Name	CWID

Midterm Exam

October 26th, 2022
10:00-11:15

CS425 - Database Organization
Results

Please leave this empty!

1.1	1.2	1.3	1.4	Sum

Instructions

- Try to answer all the questions using what you have learned in class. Keep hard questions until the end.
- **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!**
- The exam is closed book and closed notes!
- **For relational algebra questions assume set semantics!**

Consider the following database schema and example instance for a mail carrier database:

address

id	street	nr	zip	city	occupant
1	31st	104	60616	Chicago	IIT
2	35th	12	60616	Chicago	IIT
3	60th	5430	66053	Chicago	Miles
4	Lane	12	11111	Milwaukee	Smith

postemp

name	salary	hired
Alice	14000	2007-04-29
Malice	23000	2014-07-05
Bob	8000	2022-01-01

servedby

addr	emp	startdate	enddate
1	Alice	2007-05-01	NULL
2	Alice	2007-05-01	NULL
3	Malice	2014-07-06	2022-01-01
3	Bob	2022-01-02	NULL
4	Malice	2014-07-06	2022-01-01
4	Bob	2022-01-02	NULL

mail

from_addr	to_addr	sent_date	delivered_date
1	4	2020-06-03	2020-06-12
1	4	2022-05-03	2020-05-05
1	3	2020-06-03	2020-07-01
1	2	2022-01-01	2022-01-03

Hints:

- Attributes with black background form the primary key of a relation (.e.g, `name` for relation `postemp`)
- The attributes `from_addr` and `to_addr` of relation `mail` are each foreign keys to relation `address`
- The attribute `emp` of relation `servedby` is a foreign key to relation `postemp`
- The attribute `addr` of relation `servedby` is a foreign key to relation `address`
- All foreign keys have been created with the **CASCADE** option.

Part 1.1 Relational Algebra (Total: 28 Points)

Question 1.1.1 (8 Points)

Write a **relational algebra** expression that returns the *name* and *salary* of all postal employees (relation `postemp`) that were hired before `2021-01-01`.

Solution

$$\pi_{name,salary}(\sigma_{hired<2021-01-01}(\text{ postemp }))$$

Question 1.1.2 (10 Points)

Write a **relational algebra** expression that returns for each mail:

- the name of the postal employee that delivered the mail (the employee that was serving (`servedby`) the delivery address of the mail at the time of delivery
- the city of the address the mail was send
- the city of the address the mail was sent to

Note: You can assume that there exists a scalar function `isnull` which returns `NULL` if its input is `NULL`. For instance, you could use this function in a selection like this: $\sigma_{isnull(somecolumn)}(R)$.

Solution

$$
\begin{aligned}
Q_{fromcity} &= mail \bowtie \rho_{from\_addr \leftarrow id, from\_city \leftarrow city}(\pi_{id,city}(\text{ address })) \\
Q_{fromtocity} &= Q_{fromcity} \bowtie \rho_{to\_addr \leftarrow id, to\_city \leftarrow city}(\pi_{id,city}(\text{ address })) \\
Q_{servedby} &= Q_{fromtocity} \bowtie_{to\_addr = addr \wedge start\_date \leq delivered\_date \wedge (enddate \geq delivered\_date \vee isnull(enddate))} \text{ servedby} \\
Q &= \pi_{name, from\_city, to\_city}(Q_{servedby}))
\end{aligned}
$$

Question 1.1.3 (10 Points)

Write a **relational algebra** expression that returns the name of the postal employee responsible for the longest delivery delay, i.e., who have delivered the mail with the largest difference between `send_date` and `delivered_date`.

Note: Again, you can assume that there exists a scalar function `isnull` which returns `NULL` if its input is `NULL`. For instance, you could use this function in a selection like this: $\sigma_{isnull(somecolumn)}(R)$. Furthermore, if there are multiple mail deliveries with the maximal delay, then return all employees for these deliveries.

