CptS 415 Big Data: Assignment 1
Md Muhtasim Billah

Question 1.

Answer:

Where there are several big data applications that are available, the one that I decided to go with is Facebook which is a social networking website.

The Five V's of Facebook's Big Data:

There are several aspects to be satisfied for a system to be considered as a big data system which are called the five V's. These five V's are explained below in terms of Facebook's database.

I. Volume: 500TB of new data per day are ingested in Facebook databases which satisfies as the huge volume of incoming data.
II. Velocity: Facebook has now over 2 billion users as of 2020 who are connected on multiple levels and interacting with each other 24/7 (there are a million logins every minute). Thus, data is being generated at a very high rate every second of every day.
III. Variety: An average Facebook user posts status updates, pictures, audios, videos, comments and reactions etc. which are collected as various types of data that include structured, semi-structured and unstructured data.
IV. Veracity: The collected data is required to be as accurate as possible for a big data system. The user information which are collected and then cleaned at Facebook which include millions of features that are later used for targeted marketing.
V. Value: The data, if not worthy of putting to any use, is not very desirable. Data has to have some value that can be further utilized to develop a business model or user behavior prediction. For example, based on the users' likings and interests, Facebook can recommend similar items to the users.

If I am required to design a database system for Facebook, I think several data models would be necessary including the relational model, key-value store and XML among others to represent the data with a focus mostly on the key-value store and the XML. Because Facebook generates way more semi-structured (for example, XML) and unstructured data (for example text, images, audio, video) for which non-relational models will be more useful.

Question 2.

Answer:

2(a).

Relation Schema: It is the description of the relation which includes the name of relation and its attributes. As an example, the schema (partial) for the relation Airport is provided below.

Airport ID	Name	City	Country	IATA	ICAO	Latitude	Longitude	Altitude

Relational Database schema: It's a collection of all the relation schemas of a dataset. For the Airports database, the example database schema is as below.

Airport

Airport ID	Name	City	Country	IATA	ICAO	Latitude	Longitude	Altitude

Airline

Airline ID	Name	Alias	IATA	ICAO	Callsign	Country	Active

Route

Airline	Airline ID	Source airport	Source airport ID	Destination airport	Destination airport ID	Codeshare	Stops	Equipment

Domain: The set of values permitted for a specific attribute. For example, the Airport ID attribute is represented by an integer value and cannot be NULL.

Attribute: The columns in a relation are called attributes. For the Airline dataset, some of the attributes are Airline ID, Name, Alias etc.

Attribute domain: Attribute Domain refers to the predefined values of attributes in a table. For example, Airline ID is not allowed to be NULL.

A few instances of the table are as below.

507,"London Heathrow Airport","London","United Kingdom","LHR","EGLL",51.4706,-0.461941,83,0,"E","Europe/London","airport","OurAirports"

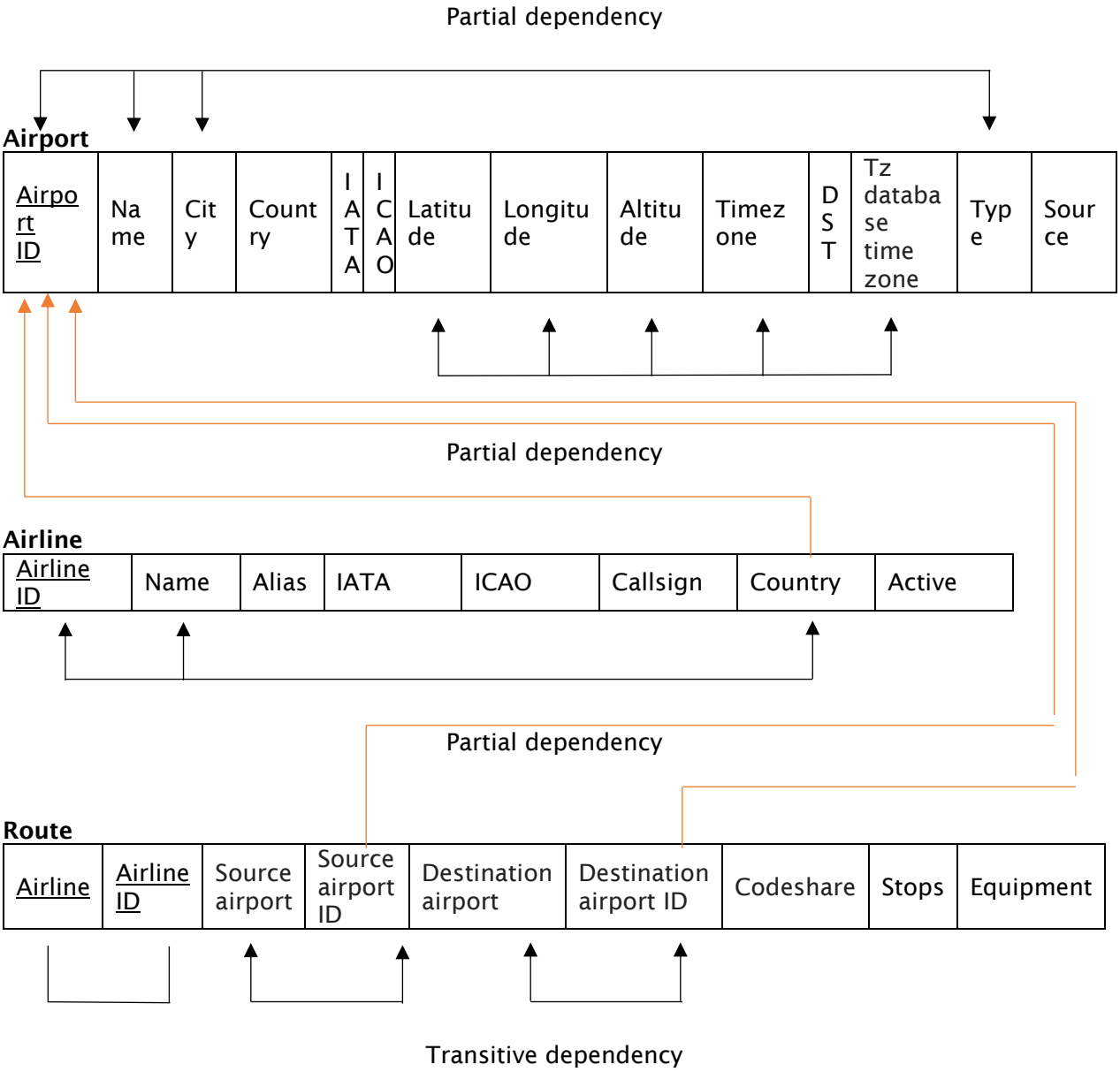
26,"Kugaaruk Airport","Pelly Bay","Canada","YBB","CYBB",68.534401,-89.808098,56,-7,"A","America/Edmonton","airport","OurAirports"

