UNIT - 3

Normalization → It is the name given to the process of simplifying the relationship among data elements in a record.

* Normalization replaces a collection of data in a record structure by another record design which is simpler, more predictable and therefore more manageable.
* The goal of relational Database design is to generate a set of relation scheme that allow us to store information without any repititive/redundant data.
* It allows us to retrieve data/information and more efficiently.

The First Step towards normalization is to convert E-R Model into table or relation. The Next Step is to examine the tables for redundancy and if necessary, change them to non-redundant form. This non-redundant model is then converted to a Database Definition, which achieves the objectives of Database design phase.

Need for Normalization

Normalization reduces redundancy. Redundancy is the unnecessary repitition of a field. It can cause problems with storage, retrival and updation of data.

Redundancy can lead to
- (a) Inconsistensis → Errors are more likely to occur when facts are repeated.
- (b) Update Anamolies → Inserting, modifying and deleting data may cause inconsistensis.

* Inconsistency occurs when we perform updation or deletion of data in one relation, while forgetting to make corresponding changes in other relation.

A Fully Normalized Record consist of -
(a) A primary key that identifies that entity.
(b) A set of attributes that describes that entity.

Types of Normalization

1. First Normal Form (1NF)

Salesperson | Sales

Employee Number	Employee Name	Store Branch	Dept.	Item No	Item Description	Sale Price
21130680	Anand K	Downtown	Hardware	TR10	Router	35.00
				SA1	Saw	19.00
				PT6	Drill	21.00
				AB16	Lawnmower	245.00
30142101	Zadoo S	Dadeland	Home Appliance	TT1	Humidifier	114.00
				DS10	Dishwasher	262.00

Salesperson

Sales

Employee Number	Employee Name	Store Branch	Dept.	Item No	Item Description	Sale Price
41984620	Balwant	Cutter Point	Autoparts	MC16	Snow time	85.00
				AC146	Alternator	65.00
				BB100	Batter	49.00
61204721	Bhagwan	Fashion Spot	Men's Clothing	MS10	Suit	215.00

Fig 1:- Unnormalized File for Sales

* Employee Number	Employee Name	Store Branch	Dept
21130680	Anand K	Downtown	Hardware
30142101	Zadoo S	Dadeland	Home Appliances
41984620	Balwant	Cutter Point	Autoparts
61204721	Bhagwan	Fashion Spot	Men's Clothing

IMP *

Fig 2(a):- Salesman Data File

* Employee Number	* Item Number	Item Description	Sales Price
21130680	TR10	Router	35.00
21130680	SA1	Saw	19.00
21130680	PT6	Drill	21.00
21130680	AB16	Lawnmower	245.00
30142101	TT1	Humidfier	114.00
30142101	DS10	Dish Washer	262.00
41984620	MC16	Snow time	85.00
41984620	AC146	Alternator	65.00
41984620	BB100	Batter	49.00
61204721	MS10	Suit	215.00

Fig 2(b):- Salesperson Item File

Fig 2:- First Normalized File for Sales

A table is in the First Normal Form when it contains no repeating groups. The repeating columns or field present in an unnormalized table are removed from the table and put into seperate table or tables. These tables are dependent on parent table from which it is derived. The key to these tables must also be a part of parent table, so that the parent table and the derived tables can be related to each other. Isolate repeating groups from an entity because they are easier to process seperately, from rest of the entity.

IMP * MAIN DEFINITION → When a table has no repeating groups, it is said to be in First Normal Form [1NF] that is, for each cell in a table (one row and one column), there can be only one value. This value should be atomic in the sense that it cannot be decompose into smaller pieces.

As in Fig 1, first four attributes [Employee Number, Employee Name, Store Branch, Dept] are virtually constant.

The remaining 3 attributes (Item Number, Item Description, Sale Price) contain data that change and are repeated with different sales person. Therefore, the repeating group should be seperated from the entity "Sales Person". The Normalized file is shown in Fig 2. It consist of two files -

(A) The Salesperson Data File with employee number as primary key.

(B) The Salesperson Item File with employee number and item number as new attributes. These attributes are added to relate the records in this files to the Salesperson Data File. The two attributes are used together for accessing data. Therefore, two keys are used together and such a key is called Catenated Key.

