29/08/22

NORMALISATION

Domain is called (acceptable by attribute)

= Normalisation - process of analysing and correcting the table structure so as do minimise data redundancy (complication) and minimise the insertion/deletion/updation anomalies.

Proj Num	Proj. Name	Emp num	Emp. name	Job class	Charge Hourly	Hours worked	Total charge
15	Research	10.5	Aman	Elec Engg	10		
?	—	10.7	Abha	DSS Analyst	15		
—	—	11.0	Raman	DB Designer	17		
—	—		Rohit	EE			
16	Development	106	~~Etc E~~ Bob	Elec E	18		
—	—	104	~~Data Sc~~ Rob	Data Sc A	20		
—	—	103	~~DBA~~ Dylan	DBA	100		
—	—	115	~~Spec Analyst~~ Chain	Sync Analyst	1500		
17	Manufactur	113	Amit		160		
—	—	117	Neeraj		115		

A → B

⇒ (1) Functional Dependencies

The attribute 'B' is fully functional dependent on attribute 'A' if each value of 'A' determines one and only one value of B. eg:-

Proj_Num → Proj - Name

~~Project~~ Proj - Num functionally determines Proj - Name which means for every project no there is a Proj name associated with it

⇒ (2) Fully functional Dependencies

If attribute 'B' is functionally dependent on a composite key A (composite - multiple attribute) but not on any subset of that composite key then the attribute B

Page No.
Date

is fully functionally dependent on A. eg.
proj-num, emp-num → Job class

(3) Partial Dependencies
An attribute B is functionally dependent on an attribute X which is a subset of the primary key A

(*) Primary key of table is Proj-Num, Emp-Num
* But Proj-Name is dependent on Proj-Num which is a subset of primary key

(4) Transitive Dependency:-
Transitive dependency is the dependency of one non-prime attribute on another non-prime attribute

- Prime attribute of a relation are is the attribute which is the member of the candidate key of R

- Non-prime attribute of a relation are those attributes which are not a member of any candidate key of R.

6/04/19

⇒ Objectives

1) Each table represent a single subject for example a course table will contain the data that is directly related to the course
2) No data item unnecessary stored in more than one table this ensures that the data is updated at one place only.
3) All the non-prime attribute in a table are dependent on the primary key this ensures that

DELTA

Page No.
Date

data is uniquely identifiable by the primary key.
4) Each table is free from insertion, updation, and deletion anomalies.

⇒ Conversion to First Normal Form (1NF)

Step-1 - Eliminate the repeating group.
Each cell in a table has a single value and there are no repeating groups.
cell contains appropriate.

Step-2 - Identify the primary key.
Eg. Proj_Num, Emp_Num

Proj_Num	Proj_Name	Emp_Numb	Emp_Name	Job_class	Chg_Hour	Hours_worked
15	Manufac	101		Manager	500	
		102		Clerk	250	
		105		Sr. Manager	600	
		110		PA	300	
		103				
16	HR	103				
		105				
		110				
		114				
18	Finance	103				
		114				
		110				

Page No.
Date

Step-3 : Identify the Dependencies

(ii) Proj Num → Proj Num
(iii) Emp Num → Emp Name, Job class, chrg-hours, hours
(iv) Job class → Chg Hour

The first Normal form describes the tabular format

(1) All the key attributes are identified
(2) There are no repeating groups in the table in ~~the~~ which each cell in a table contains one and only one value and not the set of value.
(3) All the attributes are dependent on the primary key.

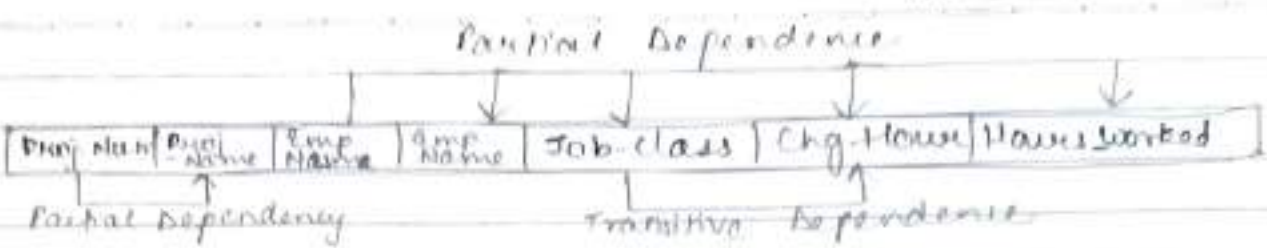
⇒ Anomalies - (in ~~2NF~~ (1NF)

(1) Insertion -
It is difficult to insert the employee details who are not yet assigned a project.

(2) Updation -
It is difficult to update employee details who ~~are~~ is working on multiple project.

(3) Deletion
If an employee working in the a project leaves the company and he is the only employee working in that project, deleting the details of the employee will also delete the project details.

NOTE - All these anomalies arrive due to the existence of partial dependency.



1NF (Proj_Num, Emp_Num, Proj_Name, Emp_Name, Job_class, chg_hours, Hours_worked)

Partial Dependency

Proj_Num → Proj_Num

Emp_Num → Emp_Name, Job_class, chg_Hour, Hours_worked

Transitive Dependency

Job_class → chg_Hour

02/04/19

⇒ Conversion to Second Normal Form (2NF)

→ It is done when 1NF has a composite primary key.

→ If 1NF has a single attribute primary key then the table is automatically 2NF

— Steps to convert

Step-1- Write each key component on a separate line and the original (composite key) on last line.

