

Name

CWID

Midterm Exam

October 22nd, 2013

10:00-11:30

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!

Consider the following database schema and example instance for a flight information system:

airline

Icode	IName	country
AE	American Airlines	USA
DAL	Delta Airlines	USA
AC	Air Canada	Canada
DHL	Lufthanse	Germany

flight

Icode	flightNumber	fromAirport	toAirport	lengthHours
AE	367	JFK	DEL	8
AC	555	ORD	YYZ	2
AC	73	YYZ	DEL	16

airport

aCode	aName	country	city
JFK	John F Kennedy International	USA	New York
DEL	Indira Gandhi International Airport	India	Delhi
ORD	Chicago O'Hare International	USA	Chicago
YYZ	Lester B. Pearson International	Canada	Toronto

schedule

Icode	flightNumber	date	time	capacity
AE	367	2013-05-04	12:40	245
AC	555	2013-07-07	02:00	80
AC	73	2013-08-08	04:00	100
AC	73	2013-08-09	04:00	100

booked

customer	Icode	flightNumber	date	time
Peter Fullbright	AE	367	2013-05-04	11:00
Alice Bobsen	AC	555	2013-07-07	02:00

Hints:

- Attributes with grey background form the primary key of a relation (.e.g, *Icode* for relation *airline*)
- The attribute *fromAirport* of relation *flight* is a foreign key to *aCode* of relation *airport*. The attribute *toAirport* of relation *flight* is a foreign key to *aCode* of relation *airport*. The attribute *Icode* of relation *flight* is a foreign key to *Icode* in relation *airline*.
- The attributes *Icode* and *flightNumber* of relation *schedule* form a foreign key to *Icode* and *flightNumber* in relation *flight*.
- The attributes *Icode*, *flightNumber*, *date*, *time* of relation *booked* form a foreign key to relation *schedule*.
- All foreign keys have been created with the **CASCADE** option.

Part 1.1 Relational Algebra (Total: 30 Points)

Question 1.1.1 (7 Points)

Write a **relational algebra** expression that returns the names of all airlines from Germany.

Solution

$$\pi_{lName}(\sigma_{country='Germany'}(\text{airline}))$$

Question 1.1.2 (7 Points)

Write a **relational algebra** expression that returns the total capacity of all scheduled flights per airline. Return pairs of airline codes (lCode) and total capacity. For example, if American Airlines flight 555 is scheduled for two times, both with capacity 100, then this would be counted as 200 when calculating the total capacity for American Airlines.

Solution

$$lCode \mathcal{G}_{sum(capacity)}(\text{schedule})$$

Question 1.1.3 (7 Points)

Write a **relational algebra** expression that returns cities from the USA that have an airport.

Solution

$$\pi_{city}(\sigma_{country='USA'}(\text{airport}))$$

Question 1.1.4 (8 Points)

Write a **relational algebra** expression that returns the names of all customers together with the destination city of flights they have booked.

Solution

$$\begin{aligned} \text{bookCity} &\leftarrow \text{booked} \bowtie \text{flight} \bowtie_{toAirport=aCode} \text{airport} \\ q &\leftarrow \pi_{customer,city}(\text{bookCity}) \end{aligned}$$

Part 1.2 SQL - DDL (Total: 16 Points)

Question 1.2.1 (8 Points)

Write an **SQL statement** that creates a new relation *flightdelay* that records delay times (*delay* in minutes) for scheduled flights. For example, we would like to be able to store that flight AC 555 scheduled for 2013-07-07 at 02:00 is delayed by 15 minutes.

Solution

```
CREATE TABLE flightdelay (  
    lCode VARCHAR(3),  
    flightNumber INT,  
    date DATE,  
    time TIME,  
    delay INT,  
    PRIMARY KEY (lCode, flightNumber, date, time),  
    FOREIGN KEY (lCode, flightNumber, date, time) REFERENCES schedule  
);
```

Question 1.2.2 (8 Points)

Write a relation *securityAlert*. Each alert is for a fixed airport. An alert consist of a message, a severity level (one of 'RED', 'ORANGE', 'GREEN'), and a time when the alert was issued. Alerts are identified by their time and the airport they have been issued for.

Solution

```
CREATE TABLE securityAlert (  
    aCode VARCHAR(3) FOREIGN KEY REFERENCES airport,  
    message VARCHAR(4000),  
    severity VARCHAR(6),  
    issueTime TIMESTAMP,  
    PRIMARY KEY (aCode, issueTime),  
    CHECK (severity IN ('RED', 'ORANGE', 'GREEN'))  
);
```

Part 1.3 SQL - Queries (Total: 39 Points)

Question 1.3.1 (7 Points)

Write an **SQL query** that returns airline code (*lCode*) of flights that are scheduled between 10:00 and 11:00.

Solution

```
SELECT lCode
FROM schedule
WHERE time BETWEEN '10:00' AND '11:00';
```

We did not ask for it, but **DISTINCT** would be ok.

Question 1.3.2 (7 Points)

Write an **SQL query** that returns the airline name (*lName*) and flight number for flights that are less than 6 hours long.

Solution

```
SELECT lName, flightNumber
FROM airline NATURAL JOIN flight
WHERE lengthHours < 6;
```

Question 1.3.3 (9 Points)

Write an **SQL query** that returns flights (airline code and flight number) that start from an airport in USA and land in an airport in India.

Solution

```
SELECT lCode, flightNumber
FROM airport s, flight f, airport d
WHERE s.country = 'USA' AND f.fromAirport = s.aCode
      AND f.toAirport = d.aCode AND d.country = 'India';
```

Question 1.3.4 (8 Points)

Write an **SQL query** that returns for each time and date, the number of flights starting at that date and their maximum capacity. For example, if there are two flights that start at July, 7th 2013 at 10:00, one with capacity 60 and one with capacity 200, then one result tuple of the query would be (2,200,2013-07-07,10:00).

Solution

```
SELECT count(*) AS numFlights, max(capacity) AS maxC, date, time
FROM schedule
GROUP BY date, time
```

Question 1.3.5 (9 Points)

Write an **SQL query** that returns flights that are over-booked. A flight is considered over-booked if the number of bookings for this flight exceeds the capacity of this flight.

Solution

```
SELECT *
FROM schedule s
WHERE capacity < (SELECT count(*)
                  FROM booked b
                  WHERE s.lCode = b.lCode AND s.flightNumber = b.flightNumber
                  AND s.date = b.date AND s.time = b.time);
```

Or

```
SELECT lCode, flightNumber, date, time, capacity
FROM schedule
     NATURAL JOIN
     booked
GROUP BY lCode, flightNumber, date, time, capacity
HAVING capacity < count(*);
```


Part 1.4 SQL - Updates (Total: 15 Points)

Question 1.4.1 (7 Points)

Write an **SQL statement** that deletes all flights from JFK to DEL. Recall that all foreign key constraints were created with the **CASCADE** option.

Solution

```
DELETE FROM flight
WHERE fromAirport = 'JFK' AND toAirport = 'DEL';
```

Question 1.4.2 (8 Points)

Write an **SQL statement** that reduces the capacity of flights to 200 if they are scheduled for a time in the morning (before 12:00 midday) and currently have a capacity above 200.

Solution

```
UPDATE schedule SET capacity = 200
WHERE capacity > 200 AND time < 12:01;
```