Lec 20: Introduction to Normalisation

Insertion, Deletion & Updation Anomaly

Normalisation is a technique to Remove or Reduce redundancy from the Table

SID	Sname	Age
1	RAM	20
2	Varun	25
1	RAM	20

Row level

Column Level

SID (Student ID)	Sname	Cid (Course ID)	Cname (Course Name)	FID (Faculty ID)	Fname (Faculty Name)	Salary
1	RAM	C_1	DBMS	F_1	John	30000
2	Ravi	C_2	JAVA	F_2	Bob	40000
3	Nitin	C_1	DBMS	F_1	John	30000
4	Amit	C_1	DBMS	F_1	John	30000

To handle row level duplicacy, have a primary key, which is unique and not null

In column level duplicacy, we have 3 types of anomaly.

* Insertion Anomaly
* Deletion Anomaly
* Updation Anomaly

Insersion anomaly

In the table above that contains university data, ~~Ex~~ if there is a new course introduced and ~~ex~~ if no student enrolls in that course, then that would be a problem.

Deletion anomaly

In the above table, if we delete the row where SID = 2 then the information about the CID will be removed.

Updation anomaly

If we want to update the salary of F.FID, we need to update the Salary 3 times, which takes more time.

To avoid all these kinds of anomaly, we can divide the table in to the partitions below. This is one of the solution

<u>SID</u>	Sname

<u>CID</u>	Cname

<u>FID</u>	Fname	Salary

Lec-21 First Normal Form in DBMS

⇒ Table should not contain any multivalued attribute

Student

Roll no	Name	Course
1	Sai	C/C++
2	Harsh	Java
3	Omkar	C/DBMS

Not in 1st NF

1)

Roll no	Name	Course
1	Sai	C
1	Sai	C++
2	Harsh	Java
3	Omkar	C
3	Omkar	DBMS

Primary key will be roll no and course combined

2)

Roll no	Name	Course 1	Course 2
1	Sai	C	C++
2	Harsh	Java	Null
3	Omkar	C	DBMS

Primary key will be Roll no.

Disadvantage of this is that if there are more than 2 courses, then we will need more columns.

3) PTO

3)

Rollno	Name
1	Sai
2	Harsh
3	Venkat

Rollno	Course
1	C
1	C++
2	Java
3	C
3	DBMS

Base Table

Primary key: Roll no Course

Foreign key: Rollno

→ Primary key: Roll no.

Lec 22: Finding Closure of Functional dependency in DBMS

Closure method is a way to find all the candidate keys possible.

Eg i) R(ABCD)

FD {A → B, B → C, C → D}

$A^+ = B, CDA$

$B^+ = BCD$

$C^+ = CD$

$D^+ = D$

Since A can determine all the attributes, A is the candidate.

Prime attribute = {A}

Non-prime attribute = {B, C, D}

$(AB)^+ = ABCD$

AB can be candidate key, but it is not minimal. So, AB can be a super key.

2) $R(ABCD)$

$FD = \{A \to B, B \to C, C \to D, D \to A\}$

$A^+ = ABCD$
$B^+ = BCDA$
$C^+ = CDAB$
$D^+ = DABC$

$\therefore$ The $CK = \{A, B, C, D\}$

Prime attribute is an attribute that is used to make candidate key.

In this case, prime attribute = $\{A, B, C, D\}$

Non-prime attribute = ϕ

3) $R(AB, CDE)$

$FD = \{A \to B, BC \to D, E \to C, D \to A\}$

1) Check what is there in the right side, all are depended.

2) Then add that attribute that completes the attribute set.

In this example,

~~A~~ BEDA are depended. So, add E on both the sides

$E^+ = BCDAE$

$E^+ = EC$

$AE^+ = ABECD$

$CK = \{AE, DE, BE\}$

~~$BE^+ = BECDA$~~

Since ~~B~~ A depends on D,

$DE^+ = DEABC$

Since $B \to D$

$BE^+ = BECDA$

$CE^+ = \times$

∴ Finally, $CK = \{AE, BE, DE\}$

Prime attributes $= \{A, D, B, E\}$

Non-prime attributes $= \{C\}$

Lec-23: Functional Dependency & its properties

Functional dependency: $X \to Y$

X Determines Y or Y is determined by X

Properties of FD

_/_/_

Reflexivity: If Y is subset of X, then X → Y

Augmentation: If X → Y, then XZ → YZ

Transitive: If X → Y and Y → Z, then X → Z

Union: If X → Y and X → Z, then X → YZ

Decomposition: If X → YZ then X → Y and X → Z
XY → Z, X → Z, Y ↛ Z, this is not true

Pseudotransitivity: If X → Y and WY → Z then WX → Z

Composition: If X → Y and Z → W then XZ → YW

If there is a confusion in Y then X acts as a factor to determine X.

