StudentName: COMPSCI 351

StudentID:

TEST 1 - Model Solutions

SECOND SEMESTER 2022/2023

COMPUTER SCIENCE

Fundamentals of Database Systems

Time Allowed: FORTY FIVE (45) minutes

NOTE:

- The test is closed book.
- No calculators are permitted.
- Attempt ALL questions in this test.
- A maximum of 30 marks is available in this test.

1. **The Relational Model of Data.**

   (a) Consider the relation schema ITEMS. It stores the *name*, *category* and *age* of items. Write down a single relation over ITEM that

   - satisfies the two **keys** $\{name\}$ and $\{category, age\}$,
   - violates all superkeys not contained in the two keys above, and
   - has as few tuples as possible.

   [3 marks]

   Solution:
   We need to show that neither $\{category\}$ nor $\{age\}$ form keys, so the first and second tuple of the following relation agree on *category*, and the second and third tuple of the following relation agree on *age*. 1 marks for each of these tuple pairs, and 1 mark for not introducing anything else.

name	*category*	*age*
Smurf	Toy	10
Football	Toy	2
Towel	House	2

   (b) Consider the relation schema ITEM from before, as well as the relation schema TRADE with attributes *name*, *category*, *trader*, and *date* expressing that a trader trades an item (uniquely identified by its *name* and *category*) on some date. Write down a single instance over {ITEM,TRADE} that

   - violates the **foreign key** $[name, category] \subseteq \text{ITEM}[name, category]$ on TRADE, and
   - for every tuple over TRADE there is a tuple over ITEM with matching values on *category*
   - has as few tuples as possible. [2 marks]

   Solution:
   Since the foreign key is violated, we need at least 1 tuple in the instance over TRADE, and since a match is required on *category*, we need 1 tuple in the instance over ITEM. If we choose non-matching values over *name*, then the foreign key is violated. 0.5 marks for violating the foreign key, 0.5 marks for satisfying the match on *category*, and 1 mark for not introducing anything else and having only two tuples.

TRADE

trader	*name*	*category*	*date*
Troy	Smurf	Toy	01 May 2023

ITEM

name	*category*	*age*
Football	Toy	8 years

2. **SQL DDL and DML.**

(a) Specify an SQL table schema that permits precisely the same instances as the relation schema ITEM={*name*,*category*,*age*} with keys {*name*,*category*} and {*name*,*age*}. In particular, choose the domain `VARCHAR` for all attributes. [3 marks]

Solution: Different solutions are possible, but 1 mark for the correct definition of attributes, and 1 mark each for specifying each key correctly. Note that all attributes must be specified NOT NULL, either explicitly or implicitly by the PRIMARY key.

```
CREATE TABLE Item (
    name VARCHAR,
    category VARCHAR,
    age VARCHAR NOT NULL,
    PRIMARY KEY(name, category),
    UNIQUE(name, age) );
```

(b) In addition to the table schema ITEM above, assume we have the following table schema TRADE:

```
CREATE TABLE Trade (
    trader VARCHAR,
    name VARCHAR,
    category VARCHAR,
    date DATE,
    PRIMARY KEY(trader, date),
    FOREIGN KEY(name, category) REFERENCES Item ON DELETE SET NULL );
```

and the following instance over {TRADE,ITEM}:

TRADE

trader	*name*	*category*	*date*
Sandra	Towel	House	10 March 2023
Julia	Smurf	Toy	15 May 2023
Joe	Football	Toy	3 June 2023

ITEM

name	*category*	*age*
Smurf	Toy	10
Football	Toy	2
Towel	House	2

Write down the instance after the following update operation has been performed:

`DELETE FROM` ITEM i `WHERE` i.*age*='2';

[2 marks]

Solution:
1 mark for removing both tuples in the instance over ITEM that have '2' in the column *age*, and 0.5 marks each for setting the correct referencing tuples to null.

TRADE

trader	*name*	*category*	*date*
Sandra	null	null	10 March 2023
Julia	Smurf	Toy	15 May 2023
Joe	null	null	3 June 2023

ITEM

name	*category*	*age*
Smurf	Toy	10

3. **SQL.** Consider the relational database schema {Item, Trader, Trade} as given below:
   - Item={name, category, age} with key {name}
   - Trader={trader, location} with key {trader}
   - Trade={trader, name, date} with key {trader, date} and foreign keys
     - [name] ⊆ Item[name]
     - [trader] ⊆ Trader[trader].

