Functional Dependency

FD is kind of integrity constraint that generalizes the concept of key. It occurs when one attribute in relation uniquely determines another attribute

determinant attribute ← $X \rightarrow Y$ ← dependent

x determines y

or

y determined by x

e.g.-

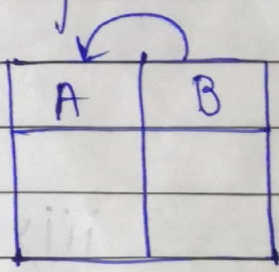
sid →	sname
1	John
1	John
1	John
2	John

It is a relationship that exists when one att uniquely determines another attribute

$X \rightarrow Y$

Y is functionally dependent on X

eg. If every attribute of B of R dependent of A, then attribute A is primary Key.



$A \rightarrow B$

sid	sname	s.surname
s_1	x	A
s_2	y	B
s_3	z	c

ID s_id → s_name
s_id → s_surname } table is functional dependence table

Purpose of Normalization

- If a database design is not perfect it may contain anomalies.
- Managing database with anamalies very difficult.
- Normalization is a technique for producing suitable relation that support data requirement.
- Normalization is a method to remove anomalies like

i) <u>Update anamalies</u> –
It we want to update data which is saved in different places, then few data will get updated other will be left as it is, such instance will leave data inconsistent.

ii) <u>Deletion anomalies</u> –
Same as above if we delete the some record in one place other data will be as it is because we are not aware about data storage.

Data Redundancy & Update Anomalies

- A major aim of relational database design is to group attributes into relation to minimize data redundancy & reduce file storage space.

- Data redundancy is a condition created within a database or data storage technology in which same copy of data is stored in 2 seperate places

e.g. -

Staff

Staff_no	S_name	posn	salary	branch_no
S_1	John	Manager	10000	B005
S_2	Ann	Asst	5000	B003
S_3	Suhas	Supervisor	7000	B007
S_4	Julie	Asst.	5000	B007
S_5	Mary	Asst.	5000	B005

Branch

Branch_no	B_address
B005	22 deccan pune
B003	10, bandra, Mumbai
B007	32, Main st, Nasik

StaffBranch

Staff_no	Sname	posn	salary	b_no	b_add
S_1	John	Mana.	10000	B005	22,d,Pune
S_2	Ann	Asst	5000	B003	10,b,Mum
S_3	Suhas	Super.	7000	B007	32, Nasik
S_4	Julie	Asst	5000	B007	32, Nasik
S_5	Mary	Asst	5000	B005	22,d,Pune

relation that have redundant data have problems called update anomalies which are classified as

1) insertion anomalies
2) Deletion anomalies
3) Modification anomalies

Types of Functional dependency

1) Multivalued dependency

M.D occurs in the situation where there are multiple independent multivalued attributes in a single table.

ex -

car_model	Mf-Yr	color
H001	2017	Gray
H001	2017	Metallic
H005	2018	Blue
H005	2018	Matallic
H010	2019	Green

in this ex mf-yr & color are independent of each other but dependent on car_model i.e two column are said to be multivalue dependent on car-model

car_model → mf-year
car-model → color

2) Trivial functional dependency

Trivial dependency is set of attributes which are called trivial if the set of attributes are included in that attribute

$x \to y$
is a trivial functional dependency if y is subset of x.

emp_id	emp_name
1	x
2	y
3	z

{emp_id, emp_name} → emp_id is trivial functional dependency as emp_id is subset of above set.
if we know value of emp_id, emp_name we can get value of emp_id uniquely

3) Non trivial dependency

Non trivial dependency occurs when A → B holds true where B is not subset of A.
In relationship B is not subset of attribute A, then it is considered as a non-trivial dependency.

company	CEO	Age
Microsoft	Satya Nadella	51
google	Sundar Pichai	46
Apple	Tim Cook	57

CEO name is not subset of comp, hence its a non-trivial dependency

4) Transitive dependency
A transitive is a type of functional dependency which happens when t is indirectly formed by two functional dependency.