Trivial FD:- Functional Dependency that is always true.

1) If Y is a subset of X, X → Y is a trivial FD

2) In such cases, LHS ∩ RHS ≠ ϕ

Eg:- SId Sname → SId.
SId is the intersection, which is not null set.

Non-trivial FD:- FD that satisfies the conditions

X → Y, X ∩ Y = ϕ

Eg:- SId → SName
SId → Semester
SId → PhoneNo.

Lec 24: <u>Second Normal Form</u> (2NF)

→ Table or relation must be in 1st Normal Form

→ All the non-prime attributes should be fully functional dependent on candidate key

<u>Customer</u>

Customer ID	Store ID	Location
1	1	Delhi
1	3	Mumbai
2	1	Delhi
3	2	Bengaluru
4	3	Mumbai

Candidate Key: Customer ID & Store ID

Prime Attributes: {Customer ID, Store ID}

Non Prime: Location

This table can be divided into 2 tables

Customer ID	Store ID
1	1
1	3
2	1
3	2
4	3

Store ID	Location
1	Delhi
2	~~Bengaluru~~ Bengaluru
3	Mumbai

→ There should be no partial dependency.

R(A B C D E F)

FD{C→F, E→A, EC→D, A→B}

EC = F A D B

EC^+ = ECFADB

1) CK = {EC}

2) Prime attributes = {E, C}

3) Non-Prime Attributes = {A, B, D, F}

Partial dependency :- {LHS should be proper subset of CK and RHS should be a Non-Prime Attribute}

C→F {PD}
E→A {PD}
EC→D {FD}
A→B {FD}

Lec 25: Third Normal Form in DBMS

→ Table or relation must be in ~~3rd~~ 2 NF

→ There should be no transitive dependency in table

Rollno	State	City
1	Punjab	Mohali
2	Haryana	Ambala
3	Punjab	Mohali
4	Haryana	Ambala
5	Bihar	Patna

Eg:- 1) R(ABCD)

FD: AB → C, C → D

CK: AB
PA: A, B
NPA: C, D

Here, Non-prime attribute determines prime attribute

2) R(ABCD)

FD: AB → CD, ~~...~~ D → A

CK: $AB^+ = ABCD$

$DB^+ = DBAC$

∴ CK = {AB, DB}
PA = {A, B, D}
NPA = {C}

For each FD,

LHS must be a CK or SK OR RHS is a prime attribute

Lec-26. BCNF Boyce Codd Normal Form (BCNF)

This table is in BCNF

Roll no	Name	Voter id	Age
1	Ravi	K0123	20
2	Varun	M034	21
3	Ravi	K786	23
4	Rahul	D286	21

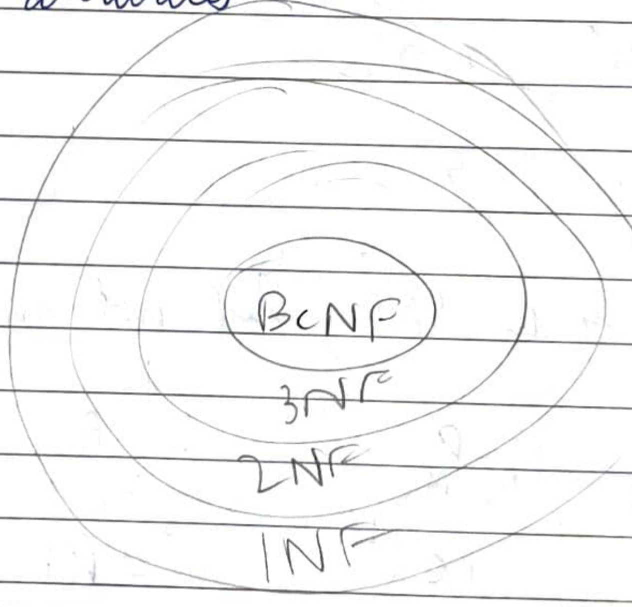
CK : {Rollno, Voter id

FD: Rollno → name
Roll no → Voter id
Voter id → age
Voter id → Roll no

Conditions of BCNF

i) The table should be in 3NF

ii) The Only candidate key must determine the non-prime attributes



Lec 22? BCNF Always ensures dependency preserving decomposition ?

* The 3NF always ensures "Dependency Preserving decomposition", but not in BCNF.

