Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Universität Hamburg
Fachbereich Informatik
Datenbanken und Informationssysteme
Prof. Dr. Norbert Ritter, Dr. Fabian Panse

Module

Databases and Information Systems

Summer Term 2018

Mock Exam
12.07.2018, 8:00 to 10:00, ESA B

Sample Solution

Part	1	2	3	4	5	6	7	8	Total	Grade
Points	18	18	12	12	15	18	17	10	120	
Result										

Rechtsmittelbelehrung (Instructions on Right of Appeal):

Gegen die Bewertung dieser Prüfungsleistung kann innerhalb eines Monats nach ihrer Bekanntgabe Widerspruch erhoben werden. In diesem Zeitraum kann die Bewertung der Klausur eingesehen werden. Der Widerspruch ist schriftlich oder zur Niederschrift beim Vorsitzenden des B.Sc./M.Sc.-Prüfungsausschusses einzulegen. Es wird darauf hingewiesen, dass ein erfolgloses Widerspruchsverfahren kostenpflichtig ist.

For your convenience, we provide an **English translation** below.
Please note that the English version is not legally binding:

You may appeal the examination result within one month after its publication. During this time period, the evaluated examination will be made accessible to you. The appeal must be made in writing or declared for recording at the B.Sc./M.Sc. examinations board ("B.Sc./M.Sc.-Prüfungsausschuss").
Please note that an unsuccessful appeal may be subject to a charge.

Part 1: Concurrency Control: Correctness

[18 P.]

a) Consider the following transaction schedules. For each schedule, do the following:

- draw the conflict (precedence) graph,
- indicate whether or not the schedule is **conflict-serializable (CSR)**, and if so give a conflict-equivalent serial schedule (you just need to list the order of transactions), and
- indicate whether or not the schedule is **view-serializable (VSR)**.

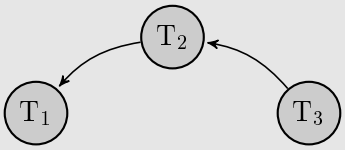
Briefly justify your answers.

Consider the following notation for operations of transactions:

$w_i(x)$: transaction i writes object x
$r_i(x)$: transaction i reads object x
c_i : transaction i commits
a_i : transaction i aborts

[4 P.] i) $\mathbf{S_1 = r_1(x)\ r_2(x)\ w_1(x)\ r_2(y)\ w_2(y)\ r_3(v)\ w_3(v)\ r_2(v)\ w_2(v)\ c_1\ c_2\ c_3}$

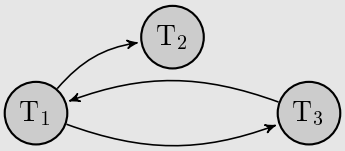
Lösungsvorschlag:



Is in CSR (and therefore in VSR) because of the acyclic conflict graph. The only possible order is T_3, T_2, T_1.

[4 P.] ii) $\mathbf{S_2 = r_1(x)\ w_1(x)\ r_2(x)\ c_2\ r_3(x)\ r_3(y)\ r_1(y)\ w_1(y)\ c_1\ c_3}$

Lösungsvorschlag:

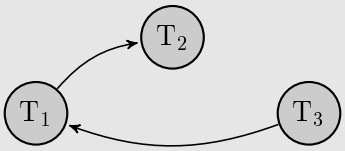


Is not in CSR, cause of the cycle $T_1 \longleftrightarrow T_3$ in the graph.

Since the schedule ist not in CSR and it contains no blind-writes, it is also not in VSR.

iii) $\mathbf{S_3 = r_1(x)\ w_1(x)\ r_2(x)\ c_2\ r_3(y)\ w_3(y)\ r_1(y)\ w_1(y)\ c_3\ c_1}$ [4 P.]

Lösungsvorschlag:



Is in CSR (and therefore in VSR) because of the acyclic conflict graph. The only possible order is T_3, T_1, T_2.

b) Give short answers to the following questions.

i) Is schedule S_3 order-preserving conflict serializable (OCSR)? Briefly justify your answer. [2 P.]

