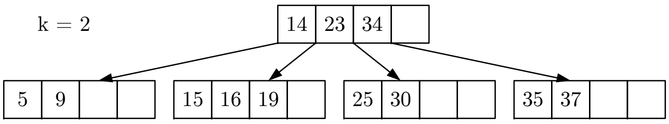
| **Schedule 1** | **Schedule 2** | **Equivalent?** |
|---|---|---|
| $\{r_1(a), w_1(a), r_2(b), w_2(b)\}$ | $\{r_1(a), r_2(b), w_1(a), w_2(b)\}$ | |
| $\{r_1(c), w_1(c), r_2(c), r_2(d), w_2(d)\}$ | $\{r_1(c), r_2(c), r_2(d), w_1(c), w_2(d)\}$ | |
| $\{r_1(e), w_1(e), w_2(e), w_2(f)\}$ | $\{w_2(e), w_2(f),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

## Task 5 Physical Design (20 points)

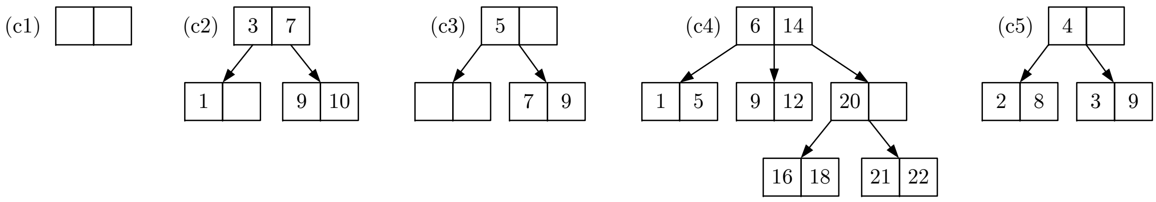
(a) Given the B-tree (k=2) below, insert key 16, then insert 26, and draw the resulting B-tree. (**5 points**)




(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**)




(d) Given a relation $R(a, b, c)$ and a query workload Q09: $\sigma_{a<3}(R)$, Q10: $\sigma_{a<7}(R)$, Q11: $\sigma_{a\geq 7}(R)$, and Q12: $\sigma_{a<3 \wedge b=2}(R)$, find a *disjoint* and *complete* horizontal partitioning into three partitions $R_1$, $R_2$, and $R_3$ that improves the cost (e.g., # scanned tuples) of all four queries. Provide the relational algebra expressions for partitioning and querying. (**5 points**)

$R_1$ :

$R_2$ :

$R_3$ :

Q09:

Q10:

Q11:

Q12: