Data Representation
Quiz Solution

Gabor Gevay, Dr. Asterios Katsifodimos
Prof. Dr. Volker Markl
Alexander Alexandrov, Stephan Ewen,
Kostas Tzoumas, Fabian Hueske, Max Heimel

Fachgebiet Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

http://www.dima.tu-berlin.de/

Question 3.1

- Which of the following statements concerning the data types CHAR(100) and VARCHAR(100) is true?

CHAR(100) can be accessed more efficiently than VARCHAR(100).
CHAR(100) is more restrictive in the charsets it supports than VARCHAR(100).
CHAR(100) requires less disk space than VARCHAR(100) for arbitrary strings.

Question 3.2

- Which of the following statements concerning the data types Double and DECIMAL is true?

DECIMAL(2,1) and Double require the same amount of disk space.
DECIMAL can only store positive values in contrast to DOUBLE.
The DECIMAL data type should be preferred over DOUBLE for applications that require precision guarantees.

Question 3.3

- Please decide which of the following statements are true regarding logical addresses for records. Logical addresses are translated to physical addresses using a mapping table.

- Each correct answer gives you 1 point, each wrong answer -1. The minimum amount of points for this question is 0.

Logical addresses are used to simplify and unify the management of stored data.
Logical addresses are used to improve the efficiency of data accesses.
Caches and indexes are among the main reasons for using logical addresses.
Transactions and recovery mechanisms are among the main reasons for using logical addresses.
Systems that use logical addresses can answer SELECT queries more efficiently than those that don't.

Questions 3.4 & 3.5

- In the lecture you have gotten to know the n-ary storage model, PAX, and columnar storage.
- Please rank these models according to their suitability for OLAP from best to worst (1 = best, 2 = second, 3 = worst).
- *To get to the answer, think about two points:*
  - *Cost of an Insert-Operation (OLTP characteristic) ?*
  - *Cost of data retrieval and filtering (OLAP characteristic) ?*