Lösungsvorschlag:

S_3 is not in OCSR, because the equivalent serial history is T_3, T_1, T_2, but T_2 is already commited before T_3 starts.

ii) Is schedule S_1 commit order-preserving conflict serializable (COCSR)? Briefly justify your answer. [2 P.]

Lösungsvorschlag:

S_1 is not in COCSR, because the commit order is T_1, T_2, T_3, but the conflict ($r_2(x)$,$w_1(x)$) has the order T_2 before T_1 and the conflicts ($w_3(v)$,$r_2(v)$), ($r_3(v)$,$w_2(v)$) and ($w_3(v)$,$w_2(v)$) have the order T_3 before T_2.

iii) Given the schedule $\mathbf{S_4 = r_1(x)\ w_1(x)\ r_2(x)\ w_2(y)\ a_1\ c_2}$, indicate whether or not the schedule can produce canonical synchronization problems (i. e. Dirty-Read, Lost-Update, Non-repeatable Read, Phantom-Problem). Which of them? [2 P.]

Lösungsvorschlag:

The operation $r_2(x)$ reads the value written by $w_1(x)$, but T_1 ends with an abort $\Rightarrow$ Dirty Read.

Addendum: A lost update cannot occur, because no write operation is enclosed by other write operations. Non-repeatable read and phantom problem cannot occur, because every transaction only reads once.

Part 2: Concurrency Control: Synchronization

[18 P.]

a) Consider a database with the following schema:

Book(ISBN, Title, Language, Pages)
Seller(SID, Name, Address)
Inventory(Book → Book.ISBN, Seller → Seller.SID, Price, Quantity)

Assume further that table Inventory contains the following data:

Inventory			
Book	Seller	Price	Quantity
53	2	9.33	1
87	1	45.35	1
48	4	45.99	1
42	4	200.00	2
12	9	39.05	1
92	9	28.22	4

Consider the transactions T_1 and T_2. T_1 always has the highest ANSI isolation level **Serializable (level 3)** (DB2's Repeatable Read).

[4 P.] i) Suppose T_2 uses **Read Committed (level 1)** (DB2's Cursor Stability) on the following SQL statements. What is the price of the book 42? Briefly explain whether or not a canonical synchronization problem (i. e. Dirty-Read, Lost-Update, Non-repeatable Read, Phantom-Problem) occurs and if so, name the problem and describe which ANSI isolation level must be set at least, to prevent it.

T_1	T_2
	`SELECT Price FROM Inventory WHERE Book = 42;`
`UPDATE Inventory SET Price = Price * 0.50 WHERE Book = 42;`	
`COMMIT`	
	`UPDATE Inventory SET Price = Price * 0.80 WHERE Book = 42;`
	`COMMIT`

Lösungsvorschlag:

The price will be **$80.00**, because of the **Non-repeatable Read** problem: T_2 holds the read lock only as long as it reads the value. Thus, T_1 can halve the price to 100 and T_2 calculates 80 percent of the value read.

The isolation level must be set at least to **Repeatable Read** (DB2's Read Stability) to prevent the **Non-repeatable Read**.

ii) Given the following SQL statements of T_1 and T_2. T_2 returns 2 for the count of inventory entries and 7 for the sum of the quantity. Briefly explain whether a canonical synchronization problem occurred and if so, name the problem and say which is the highest possible ANSI isolation level used by T_2 and which ANSI isolation level have to be set at least to prevent it. [4 P.]

T_1	T_2

	`SELECT COUNT(*)`
	`FROM Inventory`
	`WHERE Seller = 9;`

`INSERT INTO Inventory` `VALUES(54, 9, 123.33, 2);`	
`COMMIT`	

	`SELECT SUM(Quantity)`
	`FROM Inventory`
	`WHERE Seller = 9;`

	`COMMIT`

Lösungsvorschlag:

The highest possible isolation level for T_2 is Repeatable Read, because T_2 only counted two entries but the quantity sum includes the **Phantom** inserted by T_1. The highest ANSI isolation level that permits the **Phantom Problem** is **Repeatable Read** (DB2's Read Stability) and hence we need to set the isolation level of T_2 to **Serializable** to prevent it.

iii) Given the following SQL statements of T_1 and T_2. If T_2 encounters a Dirty Read problem, what will be the result set of the select statement. Briefly explain which ANSI isolation level has been set for T_2 and which one has to be set at least to prevent the Dirty Read. [4 P.]

