Databases (TDA357, DIT620)

Exercise 3 Theory topics: relations and dependencies 14/2 Wed

1. Functional dependencies and normal forms (Lecture 6).

Exercise 1

First we will look at the flights table, that you saw in the previous exercise session. As you already know the table, most of you will not need to study it again. Nevertheless, I attach its whole description.

flight code	airline	prime flight	operating airline	departure city	departure airport	destination city	destination airport	aircraft type	seats
SK111	SAS	SK111	SAS	Gothenburg	GOT	Frankfurt	FRA	B737	140
LH555	Lufthansa	SK111	SAS	Gothenburg	GOT	Frankfurt	FRA	B737	140
AF111	Air France	AF111	Air France	Gothenburg	GOT	Paris	CDG	A320	170
LH111	Lufthansa	LH111	Lufthansa	Frankfurt	FRA	Paris	CDG	A321	200
LH222	Lufthansa	LH222	Lufthansa	Frankfurt	FRA	Malta	MLA	A320	170
AF222	Air France	AF222	Air France	Paris	ORY	Malta	MLA	A320	170
AB222	Air Berlin	AB222	Air Berlin	Frankfurt	FRA	Munich	MUC	A320	170
KM111	Air Malta	KM111	Air Malta	Munich	MUC	Malta	MLA	A319	140
LH333	Lufthansa	KM111	Air Malta	Munich	MUC	Malta	MLA	A319	140
SK222	SAS	KM111	Air Malta	Munich	MUC	Malta	MLA	A319	140
AF333	Air France	AF333	Air France	Paris	CDG	Frankfurt	FRA	A320	170

Table 1 Flights table

We assume the following (slightly simplified) conventions for this domain:

- the “flight code” attribute determines all other attributes on a row,
- the “prime flight” is the flight code used by the airline operating the flight; the “flight code” in
- the first column can thus belong to another airline that has a code sharing agreement with the
- operating airline,
- the “prime flight” appears in the table as a “flight code” as well, having itself as prime flight
- each airport has a unique code
- every aircraft of the same type has the same number of seats

(It is a common practice that one and the same flight can be booked using different airlines. Each airline uses a different “flight code”, but the passengers end up in the same plane. The code used by the actual operating airline is called the “prime flight” code. For example, whether you book flight LH333 with Lufthansa or flight SK222 with SAS, you end up in the plane of Air Malta flight KM111.)

<u>Question 1a.</u> Identify the functional dependencies and keys in the domain as described above. You must have some functional dependencies that are not superkeys. Consider the entire Table 1 as one relation. For functional dependencies, it is enough to list a base (a minimal set that implies all the others). (3p.)

<u>Question 1b.</u> Starting with Table 1 and the functional dependencies and keys in (2a), decompose the relation into BCNF (Boyce-Codd Normal Form). Show all intermediate steps. Notice: if you find out that the relation is already in BCNF, then you have done something wrong in (2a). (4p.)

Question 1c. Suppose you know the attributes of a relation and that it has no functional dependencies. Do you have enough information to bring it to BCNF. If yes, how? If no, why? Do you have enough information to bring it to the Fourth Normal Form (4NF). If yes, how? If no, why? (2p.)

Exercise 2

Suppose we have relation $R(A, B, C, D, E, F, G)$ and functional dependencies

- $AC \rightarrow E$
- $EG \rightarrow D$
- $AC \rightarrow B$
- $DF \rightarrow A$

Question 2a. This relation R has three keys. Indicate which ones with some simple reasoning. Moreover, tell why the other two are not keys of R. (4p)

- $\{A, C, E, F\}$
- $\{A, C, F, G\}$
- $\{C, D, F, G\}$
- $\{C, E, F, G\}$
- $\{B, C, E, F, G\}$

Question 2b. Decompose this relation R to Boyce-Codd normal form. Remember to show each step in the normalisation process, and at each step write down which functional dependency is being used. (4p)

Question 2c. Indicate and justify which functional dependency (-ies) of R violate Third Normal Form (2p).

Question 2d. Decompose relation R to Third Normal Form (3p).

Exercise 3

Question 3a. Give an example of a relation that is in BCNF (BoyceCodd Normal Form) but not in 4NF (Fourth Normal Form). Show all the information that is needed: attributes, dependencies, keys, etc, clearly stating what the 4NF violations are, as well as an instance (a set of tuples). (2p)

Question 3b. Transform your relation in (2a) to 4NF. (2p)

2. Relational Algebra and query compilation (Lecture 7)

Exercise 4

Given the relation:

Country(continent, country_name, capital, cities_above_milion, area, population, sea_access)

Question 4a Write a relational algebra query that returns, for each continent with more than 30 countries, the total combined cities with population above one million with an access to the sea (ocean). The query should return tuples of the form(continent, totalCities) (3p)

Exercise 5

Now, going back to flights table from Exercise 1. Now, let's assume a the following schema.

Airports(code ,city)

FlightCodes(code , airlineName)

Flights(departureAirport, destinationAirport, departureTime, arrivalTime, code)
departureAirport → Airports.code
destinationAirport → Airports.code
code → FlightCodes.code

Question 5a : Below you can find an SQL query that finds all airports that have departures or arrivals (or both) of flights operated by Lufthansa or SAS (or both), based on the second schema in the description of exercise 5. Express this query by a relational algebra expression. (5p)

```sql

SELECT DISTINCT served
FROM (( SELECT destinationAirport AS served,airlineName
        FROM FlightCodes
        JOIN Flights
        ON Flights.code = FlightCodes.code)
UNION
     (  SELECT departureAirport AS served,airlineName
        FROM FlightCodes
        JOIN Flights
        ON Flights.code = FlightCodes.code)) AS D
WHERE D.airlineName = 'Lufthansa' OR D.airlineName = 'SAS' ;
```

Question 5b Translate the following relational algebra expression to an SQL query: (3p)

$$\pi_{First.depatureTime, Second.arrivalTime}((\rho_{First}(Flights)) \bowtie_{First.destinationAirport = Second.departureAirport} (\rho_{Second}(Flights)))$$