FUNCTIONAL DEPENDENCIES

* When in a given Relation R, attribute Y of R is functionally dependent on attribute X of R if and only if each x-value of R is associated with it precisely one y-value in R.

* A functional dependency is denoted by $X \rightarrow Y$, between two sets of attributes X and Y.

Suppliers [S]

S. No.	Name	Status	City
S1	Shyam	20	Bombay
S2	Ram	10	Calcutta
S3	Amit	30	Calcutta
S4	Chirag	20	Delhi
S5	Ramesh	30	Calcutta

S. SNo → S. Name

S. SNO → S. Status

S. SNO → S. City

* In suppliers databases, attributes name, status, and city of Relation S are each functionality dependent on attribute, SNo because given a particular value for SNo, there exist precisely one corresponding value for each of Name, Status and City.

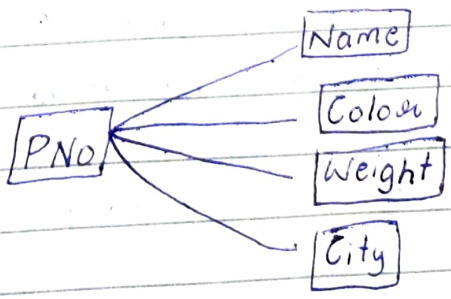
Definition → * Functional Dependency plays an important role in differentia-ting good Database design.

* Functional Dependency are the consequences of the inter-relationship among attributes of an entity representing in a relation.
* The Database containing information concerning suppliers (S) and parts (P)
* The suppliers and parts are uniquely identified by suppliers numbers (S.No) and Part No (P No).

Parts (P)

P. No.	Name	Colour	Weight	City
P_1	Nut	Red	12	Bombay
P_2	Bolt	Green	17	Calcutta
P_3	Screw	Blue	17	Goa
P_4	Screw	Red	14	Bombay
P_5	Handle	Blue	12	Bombay
P_6	Wire	Red	19	Delhi

P. PNo → P. Name
P. No → P. Colour
P. No → P. Weight
P. No → P. City



Ex → Student Relation (Name, Address, Subject, Grade)

Name → Address
Name Subject → Grade

Functional Dependencies of Student Relation

* The name of a student is unique.
* In each subject a student gets a unique grade.

Ex → Employee Relation (Emp-code, Emp-Name, Dept, Grade, Salary, Age, Address)

f1: Emp_code → Emp_name;
Each employee has a unique Emp-code.

f2: Emp_code → Dept;
An employee can work only in one department only

f3: Emp_code Grade Age → Salary;
Employee salary is depends on his Age and Grade

f4: Emp_code → Age;
Each employee has a unique age.

f5: Emp_code → Address;
Each employee has unique address.

* In the relation, employee code is not functionally dependent on salary or age, because more than one employee can have

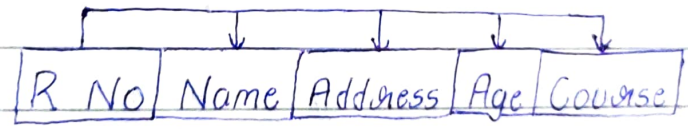
the same salary. and can be of the

Types of Functional Dependencies

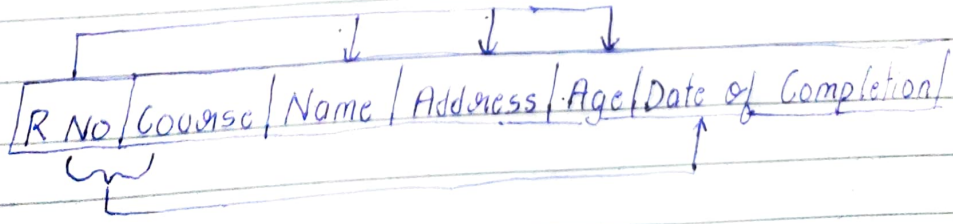
1. Full-functional ~~Dep~~ Dependency
2. Partial Dependency
3. Transitive Dependency
4. Multivalue Dependency

1. Full Functional Dependency → When all non-key attributes are dependent on key attribute then it is called Full Functional Dependency.

For Ex → In student relation, the non-key attribute (name, age, address and course) are dependent on key attribute roll no.