T_1	T_2
`UPDATE Inventory` `SET Price = Price * 1.20;`	

	`SELECT Book, Seller`
	`FROM Inventory`
	`WHERE Price >= 50;`

	`COMMIT`
`ROLLBACK`	

Lösungsvorschlag:

The result set will be $\{(87, 1), (48, 4), (42, 4)\}$ and therefore the isolation level for T_2 has been set to **Read Uncommited** (DB2's Uncommited Read). To prevent the Dirty Read we have to set it to **Read Commited** (DB2's Cursor Stability).

b) Give short answers to the following questions.

[1 P.] i) Which locks are compatible with the RIX lock?

Lösungsvorschlag:

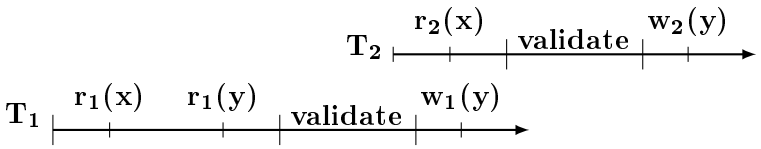
Only with IR.

[2 P.] ii) Which variant of 2PL prevents deadlocks altogether? Briefly explain why.

Lösungsvorschlag:

Conservative 2PL (C2PL) ⇒ every transaction requests all locks before the first operation is executed (preclaiming) this means, if transactions are blocked, they do not hold any locks.

[3 P.] iii) Consider optimistic concurrency control as applied to the following transactions T_1 and T_2:



Explain the outcome of **BOCC** for T_1 and T_2.

Lösungsvorschlag:

T_1 is accepted because there are no other transaction prior to its validation phase.

The BOCC validation of T_2 checks its read set $\{x\}$ with the write set from T_1 $\{y\}$, because T_1 had not been completed before T_2. T_2 is accepted, since $\{x\} \cap \{y\} = \emptyset$.

Part 3: Logging and Recovery

[12 P.]

Consider a database using a no-force/steal/non-atomic recovery procedure with fuzzy checkpoints as described in the lecture. The table below shows the log on disk for a period of time during which 2 checkpoints were taken. The system crashes just after the last log record has been written.
At the time of Checkpoint 1, there were no running transactions and no dirty pages in the system. No pages have been written out to disk after Checkpoint 1.

LSN	Transaction	Type	Page	
– Checkpoint 1 –				
11	T1	BOT		
12	T1	UPDATE	P1	
13	T2	BOT		
14	T3	BOT		
15	T2	UPDATE	P5	
16	T2	COMMIT		
17	T3	UPDATE	P3	
– Checkpoint 2 –				
18	T3	UPDATE	P1	
19	T3	COMMIT		

a) What are the winner transactions? [1 P.]

Lösungsvorschlag:

Transactions 2 and 3.

b) What are the loser transactions? [1 P.]

Lösungsvorschlag:

Transaction 1.

c) At what LSN does the analysis phase begin? Briefly justify your answer. [2 P.]

Lösungsvorschlag:

Directly after the last checkpoint: at LSN 18.

d) At what LSN does the redo phase begin? Briefly justify your answer. [2 P.]

Lösungsvorschlag:

MinDirtyPageLSN: at LSN 12.

The buffer manager stores StartLSN (i.e. the LSN of the first modification since the page had been read from disk) for each modified page. The minimum of all StartLSN is MinDirtyPageLSN. This is the first log entry corresponding to an update that might have been lost.

e) Please list all log records that are undone during undo phase in the order that the system undoes them. Briefly justify your answer. [2 P.]

Lösungsvorschlag:

LSN 12: The only loser transaction is Transaction 1 which only made a single update.

The LSN list (12,11) is also an accepted solution.

[4 P.] f) What information does the system store for Checkpoint 2?

