Unit - IV

Normalization (schema Refinement)

Purpose of Normalization :

Student

Sid	Sname	credicts	Dept_name	Building	Room_no
1.	Rahul	5	CSE	B_1	101
2	Ramu	8	CSE	B_1	101
3	Suma	9	AIML	B_2	201
4.	Hema	9	AIML	B_2	201
5.	Hari	7	Civil	B_1	110
6.	pallavi	8	ECE	B_1	115
7	Rani	8	Civil	B_1	110
8	Divya	7	CSE	B_1	101

Def : The purpose of Normalization in a DBMS is to organize data in a way that eliminates redudant data and anomalies. This is done through a process called Normalization which involves breaking down tables to remove and undesirable characteristics.

Normalization : It is the process of minimizing redundancy from a relation (or) set of relations.

Redundancy in a relation may cause insertion, updation & deletion anamalies.

1. Insertion Anamaly (occurs during inserting data) :

If unable to insert data without the presence of others.

2. updation Anamaly (occurs during updating the data)

change of row values may change all the similar type of values else may redundancy occur. which leads to loss of data integrity.

3. Deletion Anamoly (occurs during deletion of data)

If some data is deleted with deletion of other data (or) unnecessary

In the above table, the department information is repeated that means redundancy may occur in the above table. The problems due to redundancy are :

insertion anamoly :

we want to insert department information i.e,

ME	B1	120

we can't insert this information until one student enroll in it

updation anamoly :

Suppose we are updating the CSE dept information i.e, building to C1 & room no to 301

As CSE dept info is appeared in three tuples (or) rows, but we forgot to update one tuple then data inconsistency may arise. This is nlg but updation anamoly.

Deletion anamoly :

In the above table, we want to delete student sid = 6 information. but along with the student information the ECE dept details also deleted. This is nlq but Deletion anamoly.

→ By Breaking the above table into two tables we reduce the redundancy and anamalies that occur in the table.

Student

Sid	Sname	Credits	Dept-name
1.	Rahul	5	CSE
2	Ramu	8	CSE
3	Suma	9	AIML
4	Hema	9	AIML
5	Hari	7	civil
6	pallavi	8	ECE
7.	Rani	8	civil
8	Divya	7	CSE

Dept

Dept name	Building	Roomno
CSE	B_1	101
AIMl	B_2	201
civil	B_1	110
ECE	B_1	115

Functional dependency :

In a relational dbms, the functional dependency is a relationship that exists between two sets of attributes. Where 1 attribute determines the value of another attribute. It is denoted by $x \to y$ where the attribute set on the left side of the arrow x is called determinant. and y is called dependent, and arrow represents the functional dependency.

Scanned with OKEN Scanner

The condition for the functional dependency $x \to y$ exists is

if $t_1.x = t_2.x$

then $t_1.y = t_2.y$

Student

Rno	Name	Marks	Dept	course
1	a	78	CS	C_1
2	b	60	EE	C_2
3	a	78	CS	C_2
4	b	60	EE	C_3
5	c	80	IT	C_3
6	a	80	EC	C_2

all different means occurs FD

X → Y	FD
Rno → Name	✓
Name → Rno	×
Rno → Marks	✓
Dept → course	×
course → Dept	×
Marks → Dept	×
(Rno, name) → marks	✓
Name → marks	×
(Name, marks) → (Dept, course)	×
Rno → (name, marks)	✓
(Dept, course) → name	✓
(Rno, marks) → Dept	✓
Name → course	×

Types of Functional Dependency : Non-trivial, Multi-valued, Transitive

1. Trivial F.D :

student

Rno	Name	Marks	Dept	City
1.	Ramu	90	cse	Delhi
2	Ramesh	80	IT	Mumbai
3	Suresh	90	ece	Hyd
4.	Ramu	60	mech	Vijayawada

1. Trivial F.D :

In trivial F.D, a dependent is always a subset of the determinant

i.e, if $A \rightarrow B$, B is subset of A $(B \subseteq A)$

then it is called Trivial F.D

Ex : {rno, name} → rno

rno → rno

2. Non-trivial F D :

In non-trivial F.D. the dependent is strictly not a subset of the determinant

i.e, if $A \rightarrow B$, is F.D. and B is not subset of A $(A \cap B = \phi)$ then it is called non-trivial F.D

Ex : rno → name

{rno, name} → marks.

Scanned with OKEN Scanner

Multivalued F.D :

In a Multivalued F.D, entities of the dependent set are not dependent on each other.