	OLAP	OLTP
NSM	3	1
PAX	2	2
Columnar Storage	1	3

Questions 3.6-3.9

- Given is the following relation:
  - `Person( ID INTEGER, LastName CHAR(20), FirstName CHAR(20), Birthday DATE, Gender CHAR(1), Address CHAR(100) );`
- The data types have the following storage requirements:
  - INTEGER: 4 Byte, CHAR(1): 1 Byte, and DATE: 8 Byte.
- The Person relation holds 100,000,000 (100 million) records.
- Furthermore given is the following query:
  - ```
    SELECT FirstName, LastName, Gender
    FROM Person
    WHERE Birthday > '18-05-2010';
    ```
- The relation is stored on a disk with the following properties:
  - Blocksize: 16KB
  - Blocks per Cylinder: 512 (assume all cylinders have the same amount of blocks!)
  - Time to transfer one block: 0.1ms
  - Seek time: 10ms (remember: switching a cylinder requires seeking as well!)
    ```

Question 3.6

A column layout is used to store the Person relation. The storage layout does not allow to write attributes across block borders. A block starts with a header of 128 Byte. Inserts are attached at the end and all blocks are filled as much as possible. How long does it take to read the data from disk that is required to answer the query stated above? Give your result rounded to seconds.

Variable	Description	Value
C	Cardinality	100,000,000
BS	Block Size	16KB (= 16384B)
BH	Block header size	128B
BC	Blocks per cylinder	512B
BR	Block reading time	0.1ms
ST	Seek time	10ms

Attr. Size (AS)	Val/Block (VB)	Blocks (B)	Cylinder Hops (CH)	Reading Time (RT)
	(BS-BH) / AS	C / VB	B / BC	B*BR + CH*ST
20B	812.8 -> 812	123152.7 -> 123153	240.5 -> 241	14725.3 ms
8B	2032	49212.5 ->49213	96.1 -> 97	5891.3 ms
1B	16256	6151.5 -> 6152	12.01 -> 13	745.2 ms

Total Time: RT(20Byte)*2 + RT(8Byte) + RT(1Byte) = 36087.1 ms -> 36s

Question 3.7

A NSM layout is used to store the Person relation. The storage layout does not allow to write records across block borders. A block starts with a header of 128 Byte. Assume the data was imported and no updates or deletions were performed. All blocks are filled as much as possible. How long does it take to read the data of the relation from disk with a full scan? Give your result rounded to seconds.

Variable	Description	Value
C	Cardinality	100,000,000
BS	Block Size	16KB (= 16384B)
BH	Block header size	128B
BC	Blocks per cylinder	512B
BR	Block reading time	0.1ms
ST	Seek time	10ms

Rec. Size (RS)	Rec./Block (RB)	Blocks (B)	Cylinder Hops (CH)	Reading Time (RT)
	(BS-BH) / RS	C / RB	B / BC	B*BR + CH*ST
153B (Person)	106.2 -> 106	943396.2 -> 943397	1842.5 -> 1843	112769.7 ms

Total Reading Time: 112769.7 ms -> 113 s

Question 3.8

A NSM layout is used to store the Person relation. The storage layout does not allow to write attributes across block borders. A block starts with a header of 128 Byte. Assume the data was imported and no updates or deletions were performed. All blocks are fill as much as possible. A secondary index on the attribute 'Birthday' was created. How long does it take to read the data of the relation from disk using the index? Do not take time to access the index into account. Compute the time that is required to retrieve the data from disk. The cache is filled and will give a hit with a probability of 20%. Assume that the elevator algorithm does not prevent any seek operations. Assume that the filter predicate "Birthday > '18-05-2010' has a selectivity of 0.002. That means that only 0.2% of all tuples satisfy the filter predicate and 99.8% of all tuples are filtered out.

Variable	Description	Value
C	Cardinality	100,000,000
S	Selectivity	0.002
CH	Cache Hit Rate	0.2
BR	Block reading time	0.1ms
ST	Seek time	10ms

	Req. Records (RR)	Records to Read (RD)	Reading Time (RT)
Formula	C * S	RR * (1-CH)	RD * (BR+ST)
	200,000	160000	1616000 ms

Total Reading Time: 1616000ms -> 1616 s
This answer assumes that RID have not been sorted

Question 3.8 (assuming sorted RIDs)

- If RID have been sorted:
  - Still 160,000 records to read
  - But number of seeks is reduced to number of cylinder hops!

  - 160,000 * BlockReadTime + CylinderHops * SeekTime
  - 160,000 * 0.1ms + 1843 * 10ms = 34.43s -> 34s

- Both answers are rated as correct

Question 3.9

A NSM layout is used to store the Person relation. The storage layout does not allow to write attributes across block borders. A block starts with a header of 128 Byte. Assume the data was imported and no updates or deletions were performed. All blocks are fill as much as possible. A secondary index on the attribute 'Birthday' was created. How long does it take to read the data of the relation from disk using the index. Do not take time to access the index into account. The cache is empty, so no cache hits occur. Assume that the elevator algorithm does not prevent any seek operations. Assume that the filter predicate "Birthday > '18-05-2010' has a selectivity of 0.002. That means that only 0.2% of all tuples satisfy the filter predicate and 99.8% of all tuples are filtered out.

Variable	Description	Value
C	Cardinality	100,000,000
S	Selectivity	0.002
BR	Block reading time	0.1ms
ST	Seek time	10ms

	Rec. to read (RR)	Reading Time (RT)
Formula	C * S	RR * (BR+ST)
	200,000	2020000ms

Total Reading Time: 2020000 ms -> 2020 s
Again: answer assumes that RIDs have not been sorted!

Question 3.9 (assuming sorted RIDs)

- If RID have been sorted:
  - Still 200,000 records to read
  - But number of seeks is reduced to number of cylinder hops!

  - 200,000 * BlockReadTime + CylinderHops * SeekTime
  - 200,000 * 0.1ms + 1843 * 10ms = 38.43s -> 38s

- Both answers are rated as correct

Questions 3.10, 3.11, and 3.12

Question		
Data stored in columnar layouts can be compressed more efficiently.	TRUE	FALSE
Column stores have better performance for inserting tuples than traditional row stores.	TRUE	FALSE
Disk accesses can be reduced by column stores.	TRUE	FALSE