Lösungsvorschlag:

i) MinDirtyPageLSN: LSN 12

ii) all active transactions and their respective BOT LSNs (T1: LSN 11, T3: LSN 14)

Part 4: Distributed Transactions & NoSQL

[12 P.]

a) In a distributed DB, there are three database servers A, B, C. The three servers co-operate in a distributed transaction using the **Two-Phase Commit (2PC)** protocol. A is the coordinator.

i) Suppose that A has sent all the prepare messages but has not yet received a prepared (or abort) message from server B. Would it be correct for A to commit the transaction at this point? Why or why not? [2 P.]

Lösungsvorschlag:

No, server B might be unable to execute the transaction. Perhaps server B sent a prepare-abort that the network lost or delayed, or perhaps server B has crashed.

ii) Same situation, would it be correct for A to abort the transaction at this point, sending abort messages to the servers and an failed message to the client? Why or why not? [2 P.]

Lösungsvorschlag:

Yes, no server has executed the transaction since A hasn't sent any commit messages. Thus it's still ok to abort.

iii) Suppose that server B has received the prepare, sent the ready message, but has not yet received a commit or abort message. Server B contacts server C and discovers that server C has received a commit message. Would it be correct for server B to commit its part of the transaction at this point? Why or why not? [2 P.]

Lösungsvorschlag:

Yes, server A must have sent commit messages to all servers, so it's ok for server B to pretend it received the message.

[2 P.] iv) Suppose again that server B has received the prepare, contacts server C and discovers that server C has received the prepare, sent the ready message, but has not yet received a commit or abort, too. Would it be correct for Server B to commit its part of the transaction at this point? Why or why not?

Lösungsvorschlag:

It would not be correct for B to commit the transaction. Server A might have timed out waiting for one of the ready messages and decided to abort.

b) In NoSQL systems, the BASE paradigm is usually used instead of the ACID paradigm.

[2 P.] i) The E in BASE stands for eventually consistent. What does that mean?

Lösungsvorschlag:

Eventual Consistency is a liveness property that guarantees that in the absence of network partitions the system converges to a consistent state in finite time.

[2 P.] ii) Describe an example application scenario that produces inconsistent behavior due to the lack of causal consistency.

Lösungsvorschlag:

A user posts a response to a comment in a social network. Another user reads the response before the original post is visible to him.

Part 5: Data Warehouse

[15 P.]

a) Consider the following fact table:

Sales

product	region	customer	year	#sales	profit
TV	North	Doe	2000	1	500
Radio	South	Smith	2000	2	100
Radio	East	Bush	1999	1	50
TV	East	Bush	2000	2	1000
Radio	South	Doe	1999	3	150
TV	North	Smith	1999	1	500

i) What is the primary key of this table? [1 P.]

Lösungsvorschlag:

The primary key is the combination of all foreign keys that refer to the primary keys of the dimension tables. Thus, in this case, it is the combination of the attributes product, region, customer and year

ii) Is this fact table cumulative or is it a snapshot table? [1 P.]

Lösungsvorschlag:

This fact table is cumulative because it describes what has happened over a specific period of time (has only additive facts).

iii) Consider the following SQL query: [6 P.]

```
SELECT    product, region, SUM(#sales) as totalSales
FROM      Sales
GROUP BY  GROUPING SETS ((product),(region),());
```

Fill in the result of this query in the empty table below.

Lösungsvorschlag:

product	region	totalSales
TV	ALL	4
Radio	ALL	6
ALL	South	5
ALL	North	2
ALL	East	3
ALL	ALL	10

b) Name three of the four strategies that can be used to extract data from a source into a Data Warehouse. Briefly describe each of these three strategies with a short sentence. [3 P.]

Lösungsvorschlag:

- Snapshots (periodic copying of the database into a data file)

- Log-based (analyses of transaction-log files of the DBMS to detect relevant changes)
- Trigger (activation of triggers in the case of data changes and copying of the changed tuples)
- Usage of DBMS-specific mechanisms for data replication

[2 P.] c) While an operational database is called application-oriented, a Data Warehouse is considered to be subject-oriented. Briefly describe the difference between these two properties.

Lösungsvorschlag:

- Subject-oriented means that all data on one subject is stored within a single system even if it is used by different applications
- Application-oriented means that all data about one application is stored within a single system and data on subjects which are involved in different applications can be stored in multiple systems

[2 P.] d) Determine which properties MOLAP and ROLAP do have by marking the corresponding boxes with a cross (note, each property belongs to exactly one of both concepts).