$A \to BC$ is a MVD, then B &C should not be dependent i.e $\left.\begin{array}{l} B \to C \\ C \to B \end{array}\right\}$ is not F.D

There exists no F.D b/w B &C. then it is called MVFD.

Ex: Rno $\longrightarrow$ {name, Marks} is MVD

Where, name $\to$ marks is not F.D

marks $\to$ name is not F.D

Transitive F.D :

In Transitive F.D, dependent is directly dependent on determinant i.e if $A \to B$ & $B \to C$, then according to "axiom of transitivity" $A \to C$. This is a Transitive FD

Ex: $\left.\begin{array}{l} \text{Rno} \to \text{city} \\ \text{city} \to \text{Dept} \end{array}\right\}$ FD then Rno $\to$ dept is TFD

Armstrong Axioms / Inference Rules :

Rno	Name	Marks	Dept	Course
1	a	78	CS	C_1
2	b	60	EE	C_1
3	a	78	CS	C_2
4	b	60	EE	C_3
5	c	80	IT	C_2

Scanned with OKEN Scanner

1. Reflexivity :

Same as trivial property

$X \to Y$ FD

$Y \subseteq X$, $X \to X$

Ex : {Rno, name} → name

name ⊆ {rno, name}

2. Transitivity :

1, 2, 3

primary rules

if $(A \to B$ & $B \to C)$ then $A \to C$

Ex : Name → marks
marks → dept } then name → dept

3. Augumentation :

make
adding something bigger by adding Something

if $X \to Y$ then $XA \to YA$

Ex : Rno → Name

{Rno, marks} → {name, marks}

4. Union :

if $x \to y$ &
$x \to z$

then $x \to yz$

Ex : Rno → name

Rno → marks

then Rno → (name, marks)

5. Decomposition (Splitting) :

if $x \to yz$ then $x \to y$, $x \to z$

Ex : Rno → (name, marks)

then Rno → name

Rno → marks

Scanned with OKEN Scanner

6. pseudo Transitivity :

if $x \to y$ & $yz \to A$

then $xz \to A$

Ex : Rno $\to$ name

name, marks $\to$ dept

then Rno, marks $\to$ dept

7. Composition :

if $x \to Y$ & $A \to B$ then $XA \to YB$

Ex : Rno $\to$ name

marks $\to$ Dept

then Rno, marks $\to$ name, dept.

Keys in DBMS

attribute/columns

Student Table/relation

Sid	Name	Marks	Dept	Course
1	a	78	CS	C_1
2	b	60	EE	C_1
3	a	78	CS	C_2
4	b	60	EE	C_3
5	c	80	IT	C_2

→ A key is an attribute (or) set of attributes that uniquely identify each record in the table.

Ex : {Sid} ⟶ Key

{name, marks} → not a key

{Dept, course} → Key

Types of Keys :

Super Key : A super key is an attribute (or) set of attributes that uniquely identify each record in the relation.

Ex : {sid , (sid,name), (sid, marks) , (sid, dept), (sid, course), (name, course) , (marks, course), (Dept, course) .... }

→ To find the maximum no of super key : $2^n - 1$

∴ Here , 'n' is is no. of columns.

A	B	C	D
1	1	5	1
2	1	7	1
3	1	7	1
4	2	7	1
5	2	5	1
6	2	5	2

Superkey = {A, AB, AC, AD, ABC, ACD, ABD, ABCD} = 8

C.K = {A}

proper subset

$S_1 = \{1,2,3\}$, $S_2 = \{1,2\}$

$S_2 \subseteq S_1$ subset

$S_1 \nsubseteq S_2$ not subset

$S_2 \subset S_1$ proper "

Candidate Key :

A candidate key is also a superkey whose proper subset is not a superkey

A	B	C
1	1	1
2	1	2
3	2	1
4	2	2

Scanned with OKEN Scanner

Super key = {A, AB, AC, BC, ABC}

BC = {B, C}

A = {}

Candidate Key = {A, BC}

Minimal Superkey :

→ It is Candidate key

A B ~~C~~ BC

A ~~B~~ ↓

A minimal S.K

Primary key : One candidate key that have no null value that can be selected as primary key.

→ Remaining C.K are selected as Alternate keys (or) Secondary keys

Attribute closure | closure Set

X — Set of attributes

closure of X, X^+ — Contains set of attributes determined by X

→ The purpose of closure set to find Super Key and candidate key.

