

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA

Shri Vaishnav Institute of Information and Technology

Department of Computer Science Engineering



Session 2019-20

BTCSI 405

Database Management System

MST II Assignment

Submitted to:

Prof. Archana Choubey

Submitted by:

Mahiransh Sharma

Enrollment:

18100BTCMCI02957

1. Given the following tables:**runners**

ID	NAME
1	John
2	Jane
3	Alice
4	Bobby

races

ID	EVENT	WINNER_ID
1	100 meter dash	2
2	500 meter dash	3
3	cross-country	2
4	triathlon	Null

What will be the result of the query below?

SELECT * FROM runners WHERE id NOT IN (SELECT winner_id FROM races);

Explain your answer and also provide an alternative version of this query that will avoid the issue that it exposes.

The result of given query is nothing may be an empty set.

The result is empty because NULL entry cannot be compared with the conditional operations (for instance, =, >=, <=, !=, etc). In the above query the runners table id is not comparable with all winner_id of races table. If you want to compare any element of table with NULL values, then you must use "IS NULL" or for not null values "IS NOT NULL".

To avoid this issue, we can apply another condition in which our query or said to be subquery select to be not NULL.

Alternative of given query as follows:

SELECT * FROM runners WHERE id NOT IN (SELECT winner_id FROM races WHERE winner_id IS NOT NULL);

Desired output would be:

id	name
1	John
4	Bobby

2. Let $R = (A, B, C, D, E, F)$ be a relation scheme with the following dependencies: $C \rightarrow F, E \rightarrow A, EC \rightarrow D, A \rightarrow B$.
Give a primary key of the relation $R (A, B, C, D, E, F)$. Prove your answer.

Given dependencies are F.D. = $\{C \rightarrow F, E \rightarrow A, EC \rightarrow D, A \rightarrow B\}$

First, we write the right-hand side attribute of relations: FADB,
Then equate it to form a candidate key;

CK : $EC = FADB$

Now, Check the closure of EC,

$EC^+ = EC$ (because EC can determine EC),

$EC^+ = ECF$ (because F can be determined by C),

$EC^+ = ECFA$ (because A can be determined by E),

$EC^+ = ECFAD$ (because D can be determined by EC),

$EC^+ = ECFADB$ (here, because B can be determined by A)

C.K. = $\{EC, \dots\}$

Let's check if any other candidate key is available in the table. So, either E or C is present at right hand side of relation.

F.D. = $\{C \rightarrow F, E \rightarrow A, EC \rightarrow D, A \rightarrow B\}$

Here, at right hand side, not availability of E and C. It means the table contain only one candidate / primary key i.e. EC.

C.K. = $\{EC\}$, i.e. primary key.

3. Describe and illustrate the process of normalizing the table shown in Figure to 3NF. State any assumptions you make about the data shown in this table.

Let's assume that a patient is registered at only one surgery and he/she may have more than one appointment on a given day. All the schedules have been fixed for the whole days and week.

staffNo	dentistName	patientNo	patientName	Appointment Date Time		surgeryNo
S1011	Tony Smith	P100	Gillian White	12-Aug-03	10:00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-03	12:00	S15
S1024	Helen Pearson	P108	Ian Mackey	12-Sep-03	10:00	S10
S1024	Helen Pearson	P108	Ian Mackey	14-Sep-03	10:00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-03	16:30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-03	18:00	S13

To convert this table into 1NF we have to remove two different attribute, time and date from the table.

staffNo	dentistName	patientNo	patientName	apDate	apTime	surgeryNo
S1011	Tony Smith	P100	Gillian White	12-Aug-03	10:00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-03	12:00	S15
S1024	Helen Pearson	P108	Ian Mackey	12-Sep-03	10:00	S10
S1024	Helen Pearson	P108	Ian Mackey	14-Sep-03	10:00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-03	16:30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-03	18:00	S13

Now, we have to remove partial dependency and spit all the columns of tables into more than one table to convert into 2NF. In below tables, underline are candidate/primary key.