2. Partial Dependency → Partial dependency in a record type occurs when some non-key attributes depend on the key attribute and the ~~key attribute~~ remaining non-key attributes depend on key attribute. and on one or more non-key attributes.



→ The non-key attributes Name, Address, and age are dependent on RNo and Date of Completion i.e. non-key attributes depends on RNo (key attribute) as well as course (non-key attributes).

3. Transitive Dependency

(a)

SNo	Origin	Destination	Distance

(b)

RNo	Dept	Year	Hoste

→ If A, B, C are 3 columns in a table
If C is related to B
If B is related to A
Then C is directly related to A

A Functional Dependency $x \rightarrow y$ in a relation scheme R, is a transitive dependency if there is a set of attributes. Z that is neither a candidate nor a subset key of R and both $x \rightarrow z$ and $z \rightarrow y$ holds.

* When one non-key attribute depend on other non-key attribute it is called as transitive dependency.

→ To calculate distance in figure (a) which is a non-key attribute we must know the origin and destination which are also non-key attribute.

Atomic → Single Value

→ In figure (b) if the student is in first year, then he will be assign hostel gandhi if he is in second year, ~~hostel~~ then he will be assign hostel jawahar, and if he is in third year, he will be assign hostel indra. Therefore, the hostel assigned to a student is dependent on the year of study in the college.

4. Multivalued Functional Dependency

In this, entities of the dependent set are not dependent on each other.
i.e. If $a \rightarrow \{b, c\}$ and there exists no functional dependency between b and c, then it is called a multivalued functional Dependency.

Car_model	Manufacture_Year	Color
H002	2017	Mettalic
H002	2018	Greeen
H005	2017	Blue
H0012	2012	Mettalic
H0033	2010	Grey

→ Manufacturing year and and color are independent of each other but dependent on car_model.

car_model →→ Manufacture_Year

car_model →→ Color

(2) Second Normal Form [2NF] → A relation that is in first normal form and every non-primary key attribute is fully-functionally dependendent on the primary key, then the relation is in Second Normal Form [2NF].

* A table is in the 2NF, if all its non-key fields are fully dependent on the whole key. This means that each field in the table, must depend upon the entire key. Those that do not depend upon the combination key are moved to another table on whose key they depend upon.

* The Second Normalization make sure that each non-key attribute depends on a key-attributes. For Ex → In given figure each attribute in the sales person data file depends on the primary key [employee_number]. In the Salesperson Item File, the attribute [sales Price] depends on a composite key [employee-number] and [item_number].

* Employee Number	Employee Name	Store Branch	Dept
21130680	Anand K	Downtown	Hardware
30142101	Zadoo S	Dadeland	Home Appliances
41984620	Balwant	Cutter Point	Autoparts
61204721	Bhagwan	Fashion Spot	Men's Clothing

Fig :- Sales Person Data File

* Employee	* Item	Sales Price [₹]
21130680	TR10	35.00
21130680	SA1	19.00
21130680	PT6	21.00

21130680	AB16	245.00
3014 2101	TT2	114.00
3014 2101	DS10	262.00
41984620	MC16	85.00
41984620	AC146	65.00
41984620	BB100	49.50
61204721	HS10	215.00