Ex : R (A, B, C, D, E) FD - {A→B, D→E}

A^+ = {A, B}

$ABCDE^+$ = {A, B, C, D, E} → Super Key

all elements present in relation

Scanned with OKEN Scanner

$ABCDE^+ = \{A, B, C, D, E\}$

$ACDE^+ = \{A, C, D, E, B\}$ $\rightarrow$ SK

$ACD^+ = \{A, C, D, B, E\}$

↓

candidate key

Check in F.D whether dependent is A, C, D if not ACD is only C.K

∴ prime attributes = {A, C, D}

R(A, B, C, D) FD – {A→B, B→C, C→A}

$ABCD^+ = \{A, B, C, D\}$

$ACD^+ = \{A, C, D, B\}$

$AD^+ = \{A, D, B, C\}$

AD = {A, D} not in S.K

AD is C.K

CD → {C, D} not in S.K

CD is C.K

BD → {B, D} not in S.K

BD is CK

C.K = {AD, BD, CD}

prime attributes = {A, B, C, D}

Full functional dependency & partial F.D

Rno	name	marks	dept	course
1	a	78	cs	C_1
2	b	60	ee	C_1
3	a	78	cs	C_2
4	b	60	ee	C_3

Scanned with OKEN Scanner

(name, marks) $\rightarrow$ dept

name $\rightarrow$ dept

partial F.D

(dept, course) $\rightarrow$ Rno

dept $\not\rightarrow$ Rno

full F.D

Scanned with OKEN Scanner

1st Normal Form :

A relation (or) table is said to 1NF if

i) Each column in a table must contain atomic Values.
ii) Each column name must be unique
iii) Attribute domain should not be change
iv) Order of rows is not important

Example :

Roll NO	name	course
1	Sai	c/c++
2	Harsha	Java
3	Depak	c/DBMS

Here, as the course attribute contains multiple values the above table is not 1NF. It is changed as below

Rno	name	course
1	Sai	c
1	Sai	c++
2	Harsha	Java
3	Depak	c
3	Depak	DBMS

The above table is in 1NF. Here Rno + course is primary key.

2nd Normal Form :

A relation (or) table is said to be 2NF if

i) relation should be in 1NF
ii) Relation should not have partial functional dependence

SId	cid	c-fee
101	Java	10000
102	python	15000
101	C++	2000
103	C	10000
103	Java	10000
102	Java script	20000

Sid, cid $\rightarrow$ c_fee

sid $\not\rightarrow$ c_fee (not determined)

cid $\rightarrow$ c - fee

so partial dependency exists, the above table is not in 2NF

So we decompose the above table into two tables.

Student

Sid	cid
101	Java
102	python
101	C++
103	C
103	Java
102	JS

Course

cid	c_fee
Java	10000
python	15000
C++	2000
C	10000
J.S	20000

Here, cid in Student table refers cid in course table

3 Normal Form :

A relation (or) table is said to be 3NF if

i) Relation should be in 2NF

ii) No transitive dependencies for Non prime attributes.

Rno	state	city
1	AP	Vjw
2	Telangana	Hyd
3.	AP	Vjw
4	Telangana	Hyd
5.	Banglore	Manglore

Rno → state ✓

Rno → city ✓

State → city ✓

State → Rno ×

City → Rno ×

city → State ✓

Here Rno → state

State → city ×

therefore, Rno → city but in state → city both state and city are non-prime attributes which dependend on one another & transitive dependency exists here.

The above is not 3NF table. So we decompose the above table into two tables as below:

Rno	State
1	AP
2	Telangana
3	AP
4.	Telangana
5.	Bangalore

State	city
AP	Vjw
Telangana	Hyd
Banglore	Manglore

Scanned with OKEN Scanner

Decomposition must follows Some rules :

i) Dependency preserving decomposition

ii) Lossless decomposition.

Dependency Preserving decomposition :

