

Name: Gurdev singh

Id:100376466

Lab-3

Question:1:

1. Using the Airport KLX Table, describe an example that illustrates the insertion anomaly.

Ans: In the given table if we only need to add a new Terminal and number of gates like this row:

Terminal Id	Number of Gates	Departing Flights
D	30	0

We cannot add this row directly without adding, a new Airline Id and Airline name because All Airline id is Unique. So, we must add a new Airline Name and, although that Airline may not exist.

Airline Id	Airline Name
------------	--------------

This will create an insertion anomaly.

2. Using the AIRPORT KLX Table, describe an example that illustrates the deletion anomaly

Ans:

So, if we want to delete only one terminal Id like A and Number of Gates like this:

Terminal Id	Number Of Gates
A	20

It will also remove all the other Airline Id and Airline Name although they exist, and we do not want to delete:

Airline Id	Airline Name	Terminal
UA	United	A
NW	Northwest	A
AA	American	A

This shows that deleting one part from any column will create a delete anomaly

3. Using the AIRPORT KLX Table, describe an example that illustrates the modification anomaly.

Ans: In the table, if we try to modify the Terminal Id: A, we must modify it from all the rows one by one and it will take a lot of time as there are multiple rows of A Terminal Id.

Airline Name	Terminal Id
United	A
Northwest	A
American	A
United	A

So, modifying each value from each row is very hard if the data is too long. So it will create a modification anomaly.

4. In the AIRPORT KLX Table, identify [6 Marks]

- a) Full (key) Functional Dependencies:
Date, Airline Id \longrightarrow Number of departing flights
- b) Partial (key) Functional Dependencies (if any)
Airline id, Terminal Id \longrightarrow Number of gates
- c) Transitive Functional Dependencies (if any).
Terminal Id \longrightarrow Number of gates

5. Convert into BCNF , Given Table :

<u>Date</u>	<u>Airline ID</u>	Airline Name	Terminal Id	Number of gates	Number of departing flights
11 DEC	UA	United	A	20	34
11 DEC	NW	Northwest	A	20	17
11 DEC	AA	American	A	20	11
11 DEC	DL	Delta	B	15	20
11 DEC	JB	Jet Blue	B	15	6
12 DEC	UA	United	A	20	29
12 DEC	DL	Delta	B	15	20
12 DEC	SWA	Southwest	C	15	17

In this table, there are **some partial dependencies** exist like

Airline id, Terminal \longrightarrow Number of gates

Airline Id \longrightarrow Airline Name etc.

So convert into 2NF.

CONVERT INTO 2NF

<u>Date</u>	<u>Airline ID</u>	Number of departing flights
11 DEC	UA	34
11 DEC	NW	17
11 DEC	AA	11
11 DEC	DL	20
11 DEC	JB	6
12 DEC	UA	29
12 DEC	DL	20
12 DEC	SWA	17

<u>Airline ID</u>	Airline Name	<u>Terminal Id</u>	Number of gates
UA	United	A	20
NW	Northwest	A	20
AA	American	A	20
DL	Delta	B	15
JB	Jet Blue	B	15
UA	United	A	20
DL	Delta	B	15
SWA	Southwest	C	15

CONVERT INTO 3NF OR BCNF

In this Step, A table does not contain transitive functional dependencies

<u>Date</u>	<u>Airline ID</u>	Number of departing flights
11 DEC	UA	34
11 DEC	NW	17
11 DEC	AA	11
11 DEC	DL	20
11 DEC	JB	6
12 DEC	UA	29
12 DEC	DL	20
12 DEC	SWA	17

<u>Airline ID</u>	Airline Name	<u>Terminal Id</u>
UA	United	A
NW	Northwest	A
AA	American	A
DL	Delta	B
JB	Jet Blue	B
UA	United	A
SWA	Southwest	C

<u>Terminal Id</u>	Number of gates
A	20
B	15
C	15

Ques 2. Finding Keys and Normalization

Consider the following relation scheme and FDs:

Employee (EmpID, EmpName, Specialization, ManagerID, ProjID, ProjTitle, SupervisorName, SupervisorLocation, Bonus) The table is :

Emp Id	Emp Name	Specialization	Manger Id	Proj Id	Proj Title	Supervisor Name	Supervisor Location	Bonus
--------	----------	----------------	-----------	---------	------------	-----------------	---------------------	-------

1) Find all keys (candidate keys) [5 Marks]:

Ans: So, the Candidate Keys are:

1. Empld, Specialiaztion, Projld,
2. Empld, Mangerld, Projectld.

2) Normalize this relation up to BCNF (explain all steps of your normalization, mention functional dependencies to justify the normalization process):

Covert into 2NF:

All the partial dependencies are separated:

<u>EmpId</u>	<u>Specialization</u>	<u>ProjId</u>	Bonus
--------------	-----------------------	---------------	-------

<u>EmpId</u>	EmpName
--------------	---------

<u>ProjId</u>	ProjTitle	SupervisorName	SupervisorLocation
---------------	-----------	----------------	--------------------

<u>EmpId</u>	<u>Specialization</u>	ManagerId
--------------	-----------------------	-----------

Covert into 3NF Or BCNF:

All the transitive dependencies are separated like:

SupervisorName to SupervisorLocation, MangerId to specialization.

<u>EmpId</u>	<u>Specialization</u>	<u>ProjId</u>	Bonus

<u>EmpId</u>	EmpName
--------------	---------

<u>ProjId</u>	ProjTitle	SupervisorName
---------------	-----------	----------------

<u>SupervisorName</u>	SupervisorLoaction
-----------------------	--------------------

<u>EmpId</u>	<u>Specialization</u>	ManagerId
--------------	-----------------------	-----------

<u>MangerId</u>	Specialization
-----------------	----------------