Lösungsvorschlag:

	MOLAP	ROLAP
fast read access	☒	☐
low storage requirements	☐	☒
many joins	☐	☒
supports galaxy schema	☐	☒

Part 6: Data Mining

[18 P.]

a) Consider the following eight one-dimensional data objects: [6 P.]

$$9, 12, 18, 21, 28, 31, 35, 38$$

Cluster these objects by using the canopy clustering algorithm where

- the two thresholds are set to $T_1 = 8$ and $T_2 = 4$, and
- in each iteration step the smallest object in the candidate list is selected as new centroid.

Describe the clustering algorithm by naming the new centroid, the computed canopy as well as the objects which are removed from the candidate list for each iteration step (see layout shown below).

Lösungsvorschlag:

Clustering of the numbers $9, 12, 18, 21, 28, 31, 35, 38$

1. Iteration Step:

- Centroid: 9
- Canopy: $\{9, 12\}$
- remove $\{9, 12\}$ from the candidate list

2. Iteration Step:

- Centroid: 18
- Canopy: $\{12, 18, 21\}$
- remove $\{18, 21\}$ from the candidate list

3. Iteration Step:

- Centroid: 28
- Canopy: $\{21, 28, 31, 35\}$
- remove $\{28, 31\}$ from the candidate list

4. Iteration Step:

- Centroid: 35
- Canopy: $\{28, 31, 35, 38\}$
- remove $\{35, 38\}$ from the candidate list

- Stop because candidate list is empty.

[12 P.] b) Consider the following five market basket transactions each of which contains some of the six items $\{A, B, C, D, E, F\}$:

k	T_k
001	$\{A, B, D\}$
002	$\{B, C, E\}$
003	$\{B, E\}$
004	$\{A, B, C, D, E\}$
005	$\{A, B, D, F\}$

[8 P.] i) Apply the Apriori algorithm from the lecture to compute all frequent itemsets while using the minimal support threshold $s_{cut} = 0.5$.

Lösungsvorschlag:

Itemsets of size 1:

I_k^1	#	$s(I_k^1)$
$\{A\}$	3	0.6
$\{B\}$	5	1.0
$\{C\}$	2	0.4
$\{D\}$	3	0.6
$\{E\}$	3	0.6
$\{F\}$	1	0.2

Except $\{C\}$ and $\{F\}$ are all 1-itemsets frequent

Itemsets of size 2:

I_k^2	#	$s(I_k^2)$
$\{A, B\}$	3	0.6
$\{A, D\}$	3	0.6
$\{A, E\}$	1	0.2
$\{B, D\}$	3	0.6
$\{B, E\}$	3	0.6
$\{D, E\}$	1	0.2

The frequent 2-itemsets are $\{A, B\}$, $\{A, D\}$, $\{B, D\}$ and $\{B, E\}$.

Remark: According to the lecture two k-itemsets are only combined to a (k+1)-itemset if they agree in their first (k-1) positions (and therefore only disagree in their last position). Therefore the itemset $\{A, B, E\}$ is not considered.

Itemsets of size 3:

I_k^2	#	$s(I_k^2)$
$\{A, B, D\}$	3	0.6
$\{B, D, E\}$	1	0.2

The only frequent 3-itemset is $\{A, B, D\}$.

Itemsets of size 4: Because we only have one frequent 3-itemsets, there cannot be a 4-itemsets which is frequent and the algorithm stops.

In conclusion, the complete set of all frequent itemsets is:

$\{A\}$
$\{B\}$
$\{D\}$
$\{E\}$
$\{A, B\}$
$\{A, D\}$
$\{B, D\}$
$\{B, E\}$
$\{A, B, D\}$

ii) Name two association rules that can be derived from the itemset $\{A, C, E\}$. [2 P.]