* Both 3NF & BCNF ensures lossless decomposition

$R(ABCD)$

$\{AB \to CD, D \to A\}$

$B^+ = B$

$AB^+ = ABCD$

$CK = \{AB, DB\}$

$CB^+ = CB$ X

$DB^+ = DBAC$

$PA = \{A, B, D\}$

$NPA = \{C\}$

→ This is in 3NF, but not in BCNF

$ABCD$ → R_1: DA $\{D \to A\}$, R_2: BCD $\{BD \to C\}$	$B^+ = B$ $C^+ = C$ $D^+ = DA$ A is not in the relation	$BC^+ = BC$ $BD^+ = BDAC$

In this example, if we take the closure of AB, we get only AB. Thus, the dependencies in BCNF are not preserved.

Lec - 28: Lossless & lossy decomposition

R

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

$\rightarrow R_1$

$\rightarrow R_2$

R_1

A	B
1	2
2	2
3	3

R_2

B	C
2	1
2	2
3	2

Query: Find the value of C if the value of A is 1.

~~SELECT~~ Select $R_2.C$ from R_2 Natural Join R_1
Where $R_1.A = '1'$

R_1 R_2 R'

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

Spurious Tuples

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

In R there were 3 tuples, but in R', there were 5 tuples. This is called lossy decomposition.

The extra tuples are called spurious tuples

To avoid lossy decomposition, the common attribute should be CK or SK of either R_1 or R_2 or Both

In R table A is fit to be the candidate key, as B and C contain duplicate values

Therefore the R table can be divided as $R_1(AB)$ & $R_2(AC)$

This decomposition is called lossless decomposition.

The condition of lossless decomposition is:

1) $R_1 \cup R_2 = R$

$AB \cup AC = ABC$

~~$ABC \cap C = ABC$~~

2) $R_1 \cap R_2 \neq \phi$

$AB \cap AC = A \neq \phi$

3) ~~R_1~~ The common attribute must be the candidate key of either R_1 or R_2 or both.

Lec-30: Minimal Cover in DBMS

For the following functional dependencies, find the correct minimal cover

$\{A \rightarrow B, C \rightarrow B, D \rightarrow ABC, AC \rightarrow D\}$

a) $A \rightarrow B, C \rightarrow B, D \rightarrow A, AC \rightarrow D$

b) $A \rightarrow B, C \rightarrow B, D \rightarrow C, AC \rightarrow D$

c) $A \rightarrow BC, D \rightarrow CA, AC \rightarrow D$

d) $A \rightarrow B, C \rightarrow B, D \rightarrow AC, AC \rightarrow D$

Soln. ~~$A \rightarrow B, C \rightarrow B, D \rightarrow A, D$~~

Step 1:- First ensure that the RHS contains only one attribute.

$A \to B$, $C \to B$, $D \to A$, $D \to B$, $D \to C$, $AC \to D$

Step 2:- Remove redundant dependencies.

If we ~~to~~ remove $A \to B$, we get $A^+ = A$
∴ we ~~cannot~~ retain that

$C^+ = C$, retain $C \to B$

$D^+ = \cancel{BC}\ DBC$, A is not there. Retain $D \to A$

~~B~~ Remove $D \to B$, $D^+ = ACBD$, ~~there~~ ∴ Remove.

Remove $D \to C$, $D^+ = A\cancel{C}B$, ∴ ~~Remove~~ retain

∴ We retain $A \to B$, $C \to B$, $D \to A$, $D \to C$, $AC \to D$

Step 3:- Take care of LHS

$A \to B$, ~~$B \to$~~ $C \to B$, $D \to AC$, $AC \to D$

∴ Option d is correct

Lec 35: Equivalence of FD

Eg 1:- $X = \{A \to B, B \to C\}$

$Y = \{A \to B, B \to C, A \to C\}$

If we want to check whether X & Y are equivalent, first we need to check whether X covers Y [$X \supseteq Y$] & if Y covers X [$Y \supseteq X$]

We have to find the closure of LHS of A, but with the FD's of Y.

$A^+ = ABC$

$B^+ = BC$

$\therefore$ X covers Y

To prove Y covers X.

$A^+ = ABC$

$B^+ = BC$

$\therefore$ Y covers X.

Eg 2:- $X = \{AB \to CD, B \to C, C \to D\}$

$Y = \{AB \to C, AB \to D, C \to D\}$

X covers Y

ABC

$AB^+ = CD$

$C^+ = CD$

Y covers X

$AB^+ = CDAB$

$\cancel{B^+ = BCD}$ $\quad B^+ = B$

$\cancel{C^+ = CD}$ $\quad C^+ = CD$

$B \to C$ is not covered by Y.

$\therefore$ Y does not cover X.

X and Y are not equivalent