Univ.-Prof. Dr.-Ing. Matthias Boehm
Technische Universität Berlin
Faculty IV - Electrical Engineering and Computer Science
Berlin Institute for the Foundations of Learning and Data (BIFOLD)
Big Data Engineering (DAMS Lab) Group

February 15, 2024

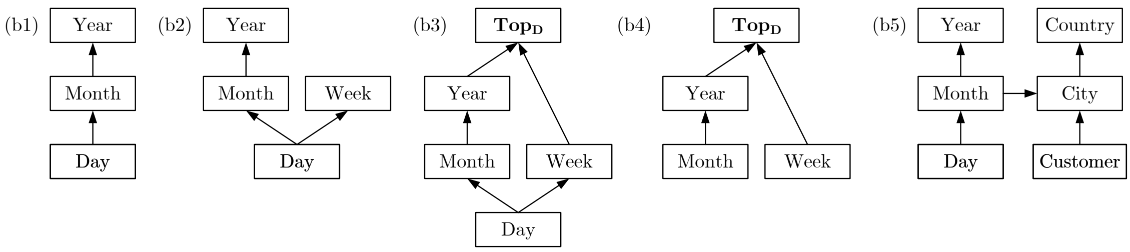
Exam Data Integration and Large-Scale Analysis (WiSe23/24)

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 your own paper. You may give the answers in English or German, written directly into the task description.

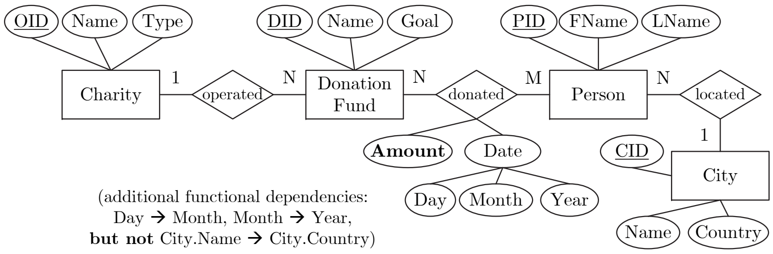
Task 1 Data Warehousing (25 points)

(a) Describe the overall system architecture of a *data warehouse* (not a data center), name its components, and briefly describe the purpose of these components. (**6 points**)

(b) The central metaphor of multi-dimensional modeling is the data cube, described by dimensions and measures. Which of the following *Date* dimension hierarchies are well-formed. Mark each hierarchy as valid (✓) or invalid (×) and name the violations. (**5 points**)



(c) Given the entity relationship (ER) diagram below, create corresponding relational *star and snowflake schemas*. Data types can be ignored, but indicate primary and foreign key constraints. (**7+7 points**)



Star Schema:

Snowflake Schema:

Task 2 Message-oriented Middleware (5 points)

Assume a message-oriented middleware with a single *FIFO* message queue. Indicate, in the table below, true (✓) properties of the following three message delivery guarantees.

	At Most Once	**At Least Once**	**Exactly Once**
Requires Message Persistence			
Requires Transaction Mechanism			
Prevents Message Outrun			
Prevents Message Loss			
Prevents Message Double Delivery			

Task 3 Schema Matching and Mapping (6 points)

Characterize the concepts of schema matching and schema mapping by indicating in the table below true (✓) characteristics.

	Schema Matching	**Schema Mapping**
Produces Schema Correspondences		
Consumes Schema Correspondences		
Applies Similarity Functions		
Analyzes Available Data		
Utilizes Schema Constraints		
Produces Transformations Programs		

Task 4 Entity Resolution (16 points)

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

Task 5 Data Cleaning (8 points)

In the context of missing value imputation, describe the following types of missing data, name related techniques for *missing value imputation*, and provide imputed values for the missing values on the right (once with an MCAR technique, and once with MAR).

Name	Age	Salary
Red	45	4500
Orange	50	NULL
Yellow	20	2000
Green	40	NULL
Blue	25	2500
Violet	35	NULL

- Missing Completely at Random (MCAR):

- Missing at Random (MAR):

- Not Missing at Random (NMAR):

Task 6 Data Provenance (8 points)

(a) Explain the general goal and concept of *data provenance* in a broad sense. (**3 points**)

(b) Given the tables R and S below (with tuples r_i and s_i, respectively), provide the *provenance polynomials* for every result tuple in the table on the right. (**5 points**)

```
SELECT R.B, count(*)
  FROM R, S
  WHERE R.A = S.D
  GROUP BY R.B
```

R

	A	B
r_1	1	X
r_2	2	Y
r_3	3	X
r_4	4	Z

S

	C	D
s_1	E	1
s_2	F	2
s_3	G	3
s_4	H	2
s_5	I	4

⇒

Output / **Provenance Polynomials**

B	count	Provenance Polynomials
X	2	
Y	2	
Z	1	

Task 7 Cloud Computing (4 points)

Explain the concept of Function as a Service (FaaS) and discuss advantages and disadvantages.

Task 8 Distributed, Data-Parallel Computation (14 points)

(a) Given the distributed dataset D of three partitions below, describe the data-parallel (MapReduce-like) computation of $Q : \gamma_{A,sum(B)}(D)$ (group-by A, return sum(B) per group) including how shuffling works and the *actual example results.* (**7 points**)

A	B
X	3
X	4
X	1
Y	7

X	2
Y	3
X	1
X	2

Y	5
X	3
Z	7
X	4

(b) Briefly name three conceptual techniques for improving the runtime performance of the data-parallel computation above. (**3 points**)

-
-
-

(c) Name and describe techniques for ensuring fault-tolerance in distributed computation as well as distributed storage. (**2+2 points**)

Task 9 Stream Processing (14 points)

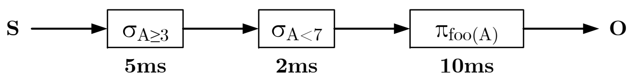
(a) Assume an input stream S with schema $S(A, T)$—where T is the event time (the smaller the older, start at zero)—and a query $Q : \gamma_{A,count()}(S)$ (group-by A, return count) with *stream window aggregation*. Compute the following output streams with schema (A, count, T_c), where T_c is the creation time (first output at full window size). (**6 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 5s, step 4s):

(b) Given the input stream S and continuous query below, compute the latency of individual tuples (in milliseconds), and maximum tuple throughput (in tuples/second). (**4 points**)



- Tuple Latency [ms]:
- Tuple Throughput [tuples/s]:

(c) Draw an optimized continuous query that produces semantically equivalent output streams O_1 and O_2, but avoids unnecessary redundancy. (**4 points**)