Group **A**

Please fill in your name and registration number (Matrikelnr.) **immediately**.

Exam 3 ON	**SOLUTION KEY**	27.06.2019
	Advanced Database Systems (**184.780**)	**GROUP A**
Matrikelnr.	Last Name	First Name

Duration: 75 minutes. Provide the solutions on the designated pages. **Good Luck!**

Question 1: (4)

For each of the statements below, decide if it is true or false and tick the corresponding circle. You get +1 credit for each correct answer, −1 credit for each wrong answer and 0 credit if you leave the answer open. In total, you always get ≥ 0 credits on the entire exam question 1.

1. In the Two-Phase Commit (2PC) Protocol, if all subordinates participating in a transaction crash in the termination phase, then the transaction must be aborted no matter what messages have been exchanged during the voting phase. true ◯ false ⊗
2. If a column store applies dictionary encoding, then this may turn a variable-width column into a fixed-width column. true ⊗ false ◯
3. The following NoSQL-systems are usually classified as follows: Redis is a key-value store, MongoDB is a document store, Riak is a wide column store, and Neo4J is a graph database. true ◯ false ⊗
4. Both scalar clocks and vector clocks satisfy the weak clock property. true ⊗ false ◯

Question 2: (4)

Consider a relational database with the following schema:

Customers (<u>id</u>, fname, lname, pcode, city) and
Sales (<u>cust_id</u>, <u>product</u>, qty, amount),

where cust_id is a foreign key from the Sales table to the Customers table.

Suppose that these tables have the following content:

Customers				
id	**fname**	**lname**	**pcode**	**city**
C1	Alice	Smith	1040	Wien
C2	Bob	Jones	6020	Innsbruck
C3	Carol	Hunt	5020	Salzburg

Sales			
cust_id	**product**	**qty**	**amount**
C1	1234	1	300.00
C1	8150	3	150.00
C2	3456	2	160.00
C2	6789	1	150.00

Give a representation of this relational database **as a document store (in JSON)**, where you should choose an appropriate denormalization to optimize the following kind of queries:

(a) Suppose that the most time-critical kind of queries asks for information on the Customers including the list of products that each customer has bought. You only need to display the collection corresponding to the Customers table. [2 credits]

```
db.customers.insert([
{ _id:  "C1",
fname:  "Alice",
lname:  "Smith",
pcode:  "1040",
city:  "Wien",
sales:  [ "1234", "8150" ] },
{ _id:  "C2",
fname:  "Bob",
lname:  "Jones",
pcode:  "6020",
city:  "Innsbruck",
sales:  [ "3456", "6789" ] },
{ _id:  "C3",
fname:  "Carol",
lname:  "Hunt",
pcode:  "5020",
city:  "Salzburg" }
]);
```

(b) Now suppose that the most time-critical kind of queries asks for information on the Sales including the lname and pcode of the customer who bought that product. You only need to display the collection corresponding to the Sales table. [2 credits]

```
db.sales.insert([
{ _id:  "S1",
cust_id:  "C1",
product:  "1234",
qty:  1,
amount:  300.00,
lname:  "Smith",
pcode:  "1040" },
{ _id:  "S2",
cust_id:  "C1",
product:  "8150",
qty:  3,
amount:  150.00,
lname:  "Smith",
pcode:  "1040" },
{ _id:  "S3",
cust_id:  "C2",
product:  "3456",
qty:  2,
amount:  160.00,
lname:  "Jones",
pcode:  "6020" },
{ _id:  "S3",
cust_id:  "C2",
product:  "6789",
qty:  1,
amount:  150.00,
lname:  "Jones",
pcode:  "6020" }
]);
```

Question 3: (4)

A friend of yours works in the IT department of a world-wide operating bank. The management of the bank has recently decided to move the payment processing database to a cluster solution. Your friend is responsible for choosing an appropriate database system for this purpose. However, he/she is a bit worried about the CAP-Theorem and even more so about the PACELC extension of the CAP-Theorem. Now he/she asks you for advice, what kind of system in the PACELC categorization might be best suited.

Please discuss the various categories of database systems in the PACELC categorization and choose one that you find best suited for this bank application. Argue why you find it better suited than the others.

Recall that PACELC refers to the availability-consistency (A-C) tradeoff in case of network partition and else the trade-off between latency and consistency (L-C). In case of a bank application, it seems as if consistency is the most critical property. Hence, a PC/EC system seems best suited, i.e., prefer consistency over availability in case of network partition and else prefer consistency over (low) latency. This means that the system stops providing service in case of network partition to avoid inconsistencies. Likewise, in normal operation, it ensures consistency even if this is at the expense of high latency. The other options would have been the following:

- PA/EL, i.e, prefer availability over consistency in case of network partition and else prefer (low) latency over consistency.
- PA/EC, i.e, prefer availability over consistency in case of network partition and else prefer consistency over (low) latency.
- PC/EL, i.e, prefer consistency over availability in case of network partition and else prefer (low) latency over consistency.

