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 08, 2021

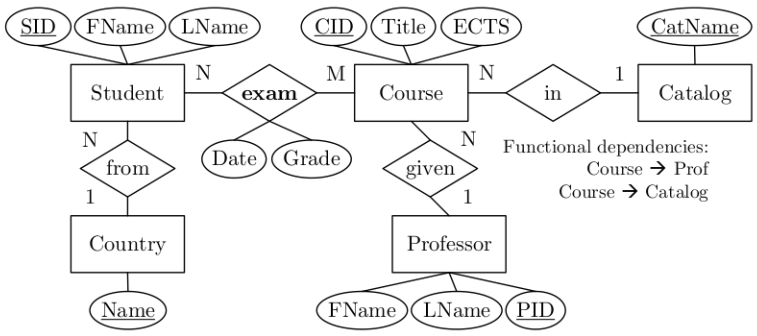
Exam 706.520 Data Integration and Large-Scale Analysis (WS20/21)

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 Warehousing (15 points)

(a) Describe the overall architecture (macroscopic) of an enterprise *data warehouse*, name its components, and briefly describe the purpose of these components. (**6 points**)

(b) Given below entity relationship (ER) diagram, create a corresponding ROLAP *star schema*. Data types can be ignored, but indicate primary and foreign key constraints. (**9 points**)



Task 2 Entity Resolution (20 points)

(a) Explain the phases of a typical *entity resolution pipeline* (deduplication pipeline), and discuss example techniques for the individual phases. (**16 points**)

(b) Assume the following two schemas from different sources. Which *pre-processing* steps are necessary for entity resolution, and how can *schema matching and mapping* help the automation of this process? (**4 points**)

- Publications(AuthorNames, PaperTitle, Venue, Pages)
- Papers(PID, Title, Abstract, Conference, NumPages)
  PaperAuthor(PID, AID, Position)
  Authors(AID, Name, Institution)

Task 3 Data Cleaning (20 points)

A	B	C	D	E
Red	2100	X	DE	35
Orange	4300	**NULL**	DE	**NULL**
Yellow	5700	Z	DE	35
Green	2500	X	AT	25
Blue	4900	Y	US	**NULL**
Violet	5200	**NULL**	US	45

(a) Given the data table above, describe two techniques for *missing value imputation* in the *categorical* column C. If possible provide the imputed values. (**6 points**)

(b) Given the data table above, describe two techniques for *missing value imputation* in the *numerical* column E. If possible provide the imputed values. (**6 points**)

(c) Explain the approach of data cleaning via *robust functional dependencies* (FDs) and *robust inclusion dependencies* (INDs). (**8 points**)

- Robust FDs:

- Robust INDs:

Task 4 Data Provenance (8 points)

(a) Explain the general goal and concept of *data provenance*, and distinguish *why-provenance and how-provenance*. (**5 points**)

- Data Provenance:
- Why-Provenance:
- How-Provenance:

(b) Given below tables R and S (with tuples r_i and s_i, respectively), query Q and the results O, specify the *provenance polynomials* for every tuple in O. (**3 points**)

R

	A	B
r_1	X	1
r_2	Y	2
r_3	Z	1

S

	C	D
s_1	1	A
s_2	2	B
s_3	2	A
s_4	2	C

```
SELECT DISTINCT S.D
  FROM R, S
  WHERE R.B=S.C
```

O	Provenance Polynomials?
A	
B	
C	

Task 5 Resource Management (7 points)

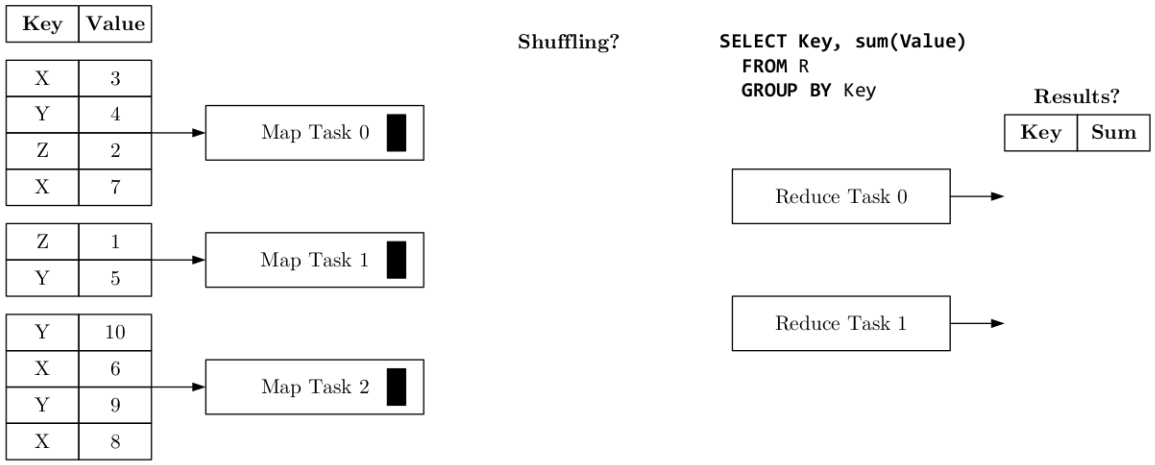
(a) Briefly explain the concept of *resource allocation* for multiple resources (e.g., dominant resource calculation for CPU and memory in YARN). (**3 points**)

(b) Assume three nodes with CPU and memory capacity N_1 (32 cores, 64 GB), N_2 (16 cores, 64 GB), N_3 (64 cores, 128 GB) and a stream of resource requests $R1 \ldots R_7$. Schedule these requests to available resources (assign requests to nodes) in order to maximize the number of fulfilled requests. (**4 points**)

- R_1: (30 cores, 8 GB)
- R_2: (6 cores, 32 GB)
- R_3: (8 cores, 64 GB)
- R_4: (10 cores, 32 GB)
- R_5: (8 cores, 32 GB)
- R_6: (16 cores, 64 GB)
- R_7: (16 cores, 16 GB)

Task 6 Distributed Storage and Computation (15 points)

(a) Data-parallel frameworks like MapReduce or Spark rely on data shuffling for distributed joins, aggregation, and partitioning. Explain—using below example of a group-by aggregation query—how *shuffling* operates and how data transfer can be reduced. (**10 points**)



(b) Describe two techniques for fine-grained *fault tolerance* in distributed storage systems and/or distributed computing frameworks. (**5 points**)

Task 7 Stream Processing (15 points)

(a) Assume an input stream S with schema $S(A, T)$—where T refers to event time (the smaller the older, start at zero)—and a continuous query $Q : \gamma_{A,count()}(S)$ (group-by A, return count) with *stream window aggregation*. Compute the output streams with schema (A, count, T) for the following three scenarios. (**9 points**)

- Input Stream:
  (x,0.5s), (y,1.1s), (x,2.1s), (y,2.9s), (x,4.1s), (x,4.4s), (x,4.5s), (x,5.2s), (x,5.9s), (y,7.1s), (y,8.8s), (x,10.1s), (x,10.7s), (y,11.8s), (x,11.9s).

- Tumbling Window (size 3s):

- Sliding Window (size 4s, step 3s):

- Sliding Window (size 5s, step 4s):

(b) Explain the following three techniques for *handling overload* situations in stream processing engines. (**6 points**)

- Back Pressure:

- Load Shedding:

- Distributed Stream Processing: