UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Exam

DIT033 / DAT335 – Data Management

Wednesday, August 16th, 2023 14:00 - 18:00

Examiner:
Philipp Leitner

Contact Persons During Exam:
Philipp Leitner (+46 733 05 69 14, philipp.leitner@chalmers.se)

Allowed Aides:
None except English dictionary (non-electronic), pen/pencil, ruler, and eraser.

Results:
Exam results will be made available no later than in 15 working days through Ladok.

Grade Limits:
0 – 49 pts: U, 50 – 69 pts: 3, 70 – 84 pts: 4, 85+ pts: 5

Review:
Exam can be reviewed at student office after the exam. Please contact the examiner if you want to discuss your results.

Task 1 – Theory and Understanding (24 pts)

Each of the following questions requires an approximately two to four paragraphs long answer **in your own words**. Many of these questions ask for examples – these examples are important, and you should develop your own examples (simply re-using an existing example from the lecture slides or the Internet is not sufficient). Use figures or sketches if appropriate. Read the questions carefully, and make sure to answer the question that is asked.

Q1.1: What is a “weak entity” in the Entity Relationship model? What is special about them (e.g., in terms of key requirements)? When would you typically use a weak entity? Provide a simple example of a weak entity, and describe why it makes sense to use a weak entity here.

Q1.2: What is “well-formedness” in the context of XML or JSON? How does it relate to validity? Provide 2 important syntactical rules for both, JSON and XML, which need to be fulfilled to make a document well-formed. *(4 pts)*

Q1.3: What isolation levels are available in SQL? Name and describe them briefly. Further look ahead to the example database introduced in Task 2, and argue which isolation level you would choose. Motivate your choice. *(5 pts)*

Q1.2: What problem do views in SQL solve? Which different types of views are there?

Q1.5: Explain the basic idea behind Map/Reduce using an example. How can you use Map/Reduce in a MongoDB database, and for what use cases would you typically use it? *(6 pts)*

Task 2 – EER Diagrams (24 pts)

Consider the following excerpt of the domain description for the database of a library. Model the described domain using an EER diagram with the notation we used in the course (find a cheat sheet in the appendix of your exam papers). Use the 1,N,M notation for describing cardinalities rather than the min-max notation. If *and only if* something is not specified, make a reasonable assumption and note it down in plain text.

Library Database:
In our database we need to store books and their editions. Every book has an unique isbn number and a title (which is also unique). For every edition of books we need to save the publication year and the number of pages. Editions do not have a key attribute by themselves, instead they are identified through a combination of an edition number (e.g., second edition) and the ISBN of their book. Each book has at least one edition, but can have many. Libraries are identified through an unique name, and store one to many editions of books. However, some editions are not available in any library.

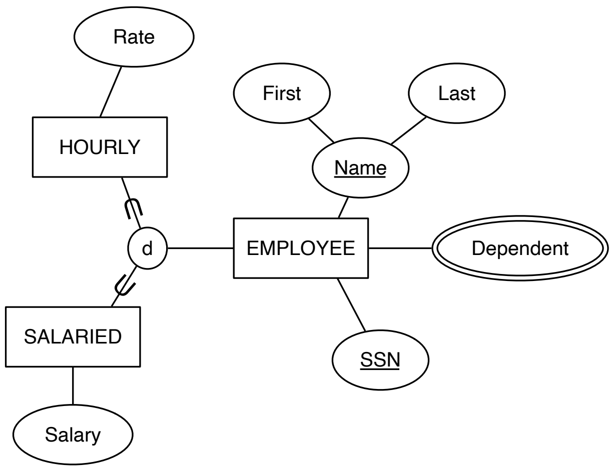
A special type of book is a translation. Translations additionally have a language attribute, which specifies the language the book has been translated to. A translation is always a translation of exactly one edition of a book, but each edition can have multiple translations (or none).

We also need to store the data of a variety of persons associated with our libraries. Each person has an unique social security number, a name, and one or multiple addresses. Each address comprises of a street name, a door number, a city name, and a country name. There are three basic types of persons in the database: library staff, lenders, and authors. A person can potentially have multiple of these roles (but will have at least one of them). Each author authors one to many books (and each book has at least one, but potentially many, authors). Lenders rent editions of books from libraries. A lender can rent multiple book editions, and each edition can be rented by multiple persons. However, both of these associations are optional. If a lender rents a book edition, we also need to store the dates when it has been rented and when it has been returned. Staff works at libraries. Each person who is a staff works in exactly one library, but of course a library can have more than one staff (but will always have at least one). Each staff member has a salary, and we also need to store the date when the staff member started working. Finally, each staff member may declare zero or one books to be their favorite book (of course, each book can be the favorite of multiple staff members), irrespective of whether that book is stocked in the library where the staff member works.

Task 3 – Mapping an EER model (12 pts)

Consider the EER model below. Construct **two alternative** mappings to the relational model using the different ways to map inheritance we covered in the course. Select primary keys and introduce foreign keys as necessary. Use the notation that we used in the course to indicate relations, attributes, primary keys, and foreign keys (see Task 4 for an example of the required notation).