A PC/EL system or even PA/EL might be an option e.g. for telebanking applications, where the user might get annoyed by long response times of the system and - above all - where the user is given the possibility to check and correct the final result of the transaction before actually committing it.

Question 4: (4)

In this question you are asked to compute the communication cost (number of bytes transferred) for the given query and three given join strategies. **Project as early as possible in all cases** to achieve the minimal communication cost. `Flight.acid` is a foreign key to `Aircraft`, i.e., **the join has a selectivity of** $\frac{1}{5\,000}$.

$$\pi_{pilot,weight}(Flight \bowtie_{id=acid} Aircraft) \tag{1}$$

Assume a distributed database with 3 sites and the relations `Flight` and `Aircraft`. See the following tables for details of the scenario. You can assume that the relations are *non-fragmented* and all records (and attributes) are *fixed-size*.

Relation	Site	# Records	Record Size (byte)
Flight	1	1 000 000	300
Aircraft	2	5 000	2 000

Relation	Attribute	Size
Flight	acid	20
Flight	pilot	20
Flight	code	30
Aircraft	id	20
Aircraft	active	4
Aircraft	weight	10

(a) Compute the communication cost if the join is computed at site 3. [1 credits]

`We use MB/kB as shorthand for` 10^6 `and` 10^3 `bytes respectively in this sample solution.`

$$Size_{Flight} = 10^6 \cdot 40 = 40MB$$
$$Size_{Aircraft} = 5 \cdot 10^3 \cdot 30 = 150kB$$
$$Total = Size_{Flight} + Size_{Aircraft} = 40,15MB$$

(b) Compute the communication cost if the join is computed at site 1 and the result then transferred to site 3. [1 credits]

$$Size_{Join} = 10^6 \cdot (30) = 30MB$$
$$Total = Size_{Join} + Size_{Aircraft} = 30,15MB$$

(c) Compute the communication cost if the join is computed by first computing $Aircraft \ltimes Flight$ at site 2, then using the result to compute the join at site 1 and finally transferring the result to site 3. **Assume a selectivity of** $\frac{1}{5}$ **for the semi join.** (Don't forget about projecting away unnecessary fields.)

[2 credits]

Naive approach:

$$Size_{Flight.acid} = 10^6 \cdot 20 = 20MB$$
$$Size_{Semijoin} = 1\,000 \cdot 30 = 30kB$$
$$Total = Size_{Flight.acid} + Size_{Semijoin} + Size_{Join} = 50,03MB$$

In fact we could do much better by observing that because acid is a foreign key, the projection will contain at most 5000 (instead of 10^6) records. Furthermore, because of the semi join selectivity we can deduce that there are in fact only 1000 distinct values for acid in the Flight table. We could thus reduce our estimate for $Size_{Flight.acid}$ to $1000 \cdot 20 = 20kB$.

Question 5: (4)

(a) Evaluate the following *Cypher* query on the Database given on the last sheet of the exam:

```
MATCH ({name:'Vienna'})-[r:Bus|Air]-(l)
RETURN type(r), l.name
```

Bus, Cosenza
Air, London

(b) Evaluate the following *Cypher* query on the Database given on the last sheet of the exam:

```
MATCH p = shortestPath( (a {weather: 'rain'})-[*]->(b {weather:'sun'}) )
RETURN a.name, b.name, length(p);
```

a.name	b.name	length(p)
Cosenza	Vienna	2
Cosenza	Budapest	1
Sao Paolo	Vienna	3
Sao Paolo	Budapest	2
Kuala Lumpur	Tokyo	1

(c) *Assume the data model described on the last sheet of the exam.* Write a *Cypher* query that returns all the cities that have at least one outgoing edge and a `temperature` less or equal to 27.

```
MATCH (n:City)-->()
WHERE n.temperature <= 27
RETURN n
```

(d) *Assume the data model described on the last sheet of the exam.* Write a *Cypher* query that returns all the subgraphs that have the following shape. **Make sure that all three nodes are distinct.**:

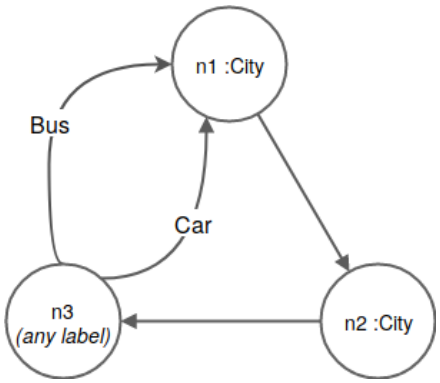

Figure 1: Subgraph for Query

```
MATCH (n3:City)-[:Bus]->(n1:City)-->(n2)-->(n3)-[:Car]->(n1)
WHERE n1 <> n3 and n2 <> n1 and n2 <> n3
RETURN n1,n2,n3
```

Overall: 20 points

Sheet for Question 5. You can tear off this sheet if you like.