3127,"Pokhara Airport","Pokhara","Nepal","PKR","VNPK",28.200899124145508,83.98210144042969,2712,5.75,"N","Asia/Katmandu","airport","OurAirports"

8810,"Hamburg Hbf","Hamburg","Germany","ZMB",\N,53.552776,10.006683,30,1,"E","Europe/Berlin","station","User"

2(b).

There are three databases in the OpenFlight dataset: Airport, Airline, and Route. The schema of these three databases and mark the primary keys (underlined), foreign keys are given below. Examples of possible functional dependencies are also identified. The partial and transitive dependencies are shown in black color where the foreign key and primary key relations are given by the orange color.



2(d).

R (A1, A2, A3, A4) is the 3NF form for the given schema.

Explanation:

For a 3F formulation, it is known that there is no transitive dependency for the non-key attributes and on top of that, it holds the condition of 2NF and 1NF. In the provided functional dependencies, there is no visible transitive dependencies (if A -> B and B -> C then A -> C). Also, they hold the condition of 1NF (single value and unique record) and 2NF (1NF and a primary key).

Question 3.

Answer:

Q1. Which theaters feature “Zootopia”?

$$\left(\pi_{Theaters} = \left(\sigma(Movies \times Location \times Schedule)\right) \begin{array}{c} \text{Movies. Title} = \text{Schedule. Title and} \\ \text{Location. Theater} = \text{Schedule. Theater} \\ \text{And Movies. Title} = \text{“Zootopia”} \end{array}\right)$$

Q2. List the names and address of theaters featuring a film directed by Steven Spielberg.

$$\left(\pi_{Theater,Address} = (\sigma(Movies \times Location \times Schedule)\ ) \begin{array}{c} \text{Movies. Title} = \text{Schedule. Title and} \\ \text{Schedule. Theater} = \text{Location. Theater} \\ And\ Movies.Director = "Steven\ Spielberg" \end{array}\right)$$

Q3. What are the address and phone number of the Le Champo theater?

$$\left(\pi_{\text{Address,Phone Number}} = (\sigma(\text{Location})\ ) \begin{array}{c} \text{Theater} = \text{"Le Champo Theater"} \end{array}\right)$$

Q4. List pairs of actors that acted in the same movie.

Let, $M = \rho_M(\text{Movies})$

$$\left(\pi_{\text{M.Actor,Movies.Actor}} \left(\begin{array}{c} (\text{M} \times \text{Movies}) \\ \text{M. Title} = \text{Movies. Title} \end{array}\right)\right)$$

Question 4.

Answer:

4(a). Block Nested Loop Join

It's an improved version of the nested loop join in which every block of inner relation is paired with every block of outer relation.
Let,

R = Outer relation
S = Inner relation
B_R = Outer relation total blocks
B_s = Inner relation total blocks
t_R = Number of tuples in R
t_S = Number of tuples in S

Algorithm:

```
for each block B_R in R
        for each block B_s in S
                for each tuple t_R in B_R
                        for each tuple t_S in B_S
                                test if pair (t_R, t_S) satisfies the join condition θ
                        end for
                end for
        end for
end for
```

$$Cost = (B_R + B_R \times B_S) = 10 + 100 = 110$$

4 (b). Sort-Merge join

The core idea of the sort-merge algorithm is to first sort the relations by the join attribute, so that intermediate linear scans will operate on these blocks at the same time.

Algorithm:

It consists of two stages:

I. Sort tuples in R & S by join key.
   a. All tuples with same key in consecutive order.
   b. Input might already be stored.
II. Join pass: Merge a scan the sorted partitions and emit tuples that match

$$Cost = B_R + B_S = 10 + 10 = 20$$

4 (c). Hash-Join

Applicable for both natural and equi joins. This join algorithm has a hash function "h" which is utilized to partition the records of the relations. The basic idea is to divide (partition) the record of each of the tables into sets that have the same hash values in the join attribute.

$$Cost = (B_R + B_S) + (B_R + B_S) + (B_R + B_S)$$
$$= 3(B_R + B_S) = 3(10 + 10) = 60$$