Solution

$$\begin{aligned}
Q_{diff} &= \pi_{delivered\_date-send\_date\rightarrow diff,to\_addr\rightarrow a,delivered\_date\rightarrow dd}(\text{ mail })\\
Q_{maxdiff} &=_{max(diff)\rightarrow diff}\ \mathcal{G}(Q_{diff}) \bowtie Q_{diff}\\
Q &= \pi_{name}(\text{ servedby } \bowtie_{a=addr\wedge start\_date\leq dd\wedge(enddate\geq dd\vee isnull(enddate))} Q_{maxdiff})
\end{aligned}$$

Part 1.2 SQL - DDL (Total: 12 Points)

Question 1.2.1 (12 Points)

Write an **SQL DDL statement** that creates a new relation `postoffice`. For each post office we want to record its location consisting of a city, zip code (5 characters), and state (2 character abbreviated state name). Post offices are uniquely identified by their zip code and state. Furthermore, we want to record for each postoffice the name of the postal employee who is the manager of this post office. A manager has to be one of the employees from table `postemp`.

Solution

```
CREATE TABLE postoffice (
    city TEXT,
    zip CHAR(5),
    state CHAR(2),
    manager TEXT REFERENCES postemp,
    PRIMARY KEY (zip, state)
);
```

Part 1.3 SQL - Queries (Total: 40 Points)

Question 1.3.1 (11 Points)

Translate the answer to question 1.1.2 from relational algebra to **SQL**.

Solution

```
SELECT emp AS name, a1.city AS fromcity, a2.city AS tocity
FROM mail m, address a1, address a2, servedby s
WHERE m.from_addr = a1.id
      AND m.to_addr = a2.id
      AND s.addr = m.to_addr
      AND s.startdate <= m.delivered_date
      AND (s.enddate IS NULL OR s.enddate >= m.delivered_date);
```

Of course, using explicit joins is ok too.

Question 1.3.2 (14 Points)

Write an **SQL query** that returns for each postal employee the number of mail items they have delivered so far. An employee delivers a mail item if they are serving the `to_addr` of the mail during the delivery date.

Solution

A fully correct answer requires dealing with employees that have not delivered any mail yet.

```sql
WITH delivered AS (
  SELECT s.emp
  FROM   servedby s, mail m
   WHERE s.addr = m.to_addr
     AND s.startdate <= m.delivered_date
     AND (s.enddate IS NULL OR s.enddate >= m.delivered_date))

SELECT name, count(d.emp) AS totalmail
  FROM postemp p
       LEFT OUTER JOIN
       delivered d ON p.name = d.emp
 GROUP BY name;
```

However, for the sake of the exam it is ok to just return employes that have delivered at least one piece of mail.

```sql
SELECT s.emp, count(*) AS totalmail
  FROM mail m, servedby s
 WHERE s.addr = m.to_addr
       AND s.startdate <= m.delivered_date
       AND (s.enddate IS NULL OR s.enddate >= m.delivered_date)
 GROUP BY s.emp;
```

Question 1.3.3 (15 Points)

Write an **SQL query** that returns the names of postal employees who have not delivered any mail yet.

Solution

```
SELECT name
FROM postemp
WHERE name NOT IN (SELECT emp
                     FROM mail m, servedby s
                    WHERE s.addr = m.to_addr
                      AND s.startdate <= m.delivered_date
                      AND (s.enddate IS NULL OR s.enddate >= m.delivered_date));
```

alternatives include aggregation (count is 0, which requires that employees that have not delivered mail will be returned) and set difference

Part 1.4 SQL - Updates (Total: 20 Points)

Question 1.4.1 (7 Points)

Write an **SQL statement** that updates all assignments (`servedby`) of postal employees to addressed by setting the `enddate` of the assignment to `2022-10-26` if it is currently `NULL`. Otherwise, set the enddate to `2022-11-26`.

Solution

```
UPDATE servedby
   SET enddate = CASE WHEN enddate IS NULL
                      THEN '2022-10-26'::DATE
                      ELSE '2022-11-26'::DATE
                 END;
```

Question 1.4.2 (13 Points)

Write an **SQL statement** that deletes all postal employees which are currently (as of date `2022-01-01`) not serving any addresses.

Solution

```
DELETE FROM postemp p
 WHERE name NOT IN (SELECT s.emp
                      FROM servedby s
                     WHERE s.startdate <= '2022-01-01'::DATE
                       AND (s.enddate IS NULL OR s.enddate => '2022-01-01'::DATE));
```