$R(A, B, C, D)$ $\quad F : \{A \to B,\ B \to C,\ C \to D\}$

$R_1(A, B)$ $\quad R_2(C, D)$

$F_1 : \{A \to B\}$ $\quad F_2 : \{C \to D\}$

$F_1 \cup F_2 = \{A \to B,\ C \to D\}$

$F \neq F_1 \cup F_2$

Dependency preserving is not exist

Ex : $R(A \quad B \quad C)$

A	B	C
1	1	1
2	1	2
3	2	1
4	2	2

$F : \{A \to B,\ A \to C,\ BC \to A\}$

$R_1(A, B)$

A	B
1	1
2	1
3	2
4	2

$R_2(B \quad C)$

B	C
1	1
1	2
2	1
2	1

$F_1 : \{A \to B\}$ $\quad F_2 = \{\ \}$

$F_1 \cup F_2 = \{A \to B\}$

$F \neq F_1 \cup F_2$

Scanned with OKEN Scanner

$R(A, B, C, D, E)$

$F : \{A \to B,\ B \to C,\ C \to D,\ D \to A\}$

$R_1\ (A, B, C) \quad R_2\ (C, D, E)$

$F_1 : \{A \to B,\ B \to C\} \quad F_2 : \{C \to D\}$

It is not proper way

$\to R_1\ (A, B, C)$

$A^+ : \cancel{A}BC\cancel{D} : A \to BC$

$B^+ : \cancel{B}C\cancel{D}A : B \to CA$

$C^+ : \cancel{C}\cancel{D}AB : C \to AB$

$AB^+ : \cancel{AB}C\cancel{D} : AB \not\to C$

$BC^+ : \cancel{BC}\cancel{D}A : BC \not\to A$

$AC^+ : \cancel{AC}\cancel{D}B : AC \not\to B$

$\to R_2\ (CDE)$

$C^+ : \cancel{C}D\cancel{AB} : C \to D$

$D^+ : \cancel{D}\cancel{AB}C : D \to C$

$E^+ : E$

$CD^+ : CDAB = \{\}$

$DE^+ : DE\ AB = \{\}$

$CE^+ : \cancel{CE}D\cancel{AB} : CE \to D$

$F_1 \cup F_2 = \{B \to CA,\ C \to AB,\ A \to BC,\ C \to D,\ D \to C\}$

$F_1 \cup F_2 \subseteq F$

$F \subseteq F_1 \cup F_2$

$F \equiv F_1 \cup F_2$

Here dependency preserving decomposition exists in Child tables R_1 & R_2

Lossless decomposition :

R

A	B	C
1	1	1
2	1	2
3	2	1
4	2	3

R_1

A	B
1	1
2	1
3	2
4	2

R_2

B	C
1	1
1	2
2	1
2	3

$R_1 \times R_2$

A	B	B	C
1	1	1	1
1	1	1	2
1	1	2	1
1	1	2	3
2	1	1	1
2	1	1	2
2	1	2	1
2	1	2	3
3	2	1	1
3	2	1	2
3	2	2	1
3	2	2	3
4	2	1	1
4	2	1	2
4	2	2	1
4	2	2	3

$R_1 \bowtie R_2$

A	B	C
1	1	1
1	1	2
2	1	1
2	1	2
3	2	1
3	2	3
4	2	1
4	2	3

$R_1 \cup R_2 = R$

$R_1 \cap R_2 \neq \phi$

$R_1 \cap R_2$

$R_1 \bowtie R_2 \neq R$ so it is not Lossless decomposition.

R_1

A	B
1	1
2	1
3	2
4	2

R_2

A	C
1	1
2	2
3	1
4	3

$R_1 \bowtie R_2$

A	B	C
1	1	1
2	1	2
3	2	1
4	2	3

$R_1 \bowtie R_2 = R$

$\therefore$ This is lossless decomposition

Scanned with OKEN Scanner

If R is decomposed into R_1 and R_2 then decomposition is lossless if

① $att(R_1) \cup att(R_2) = att(R)$

② $att(R_1) \cap att(R_2) \neq \phi$

③ $att(R_1) \cap att(R_2) \rightarrow att(R_1)$

(or)

$att(R_1) \cap att(R_2) \rightarrow att(R_2)$ the common attribute is superkey of atleast one sub relation.

Ex :

Consider a relation

Rno	name	dept
1	Raju	cse
2	Raju	ece

Student_details R_1

Rno	name
1	Raju
2	Raju

dept R_2

Rno	dept
1	cse
2	ece

① Stu_details ⋈ dept

Rno	name	dept
1	Raju	cse
2	Raju	ece

Here $R_1 \cup R_2 = R$

② $R_1 \cap R_2$ = Rno $\neq \phi$

③ $R_1 \cap R_2$ = Rno Superkey (R_1)

∴ Since 3 conditions are satisfied. R_1 & R_2 are lossless decomposition.

Scanned with OKEN Scanner

Lossy decomposition.

Rno	name	dept
1	Raju	cse
2	Raju	ece

Stu_details (R1) dept (R_2)

Rno	name
1	Raju
2	Raju

name	dept
Raju	cse
Raju	cse

Stu_details ⋈ dept

Rno	name	dept
1	Raju	cse
1	Raju	ece
2	Raju	cse
2	Raju	ece

Since $R_1 \cup R_2 \neq R$ R_1 & R_2 are lossy decomposition.