Lösungsvorschlag:

Since neither the premise nor the conclusion are allowed to be empty and both sets need to be disjoint, the total number of association rules that can be derived from this itemset is six, namely:

1.	$\{A\} \rightarrow \{C, E\}$
2.	$\{C\} \rightarrow \{A, E\}$
3.	$\{E\} \rightarrow \{A, C\}$
4.	$\{A, C\} \rightarrow \{E\}$
5.	$\{A, E\} \rightarrow \{C\}$
6.	$\{C, E\} \rightarrow \{A\}$

iii) Compute support and confidence of the association rule r: $\{A, B\} \rightarrow \{D\}$. [2 P.]

Lösungsvorschlag:

- Support of r:

$$s(r) = s(\{A, B, D\}) = 0.6$$

- Confidence of r:

$$c(r) = \frac{s(\{A, B, D\})}{s(\{A, B\})} = \frac{0.6}{0.6} = 1.0$$

Part 7: Uncertain Databases

[17 P.]

a) Consider the following possible worlds representation of a relational database table called *Person*:

W_1, Pr(W_1)=0.6

	WK	name
t_1	p1	Alice
t_3	p2	Bob
t_4	p3	Charly

W_2, Pr(W_2)=0.3

	WK	name
t_2	p1	Ally
t_3	p2	Bob

W_3, Pr(W_3)=0.1

	WK	name
t_1	p1	Alice
t_5	p3	Charles

[2 P.] i) In which way do the two tuples t_3 and t_4 depend on each other? Briefly justify your answer.

Lösungsvorschlag:

Every world that contains tuple t_4 also contains tuple t_3. Thus, the existence of tuple t_4 implicates the existence of tuple t_3 (i.e. t_3 is positively implicated by t_4).

[2 P.] ii) What is the marginal probability of tuple t_1? Briefly justify your answer.

Lösungsvorschlag:

Tuple t_1 exists in the two possible worlds W_1 and W_3. Therefore, it has the marginal probability $p(t_1) = Pr(W_1) + Pr(W_3) = 0.7$

[2 P.] iii) Can this possible worlds representation be modeled with a BID-database? Briefly justify your answer.

Lösungsvorschlag:

Since the two tuples t_3 and t_4 are neither independent nor exclusive and BID-databases are not able to model other kinds of dependencies, this possible worlds representation cannot be modeled with a BID-database.

[5 P.] iv) Model this possible worlds representation by using a pc-database. Tip: Limit the world-table to one variable.

Lösungsvorschlag:

Person

	RK	WK	name	condition
t_1	1	p1	Alice	X=1 v X=3
t_2	2	p1	Ally	X=2
t_3	3	p2	Bob	X=1 v X=2
t_4	4	p3	Charly	X=1
t_5	5	p3	Charles	X=3

World-Table

var	value	Prob
X	1	0.6
X	2	0.3
X	3	0.1

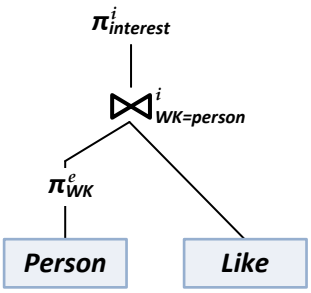
b) Consider the following BID-database and extensional query plan:

Person

	RK	WK	name	p
t_1	1	p1	Trump	0.8
t_2	2	p1	The Donald	0.2
t_3	3	p2	Merkel	1.0
t_4	4	p3	Schulz	0.4
t_5	5	p3	Scholz	0.4

Like

	RK	person	interest	p
t_6	1	p1	golf	1.0
t_7	2	p2	science	0.7
t_8	3	p3	birthdays	0.8
t_9	4	p2	rhombi	1.0
t_{10}	5	p1	science	0.2



Evaluate this query plan operator by operator and fill in the probabilities of the resultant tuples in the tables given below.

i) Subquery $Q_1 = \pi^e_{WK}(Person)$ [1.5 P.]

Lösungsvorschlag:

	WK	p
t_{11}	p1	0.8 + 0.2 = 1.0
t_{12}	p2	1.0
t_{13}	p3	0.4 + 0.4 = 0.8

ii) Subquery $Q_2 = Q_1 \bowtie^i_{WK=Person} Like$ [2.5 P.]