Proj_Num

Empl_Num

Proj_Num, Empl_Num

Each component will become a key in the new table

The original table will be divided into 3 tables.
PROJECT (Proj-Num)
Employee (Emp-Num)
Assignment (Proj-Num, Emp-Num)

Step-2 - Assign the corresponding depending attributes
PROJECT (Proj-Num, Proj-Name)
Employee (Emp-Num, Emp-Name, Job-class, Chg-Hour)
Assignment (Proj-Num, Emp-Num, Hours)

2NF

Table 1: PROJECT

Proj-Num	Proj-Name

Direct relationship

Table 2: Employee

Empl-Num	Emp-Name	Job-class	Chg-Hour

Transitive Dependency

Table 3: Assignment

Proj-Num	Emp-Num	Assign-Hour

⇒ Anamolies :-

(1) Since the 2NF has transitive dependencies which will generate Anamolies

(2) for eg :- If the charge per hour changes for a job classification held by many employees then that change must be made for each of that employee.

(3) If we forget to update some of the employee records then different employee with same job description will generate different hourly charges.

NOTE:
A table is in 3NF when
① it is in 2NF and
② it includes no partial dependencies i.e that no attribute is dependent on a portion of the primary key.

⇒ Conversion to 3NF:

- Step-1

- A Determinant is any attribute whose value determines other values within ~~the~~ a row.
- If we have 3 transitive dependencies then we will have 3 different determinants.
- In our example Job-class is the determinant.

- Step 2

Identify the dependent attributes on each determinant.
eg- Job-class → chg-hour.

- Step-3

Remove the dependent attributes from the original table

- Step 4

Create new table with determinant and its dependent attributes as a primary key.
eg- JOB(Job-class, chg-hour)

(21)

Page No.
Date

Table 1 : Project

Proj_Num	Proj_Name

Table 2 : Employee

Emp_Num	Emp_Name	Job_class

Table 3: Job

Job_class	chg_hour

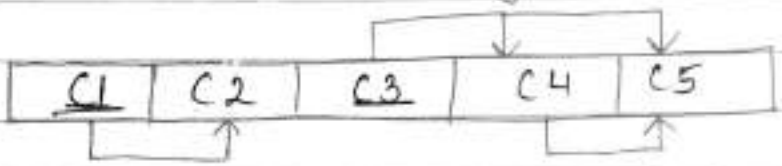
Table 4 : Assignment

Proj_Num	Emp_Num	Assign_Hour

A table is in 3NF -

1. It is in 2NF
2. It does not contain any transitive dependencies

Q.



Underlined - Primary Key - C_1, C_3
Underlined - Prime attribute
Non-underlined - Non-prime attribute

2NF

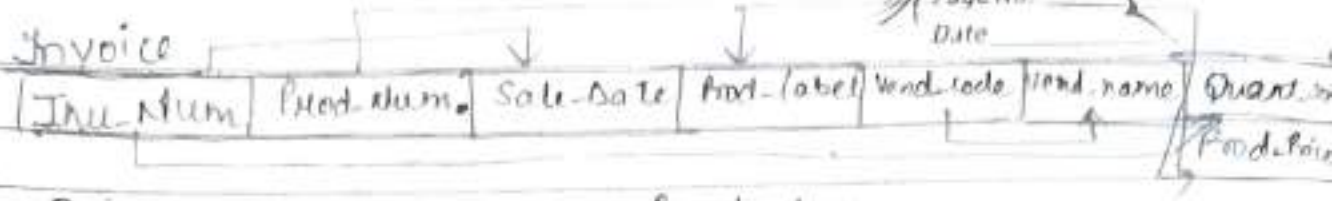
Table 1 (C_1, C_2) // C_1 is identifying C_2
Table 2 (C_3, C_4, C_5)
Table 3 (C_1, C_3) // no attribute simply write

3NF

Table 4 (C_4, C_5)
Table 2 (C_3, C_4)

Page No.
Date

Invoice



Primary key - Inv_Num, Prod_Num
Normal form - ~~1NF~~ 1NF
Dependencies - Prod_Num → Prod_label, Prod_Price
Inv_Num → Sale_Date
Vend_code → Vend_Name
Prod_Num, Inv_Num → Quant_Sold

2NF

Product (Prod_Num)
SELL (Inv_Num, Prod_Num)

Step-2 :- Invoice (Inv_Num, Sale_Date)
Prod (Prod_Num, Prod_label, Prod_Price, Ven_code, Ven_Name)
Sell (Inv_Num, Prod_Num, Qty_sold)

3NF

Step-1 :- Identify determinant
(Determinant are non-prime attribute which identify row)
In our eg :-
Vend_code is determinant

Step-2 :- Identify dependent attribute
Vend_code → Vend_Name

Step3. Remove dep. attribute from original table
"Product" table remove Vend_Name

Page No.
Date

Invoice (Inv. Name , Sale. Date)
Prod (Prod. Num, Prod. label, Prod_price, Vend_code)
Vendor (vend. code, Vend. Name)
Sell (Inv_Num, Prod_Num, Qty_Sold)

Q. What are the 3 possibilities which exist when we delete a tuple that is a target of a foreign key reference?

Ans: Employee

Emp_Num	Emp_Name	DNo

f.k.

Department

D_Num	D_Name

All the tuples that contain reference to the deleted tuple should also be deleted.
This is referred as **Cascading Deletion**.

Emp_Num	Emp_Name	DNo
4		
4		
4		

all removed

D_Num	D_Name
4	Testing ——— X

(II) Only tuples that are not referenced by another tuple can be deleted
In other words a tuple referenced by other tuple in a database cannot be deleted.

1		1	Software Development
2	→	2	Maintenance
2	→	3	HR
3	→	4	Testing X
3			

(III) The tuple is deleted and the F.K. attributes of all the referencing tuple are said to null.

NULL	1 Software_Development
2	2 Maintenance
2	3 HR
3	4 Testing
3	