Boyce_codd Normal form

A table is said to be BCNF if

1. The table should be in 3NF
2. For every non trivial dependency $X \to Y$, x is the super key of the table i.e for every F.D L.H.S is superkey.

It is also called as 3.5 Normal form and it is advanced version of 3NF and it is stricter than 3NF consider a relation R

Student

Sname	Teacher	Subject
Jhansi	P.Naresh	Database
Jhansi	K.Suresh	C
Gopal	P.Naresh	Database
Gopal	R.Prasad	C

Here F.D's are (Sname, subject) → Teacher

and Teacher → subject

Here from the definition of BCNF, the L.H.S of F.D must be superkey. but in above F.D teacher → subject, teacher is not a superkey, which violates BCNF.

If X → Y violates BCNF then divide R into R_1(X, Y) and R_2(R-Y). So R is divided into two relation R_1(Stname, teacher) R_2(Teacher, Subject)

R_1

Sname	teacher
Jhansi	P.Naresh
Jhansi	K.Suresh
Gopal	P.Naresh
Gopal	R.Prasad

R_2

Teacher	Subject
P.Naresh	Database
K.Suresh	C
R.Prasad	C

Now, the table is in BCNF

Scanned with OKEN Scanner

4 Normal Form :

A relation is said to be in 4NF if

① It is in BCNF

② Has not Multivalued dependency

For a dependency A→B, if for single value of A, multiple of value of B exists, then the relation will be a multi-valued dependency.

Student

Sid	Course	Hobby
21	DBMS	dancing
21	python	singing
34	Java	dancing
74	Java	cricket
59	C++	hockey

The given table is in 3NF, but course & hobby are independent entity. Hence there is no relation b/w Course & hobby.

In the student relation, sid 21 contains two Courses : DBMS & python and two hobbies : dancing & singing. Hence it is in multivaled dependency on sid which leads to unnecessary repetition of rows

So to make the above table into 4NF, we can decompose into two tables.

Scanned with OKEN Scanner

Stu_course

Sid	course
21	DBMS
21	python
34	Java
74	C
59	C++

Stu_hobby

sid	Hobby
21	dancing
21	singing
34	dancing
74	Cricket
59	Hockey

5th Normal Form :

A Table is said to be in 5NF If

① The table should be in 4NF

② A table doesnot contain join dependency

It is also called as Project join NF (PJNF)

Agent	Company	product
Harish	ford	car
Harish	ford	truck
Harish	GM	car
Harish	GM	Truck
charan	ford	car

R_1 :

Agent	Company
Harish	ford
Harish	GM
charan	ford

R_2 :

Company	Product
ford	car
ford	Truck
GM	car
GM	Truck

R_3 :

Agent	product
Harish	car
Harish	Truck
Charan	car

Scanned with OKEN Scanner

Now, we perform $R_1 \bowtie R_2$

Agent	company	product
Harish	ford	car
Harish	ford	truck
Harish	GM	car
Harish	GM	truck
Charan	ford	car
Charan	ford	truck

Here $R_1 \bowtie R_2 \neq R$

$R_1 \bowtie R_2 \bowtie R_3$

Agent	company	product
Harish	ford	car
Harish	ford	truck
Harish	GM	car
Harish	GM	truck
Charan	ford	car

Here, $R_1 \bowtie R_2 \bowtie R_3 = R$

∴ The decomposition R into R_1, R_2, R_3 are Lossless

Surrogate key :

A column that is not generated from the data in the database is called a surrogate key is also known as synthetic primary key and it is generated automatically when a new record is insert into the table and can declared as primary key of the table.

Scanned with OKEN Scanner

Suppose we have a tables of different schools.

Now, suppose we want to merge the details of both schools in a single table as below

Regno	name	percentage
101	Hari	90
102	Suma	70
103	Ram	80

Regno	name	percentage
20101	Ramu	85
20102	gopi	65
20103	Kiran	35

Here, we observe that reg no cannot be primary key of above table as it doesnot match all the records of the table. Now, in this case, we have to artificicially create one primary key. and i.e, called as surrogate key.

Sl No	Reg no	name	percentage
1	101	Hari	90
2	102	Suma	70
3	103	Ram	80
4	20101	Ramu	85
5	20102	gopi	65
6	20103	Kiran	35

Scanned with OKEN Scanner