Fig:- Sales Person Item File

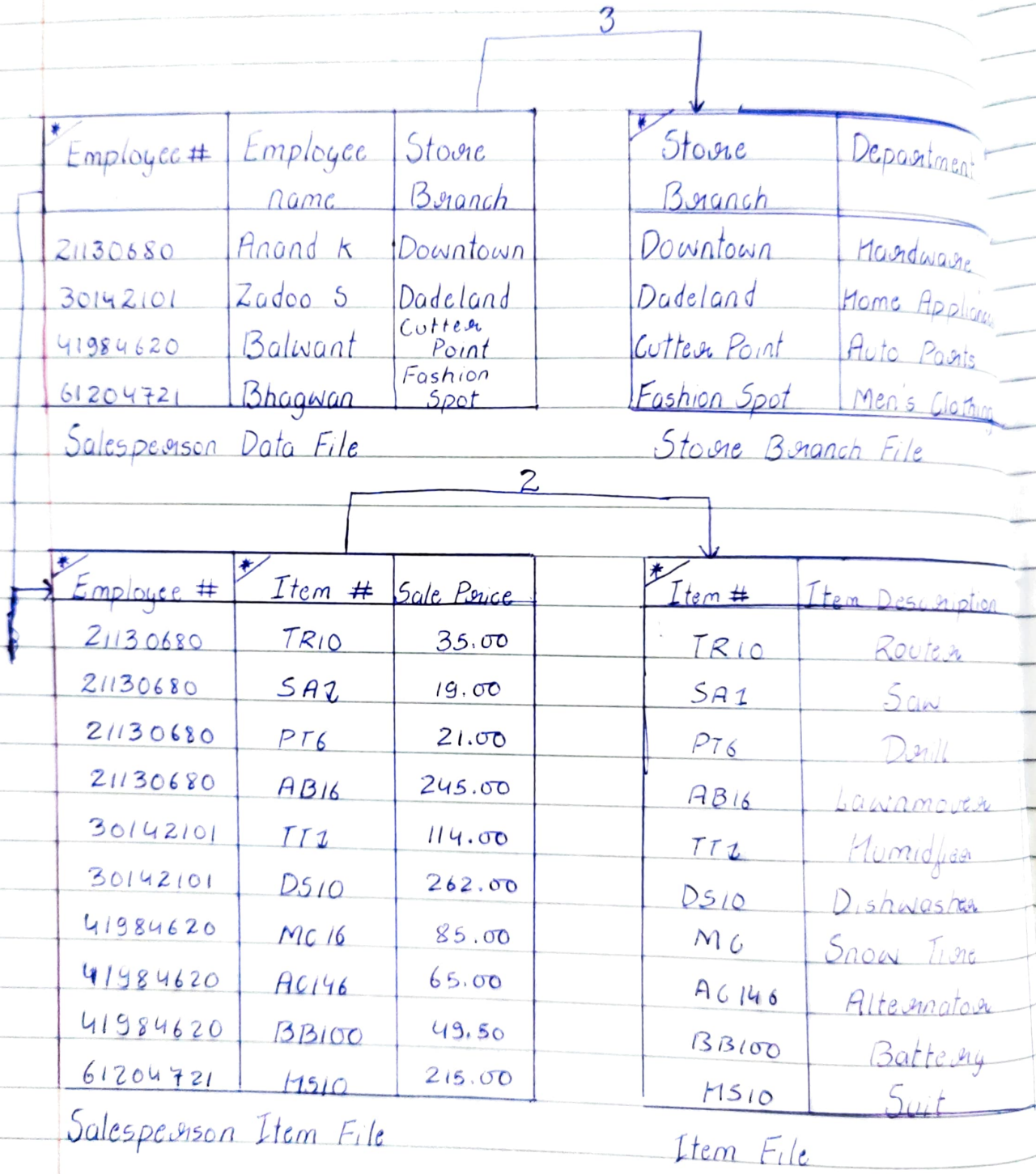
Item	Item Description
TR10	Router
SA1	Saw
RT6	Drill
AB16	Lawn Mover
TT2	Humidfier
DS10	Dish Washer
MC16	Snow Tier
AC146	Alternator
BB100	Battery
HS10	Suit

Fig:- Item File

③ Third Normal Form [3NF] → A table is said to be 3NF if all the non-key fields of the tables are independent of all other non-key fields of the table.

→ In 3NF, no non-primary attribute is functionally dependent on other non-primary attributes.

→ This normalization simplifies the relationship and provide logical links between files without using the information.



* Employee #	Employee name	Store Branch
21130680	Anand K	Downtown
30142101	Zadoo S	Dadeland
41984620	Balwant	Cutter Point
61204721	Bhagwan	Fashion Spot

Salesperson Data File

* Store Branch	Department
Downtown	Hardware
Dadeland	Home Appliances
Cutter Point	Auto Parts
Fashion Spot	Men's Clothing

Store Branch File

* Employee #	* Item #	Sale Price
21130680	TR10	35.00
21130680	SA2	19.00
21130680	PT6	21.00
21130680	AB16	245.00
30142101	TT2	114.00
30142101	DS10	262.00
41984620	MC16	85.00
41984620	AC146	65.00
41984620	BB100	49.50
61204721	MS10	215.00

Salesperson Item File

* Item #	Item Description
TR10	Router
SA1	Saw
PT6	Drill
AB16	Lawnmower
TT2	Humidifier
DS10	Dishwasher
MC	Snow Tire
AC146	Alternator
BB100	Battery
MS10	Suit

Item File

Fig:- Third Normalization

BCNF [BOYCE-CODD NORMAL FORM]

* BCNF was proposed as a simpler form of 3NF, but it is much more strict than 3NF, meaning that every relation in BCNF is also in 3NF. However, a relation in 3NF is not necessarily be in BCNF.