Separate staffNo, apDate, apTime, patientNo, patientName.

<u>staffNo</u>	<u>apDate</u>	<u>apTime</u>	patientNo	patientName
S1011	12-Aug-03	10:00	P100	Gillian White
S1011	13-Aug-03	12:00	P105	Jill Bell
S1024	12-Sep-03	10:00	P108	Ian Mackey
S1024	14-Sep-03	10:00	P108	Ian Mackey
S1032	14-Oct-03	16:30	P105	Jill Bell
S1032	15-Oct-03	18:00	P110	John Walker

Make separate table of staffNo, apdate, surgeryNo.

<u>staffNo</u>	<u>apDate</u>	surgeryNo
S1011	12-Aug-03	S10
S1011	13-Aug-03	S15
S1024	12-Sep-03	S10
S1024	14-Sep-03	S10
S1032	14-Oct-03	S15
S1032	15-Oct-03	S13

Make staffNo, dentistName and primary key is staffNo.

<u>staffNo</u>	dentistName
S1011	Tony Smith
S1024	Helen Pearson
S1032	Robin Plevin

Now, to convert our tables into 3NF, we have to remove transitive dependencies which is if $a \rightarrow b$, $b \rightarrow c$ so we have $a \rightarrow c$. In below tables, underline is primary key.

Our first table is Dentist(staffNo, dentistName)

<u>staffNo</u>	dentistName
S1011	Tony Smith
S1024	Helen Pearson
S1032	Robin Plevin

Our second table is Surgery(staffNo, apDate, surgeryNo)

FK		
<u>staffNo</u>	<u>apDate</u>	surgeryNo
S1011	12-Aug-03	S10
S1011	13-Aug-03	S15
S1024	12-Sep-03	S10
S1024	14-Sep-03	S10
S1032	14-Oct-03	S15
S1032	15-Oct-03	S15

Now, our third table is Patient(patientNo, patientName)

<u>patientNo</u>	patientName
P100	Gillian White
P105	Jill Bell
P108	Ian Mackey
P110	John Walker

For the appointments, our table is Appointment(staffNo, apDate, apTime, patientNo)

FK			
<u>staffNo</u>	<u>apDate</u>	<u>apTime</u>	patientNo
S1011	12-Aug-03	10:00	P100
S1011	13-Aug-03	12:00	P105
S1024	12-Sep-03	10:00	P108
S1024	14-Sep-03	10:00	P108
S1032	14-Oct-03	16:30	P105
S1032	15-Oct-03	18:00	P110

The above tables are finally converted into 3NF.

4. How you know when to use IMPORT vs LOAD utility?

When we need to view a target table, and the target table has constraints, we don't want the target table to be put in the Set Integrity Pending state. If the above situations are occurred then we only face towards the IMPORT utility. IMPORT utility is mainly use to insert data from an external file into a table, hierarchy, view, or nickname.

On the other hand. i.e. LOAD utility, where we have some concern in the performance of primary, if you want to writes formatted pages directly into the database so you may use LOAD utility. LOAD utility also allows you the option to not log the data or use the COPY option to save a copy of the loaded data. Load operations can fully exploit resources, such as CPUs and memory on SMP and MPP environments.

Method	LOAD utility	IMPORT utility
Purpose	To efficiently move large quantities of data into newly created tables, or into tables that already contain data.	To insert data from an external file into a table, hierarchy, view, or nickname.
Best practice usage	This utility is best suited to situations where performance is your primary concern. This utility can be used as an alternative to the import utility. It is faster than the import utility because it writes formatted pages directly into the database rather than using SQL INSERTS. In addition, the load utility allows you the option to not log the data or use the COPY option to save a copy of the loaded data. Load operations can fully exploit resources, such as CPUs and memory on SMP and MPP environments.	The import utility can be a good alternative to the load utility in the following situations: <ul style="list-style-type: none"> • where the target table is a view • the target table has constraints and you don't want the target table to be put in the Set Integrity Pending state • the target table has triggers and you want them fired.
