Name

CWID

Final Exam

May 3rd, 2018
10:30-12:30

CS520 - Data Integration, Warehousing, and Provenance

Please leave this empty!

1.1 1.2 1.3 Sum

Instructions

- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**

Consider the following datawarehouse schema (star schema) and partial example instance. There is a single fact table (`warehouse`) about items stored in a warehouse. Each row in this fact table stores the quantity of a certain product (e.g., 3 Samson Galaxy phones) stored at a particular location and time. For each such set of products we also record which supplier did supply the product. There are four dimension tables corresponding to the following dimensions:

- **Time** with three levels (year, month, day)
- **Location** with four levels (state, city, zip, street)
- **Supplier** with two levels (type, sname)
- **Product** with three levels (category, brand, pname, price) where pname is the finest granularity and brand and category are not comparable (some brands can have products from multiple categories and categories obviously can contain have products from different brands). The same holds for price and brand and price and category.

warehouse

TID	LID	SID	PID	numItems
1	4	1	1	15
2	1	5	2	10
100	1	76	4	22
...	...	...	...	...

timeDim

TID	year	month	day
1	2010	1	1
2	2010	1	2
...	...	...	...
...	2018	5	1

locationDim

LID	state	city	zip	street
1	Illinois	Chicago	60616	10 W 31st
2	Illinois	Chicago	60615	900 Cottage Grove
2	Lousiana	New Orleans	42345	12 Mark street
...	...	...	...	...

suppliedDim

SID	type	sname
1	electronics	Nokiso
2	houseware	Saemens
...	...	...

productDim

PID	category	brand	pname	price
1	computers	Apple	MacBook	1300
2	computers	Dell	Inspire	1000
3	smartphones	Samsung	Galaxy 1	600

Hints:

- Attributes with black background form the primary key of a relation (e.g., `PID` for relation `productDim`)
- Attributes `LID`, `TID`, `PID`, and `CID` in the fact table are foreign keys to the dimension tables

Part 1.1 Data Warehousing (Total: 40 Points)

Recall that you should write all queries according to the schema and not according to the example instance.

Question 1.1.1 (6 Points)

Write an SQL query that returns the 3 pairs of year and state with the highest total number of products in the warehouses in that state during that year. Note that we are dealing with snapshot facts here, since the fact table records for every TID a snapshot of the various warehouses recorded in the database and an item may contribute to multiple snapshots.

Question 1.1.2 (6 Points)

Write an SQL query that returns the total value (the number of items multiplied by the price) of products per supplier for all Apple products stored in warehouses at January 1st of 2018.

Question 1.1.3 (7 Points)

Write an SQL query that returns the number of products (numItems) in total, per state, per city, and per zip code stored in warehouses at January 1st of 2018.

Question 1.1.4 (7 Points)

Write an SQL query that returns for each year the change in the average of the total number of items in all warehouses. Return the year, the difference in average to the previous year, the average for this year, and the average for the following year.

Question 1.1.5 (7 Points)

Write an SQL query that returns the number of months during which the average of the total products in warehouses is more than 100,000.

Question 1.1.6 (7 Points)

Write an SQL query that returns the three cities with the highest average of the maximum of the total number of items per year.

Part 1.2 Big Data (Total: 30 Points)

Question 1.2.1 (12 Points)

Consider a dataset of key-value pairs `(ssn,state)` recording SSN of taxpayers and the state they live in. Describe a MapReduce workflow that computes the number of tax payers per state. First explain the workflow and then provide pseudocode for the map and reduce functions of your workflow.

Question 1.2.2 (5 Points)

Explain how group-by aggregation can be implemented using the MapReduce programming model.

Question 1.2.3 Fault Tolerance (4 Points)

Explain in a few sentences why load balancing is critical for distributed systems to scale.

Question 1.2.4 Distributed file systems (5 Points)

- ❑ HDFS automatically detects when a data node is down
- ❑ Writing of files in HDFS is append-only
- ❑ HDFS does not rely on replication to achieve fault tolerance
- ❑ HDFS scales well to large number of files
- ❑ In HDFS, clients communicate both with the name node as well as with data nodes

Question 1.2.5 MapReduce and Hadoop (4 Points)

- ❑ The map function in MapReduce is applied to single key-value pairs from the input.
- ❑ The map phase in MapReduce does not require any communication among workers
- ❑ Hadoop MapReduce uses an external merge sort algorithm to sort the input of reducers on their keys
- ❑ A shuffle only requires network communication, but not disk I/O

Part 1.3 Provenance (Total: 30 Points)

For the following the queries over the schema shown below, compute the provenance according to the following provenance models for all their result tuples.

- Why-Provenance
- Minimal Why-Provenance
- Provenance Polynomials

Before presenting provenance, show the results for each query first and label the result tuples $t_1, t_2, \ldots, t_n$.

Consider the following database schema and instance:

location

lName	city	owner	sizeSf	
Windsor Castle	Windsor	Queen	40,000	l_1
Big Ben	London	Public	3,500	l_2
Stonehedge	Amesbury	Public	14,000	l_3

account

witness	suspect	crimeId	
Bob	Peter	1	a_1
Peter	Bob	1	a_2
Queen	Bob	2	a_3

crime

crimeId	lName	time	type	victim	
1	Big Ben	10:30	murder	Alice	c_1
2	Windsor Castle	11:00	theft	Queen	c_2

Question 1.3.1 (5 Points)

$$\pi_{city}(\sigma_{sizeSf>10{,}000}(location))$$

Question 1.3.2 (8 Points)

$$q_1 \stackrel{def}{=} (location \bowtie account \bowtie crime)$$
$$q \stackrel{def}{=} \pi_{type,time,city}(q_1)$$

Question 1.3.3 (8 Points)

$$q_1 \stackrel{def}{=} \rho_{s1 \leftarrow suspect, w1 \leftarrow witness}(account)$$
$$q_2 \stackrel{def}{=} \rho_{s2 \leftarrow suspect, w2 \leftarrow witness}(account)$$
$$q \stackrel{def}{=} \pi_{crimId, w1, w2}(\sigma_{w1 \neq w2}(crime \bowtie q_1 \bowtie q_2))$$

Question 1.3.4 (9 Points)

$$q \stackrel{def}{=} \pi_{city}(\sigma_{type='murder'}(location \bowtie crime)) \cup \pi_{city}(\sigma_{victim='Queen'}(location \bowtie crime))$$