   The schema stores trade items including their name, category, and age; traders including their trade name, and location; and trades including the name of an item a trader traded on some date.

   (a) Write an **English language description** of the following query:

   ```
   SELECT t.name, COUNT(*) AS number_of_trades
   FROM Trade t, Item i
   WHERE t.name=i.name AND i.category = 'Toy'
   GROUP BY t.name
   HAVING COUNT(*) >= 100;
   ```

   [4 marks]

   Solution: 1 mark for each of the following:
   - For each name of an item
   - from the category 'Toy',
   - list its number of trades,
   - provided there are at least 100.

   (b) Write the following query in SQL: What is the name of traders from the location of 'Suzhou' who only order items from the category 'Food'? [3 marks]

   Solution:
   Different solutions are possible of course, but roughly 1 mark for returning the correct output schema and using city 'Suzhou' and category 'Food', 1 mark for using correct join conditions on the right tables, and 1 mark for ensuring that they only trade items of category 'Food'.

   ```
   SELECT t.trader
   FROM Trader t
   WHERE t.location='Suzhou' AND NOT EXISTS (
       SELECT *
       FROM Trade t1
       WHERE t1.trader=t.trader AND NOT EXISTS (
           SELECT *
           FROM Item i
           WHERE i.name=t1.name AND i.category='Food'));
   ```

(c) Write the following query in SQL: What is the location of traders who have traded items with the same name on at least two different dates? [3 marks]

Solution:
Different solutions are possible of course, but roughly 0.5 marks for returning the correct output schema, 0.5 marks for making sure items with the same name have been traded, 2 marks for ensuring the joins use correct join attributes and the same trader traded both items.

```
SELECT t.location
FROM   Trade t1, Trade t2, Trader t
WHERE  t1.name=t2.name AND t1.trader=t2.trader AND
       t.trader=t1.trader AND t1.date <> t2.date
```

4. **Relational algebra.** Consider the relational database schema {Item, Trader, Trade} as given below:
   - Item={name, category, age} with key {name}
   - Trader={trader, location} with key {trader}
   - Trade={trader, name, date} with key {trader, date} and foreign keys
     - [name] $\subseteq$ Item[name]
     - [trader] $\subseteq$ Trader[trader].

   The schema stores trade items including their name, category, and age; traders including their trade name, and location; and trades including the name of an item a trader traded on some date.

   (a) Write an **English language description** of the following query:

   $$\pi_{name,trader}(\text{TRADE}) \div \pi_{trader}(\sigma_{location=\text{`Chongqing'}}(\text{TRADER}))$$

   [4 marks]

   Solution: 1 mark for each of the following:
   - What is the name of items that
   - have been traded by
   - every trader
   - with location 'Chongqing'?

   (b) Write the following query in relational algebra: What is the name of traders from the location of 'Suzhou' who only order items from the category 'Food'? [3 marks]

   Solution: 1 mark for each of the following:
   - Q_1: Find items that are not from category 'Food'
   - Q_2: Find traders who have traded items not from category 'Food'
   - Q: Find traders from Suzhou and remove any traders who have traded items not from category 'Food'

   $$Q_1 = \text{ITEM} - \sigma_{\text{category=`Food'}}(\text{ITEM})$$
   $$Q_2 = \pi_{\text{trader}}(Q_1 \bowtie \text{TRADE})$$
   $$Q = \pi_{\text{trader}}(\sigma_{\text{location=`Suzhou}}(\text{TRADER})) - Q_2$$

   (c) Write the following query in relational algebra: What is the location of traders who have traded items with the same name on at least two different dates? [3 marks]

   Solution: roughly 1 mark for each of the following:
   - Q_1: Self-join of Trade tables with renamed attribute for date (same trader, same item name)
   - Q_2: Project to traders who have traded the same item on different dates (set difference plus projection)
   - Q: join Q_2 with Trader and project to location of those traders

$$
\begin{aligned}
Q_1 &= \textsc{Trade} \bowtie \delta_{\text{date}\mapsto\text{date'}}(\textsc{Trade})\\
Q_2 &= \pi_{\text{trader}}(Q_1 - \sigma_{\text{date}=\text{date'}}(Q_1))\\
Q &= \pi_{\text{location}}(Q_2 \bowtie \textsc{Trader})
\end{aligned}
$$