* For Example → Consider a relation scheme in 3NF that has a number of overlapping composite candidate keys.
Consider the Relation GRADE (Name, Student #, Course, Grade).

Here are the Functional Dependencies are :-

Name Course → Grade
Student # Course → Grade
Name → Student #
Student # → Name

Name	Student #	Course	Grade
James	23714	353	A
Neelam	47217	329	A
James	23714	328	In Progress
Mohan	83331	456	C
Dilip	43477	293	B
Deepak	31111	491	C
James	23714	353	A
Raj	67811	444	At
Vikas	91111	379	In Progress

Table : THE GRADE RELATION

Each student has a unique name and a unique student number.
The Relation has two candidate keys:- (Name, Course) (Student #, Course).
Each of these is a composite key and contain a common attribute course.
Here, we have 2 overlapping candidate keys:

The problem in relation grade is that it had 2 overlapping candidate keys. In the BCNF which is stronger than the 3NF, the intention is to avoid the above anomaly.

This is done by insuring that for all non-trivial functional dependencies implied by the relation, determine the functional dependencies that involve candidate key.

TRANSACTION → A transaction is a program unit whose execution may change or may not change the contents of a DB.

ACID Properties of a transaction:- (i) Atomicity
(ii) Consistency
(iii) Isolation
(iv) Durability

[We have studied all ~~the~~ about the properties in UNIT-I]

LIFE CYCLE OF A TRANSACTION

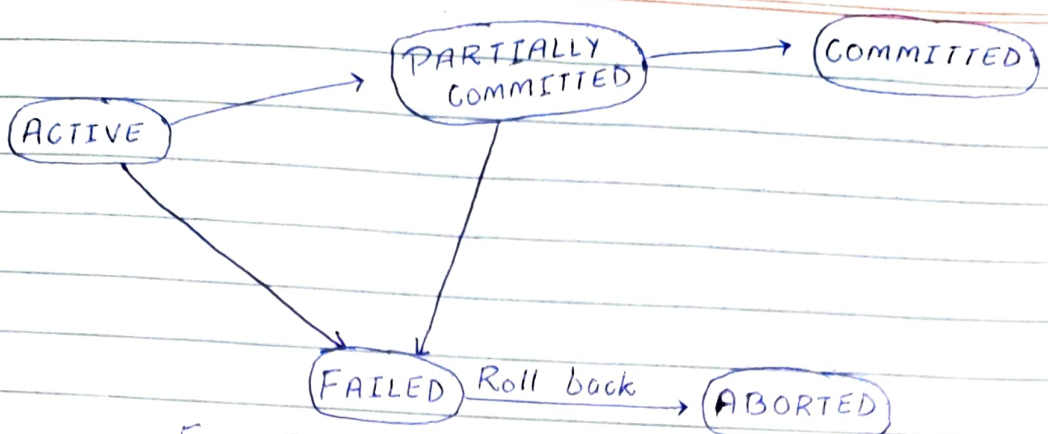


Fig:- State diagram of a Transaction

A Transaction starts in a active state and when it about to finish it enter into the partially committed state. At this point, the T has completed the execution but it is still possible that it may have to be aborted [since the actual output may still be temporaly residing in main memory and a hardware failure may precede its successful completion].

A Transaction enter the fail state after the system determines that the Transaction can no longer proceed with its normal execution due to hardware or logical errors. Such a Transaction must be roll back, then it enters aborted state. At this point, the system has 2 options -

(a) it can restart the Transaction, but only if the Transaction was aborted as a result of some hardware and software error, that was not created through the internal logic of the Transaction.

(b) It can Kill the Transaction. It usually does so because of some internal logic errors that can be corrected only by rewriting the application or program.

Page No.
Date : / /

ROLL BACK → The changes caused by the aborted Transaction are undone [undo] in the Database.