For both alternatives you provide, discuss advantages and disadvantages in your own words.



Task 4 – Relational Algebra (20 pts)

Relational Model:
DURATION(<u>id</u>, begin, end)
CONFERENCE(<u>name</u>, location, duration)
    duration $\rightarrow$ DURATION.id
DELEGATE(<u>reg_nr</u>, name, date_of_birth)
REGISTRATION(<u>delegate</u>, <u>conference</u>, registration_date, fee)
    delegate $\rightarrow$ DELEGATE.reg_nr
    conference $\rightarrow$ CONFERENCE.name

Given this relational model, write relational algebra statements that exactly represent the following queries. Use the mathematical notation from the course (for the correct notation you can again refer to the appendix).

Queries:
Q4.1: Find all delegate names registered to a conference at or after January 1st, 2023.

Q4.2: List all conference names along with their duration (calculated as end date minus begin date).

Q4.3: Find all pairs of conference names of conferences in the same location. It is ok if the same pairs appear multiple times in the list.

Q4.4: Return a list of conference names and the total value of their registrations (calculated as the sum of all conference fees paid for registrations for this conference).

Task 5 – SQL (20 pts)

Assume that the relational in Task 4 has been implemented in a relational database. Write the following SQL queries against this database. *(5 pts per query)*

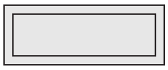
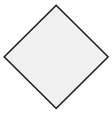
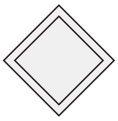
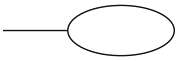
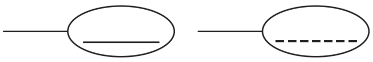
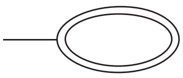
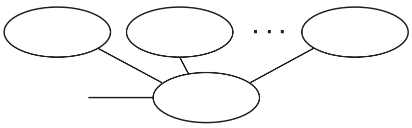
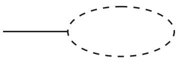
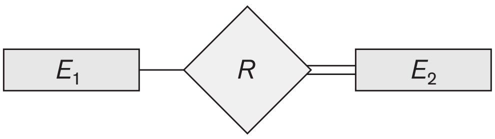
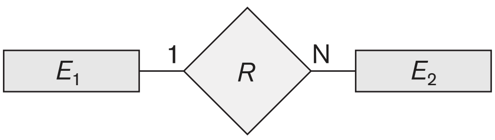
Q5.1: Delete all registrations to the conference with the name “ICSE2023” from before January 1st, 2023.

Q5.2: Return a list of all delegate names, their dates of birth,the name of the conferences that they are registered for, and the location of the conference (one return row per registration). Delegates not registered to any conference should not be contained in the result.

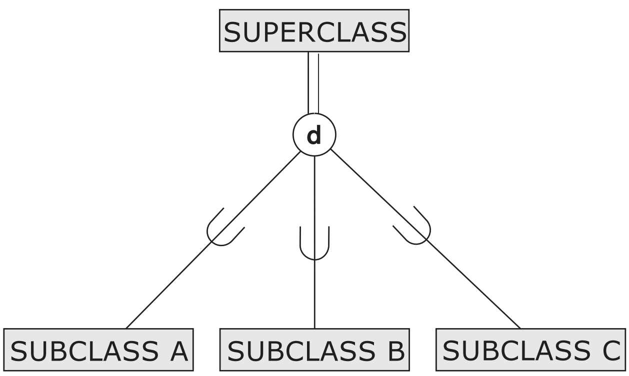
Q5.3: Return a list of conference names and the total value of their registrations (calculated as the sum of all conference fees paid for registrations for this conference). Only include conferences where the total fee is larger than 100.000.

Q5.4: Return the name of the oldest delegate (i.e., the delegate with the smallest date of birth) attending the conference with the name “ICSE2023”.

Appendix: Notation Guidelines for EER and RA

Symbol	**Meaning**
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute / Dashed Underline for Partial Key
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for E_1 : E_2 in R

Total Disjoint Specialization



Partial Overlapping Specialization

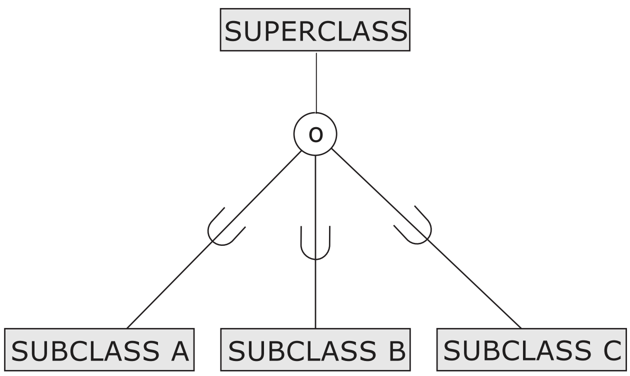


Table 8.1 Operations of Relational Algebra

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}),\ (\text{<join attributes 2>})} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}),\ (\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

```
<grouping>𝓕<functions>(R)
```

whereas `<functions>` is a list of

```
[MIN|MAX|AVERAGE|SUM|COUNT] <attribute>
```