Graph DB Data Model

The graph contains nodes for cities with the `:City` label. They all contain at least the fields `weather`, `temperature`, and `name`. A directed edge from node a to node b indicates a mode of transport – given by the type of the edge, e.g., `:Air`, `:Car`, etc. – from the city of node a to the city of node b.

Graph Database

The labels of the nodes contain the `name` *and* `weather` *attributes of the city.*

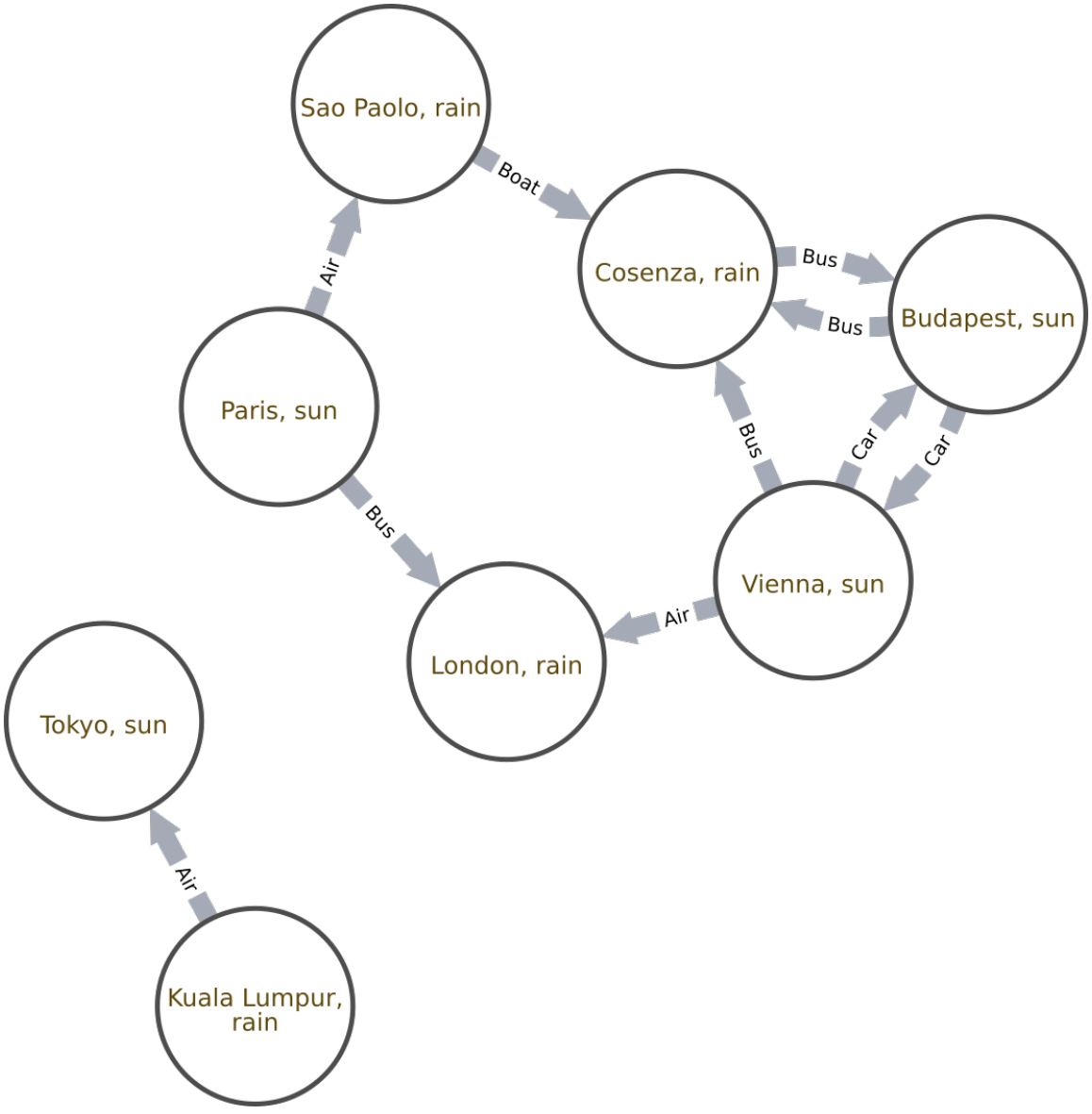

Figure 2: Database for Question 5