Lösungsvorschlag:

	WK	RK	person	interest	p
t_{15}	p1	1	p1	golf	1.0 x 1.0 = 1.0
t_{16}	p1	2	p1	science	1.0 x 0.2 = 0.2
t_{17}	p2	3	p2	rhombi	1.0 x 1.0 = 1.0
t_{18}	p2	4	p2	science	1.0 x 0.7 = 0.7
t_{19}	p3	5	p3	birthdays	0.8 x 0.8 = 0.64

iii) Subquery $Q_3 = \pi^i_{interest}(Q_2)$ [2 P.]

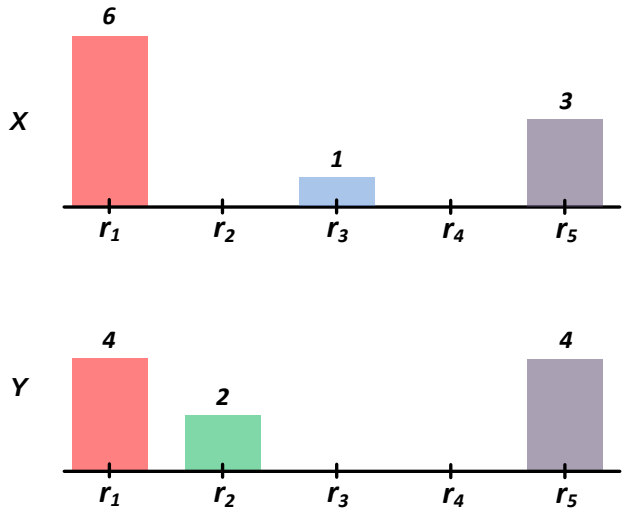
Lösungsvorschlag:

	interest	p
t_{19}	golf	1.0
t_{20}	science	1 - (1-0.2) x (1-0.7) = 1 – 0.8 x 0.3 = 0.76
t_{21}	rhombi	1.0
t_{22}	birthdays	0.64

Part 8: Similarity Search of Multimedia Objects

[10 P.]

a) Consider the following two feature signatures X and Y:



[3 P.] i) Compute the Earth Mover's Distance between X and Y by using the ground distance function $\delta(r_i, r_j) = |i - j|$.

Lösungsvorschlag:

$$\begin{aligned} EMD_\delta(X, Y) &= \frac{1}{10}\big(f(r_1, r_1) \times \delta(r_1, r_1) + f(r_1, r_2) \times \delta(r_1, r_2) \\ &\quad + f(r_3, r_5) \times \delta(r_3, r_5) + f(r_5, r_5) \times \delta(r_5, r_5)\big) \\ &= \frac{1}{10}(4 \times 0 + 2 \times 1 + 1 \times 2 + 3 \times 0) = 0.4 \end{aligned}$$

[3 P.] ii) Compute the Independent Minimization Lower Bound from X to Y by using the ground distance function $\delta(r_i, r_j) = |i - j|$.

Lösungsvorschlag:

$$\begin{aligned} LB_{IM}(X, Y) &= \frac{1}{10}\big(f(r_1, r_1) \times \delta(r_1, r_1) + f(r_1, r_2) \times \delta(r_1, r_2) \\ &\quad + f(r_3, r_2) \times \delta(r_3, r_2) + f(r_5, r_5) \times \delta(r_5, r_5)\big) \\ &= \frac{1}{10}(4 \times 0 + 2 \times 1 + 1 \times 1 + 3 \times 0) = 0.3 \end{aligned}$$

b) Why does storing feature histograms require less memory than storing feature signatures? [2 P.]

Lösungsvorschlag:

Feature histograms are defined on a predefined set of representatives so that they only differ in their weights. In contrast, each feature signature can be defined on another set of representatives and therefore can differ in their representatives and weights. Since feature histograms share the same representatives, we only need to store their weights. In contrast, since each feature signature can have another set of representatives, we need to store both, the representatives and the weights, which obviously requires more memory.

c) When does a distance function $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}^{\geq 0}$ satisfy the triangle inequality? Complete the formal definition below. [2 P.]

Lösungsvorschlag:

A distance function $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}^{\geq 0}$ satisfy the triangle inequality, iff $\forall x, y, z \in \mathbb{X}\colon \delta(x, y) \leq \delta(x, z) + \delta(z, y)$