OR

When an Indirect relationship causes functional dependency it is called T.D

if $P \to Q$ & $Q \to R$
then $P \to R$ is transitive depen.

Trivial	Non-trivial	Semi non-trivial
$A \to A$	$A \to B$	$AB \to BC$
$AB \to A$	$A \to BC$	
$x \supseteq y$	$AB \to CD$	
x is superset of y	$x \cap y = \phi$ no common in both x & y	$x \cap y \neq \phi$ some thing is common in both x & y

Scanned by CamScanner

Page:
Date: / /

Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

ex -

SID	Sname	Cid	Cname	FID	Fname	Salary
1	ABC	C_1	DBMS	F_1	John	30K
2	CD	C_2	Java	F_2	Peter	40K
3	EF	C_1	DBMS	F_1	John	30K
4	GH	C_1	DBMS	F_1	John	30K

Insertion anomaly
- If new course is started we can not add data directly without any info of student. as SID is p.K we can not leave that field blank.

Deletion anomaly
- If we want to delete the record of SID = 2 then we will loose complete data of course & faculty, suppose only one student is register for particular course then all data will get deleted.

Updation anomaly
If we want to update salary of ~~emp~~ FID=1 to 40K then it will get updated in ~~not of~~

number of rows in which data is repeated, it would take much time but technically it should be updated once only.

one Solution to this is can be divide table in

SID	Sname

CID	Cname

FID	Fname	salary

First Normal Form

As per the rule of first normal form an attribute of a table cannot hold multiple values.

Stud

RollNo	Name	course
1	A	C/C++
2	B	Java
3	C	C/DBMS

in this table roll no 1 & 3 have course more than one which violating rule.

to convert it in 1NF we have following solution

soln 1

RollNo	Name	course
1	A	C
1	A	C++
2	B	Java
3	C	C
3	C	DBMS

P.K will be Rollno + course

composite key

Sol^n 2

Rollno	Name	course1	course2
1	A	C	C++
2	B	Java	Null
3	C	C	DBMS

P.K → Rollno

problem of this solution is if one student is registered for 4 course so for one student only we need to add column course 4 other student value for that course is NULL which is not good representation

Sol^n 3

Divide table in two

Rollno	Name
1	A
2	B
3	C

Base table

PK → Rollno

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

Refrencing table

PK → Rollno + course

Second Normal Form

For a table to be in 2NF it must satisfy two conditions

1) Table should be in 1NF
2) There should not be ~~no~~ partial dependency

we already know about dependency which lets take example

stud-id	name	~~rollno~~	branch	add

here stud_id is primary key to fetch any row of the data. like

we can get the name of stud, branch of stud with its stud-id this is dependency.

what is partial dependency

lets consider table- Student

stud_id	name	reg-no	branch	add
1	A	CSE-18	CSE	X
2	A	IT-18	IT	Y
3	B	CSE-18	CSE	Z
4	C	CSE-18	IT	W

PK → stud_id

lets consider another table subject

sub-id	sub_name
1	Java
2	C++
3	Php

PK → Sub_id

Now consider the score table which store marks of stud in perticular subject will also have name of teacher for that subject

Score

Score_id	Stud_id	Sub_id	marks	teacher
1	10	1	70	Java teacher
2	10	2	75	C++ teacher
3	11	1	80	Java teacher

Stud_id & sub_id is added to know which stud score in which subject.

P.K → Stud_id + Sub_id

if Stud_id = 10 marks to be known we require subject_id (for which subject)

if we ~~requ~~ provide sub_id then stud_id is required (for which stud)

Hence PK → Stud_id + Sub_id

in score table teacher column is dependent on subject
Java – Java teacher, C++ – C++ teacher
as PK for score is stud_id + sub_id

teachers name dependent on sub_id

it has nothing to do with ~~stud_id~~

this is partial dependency

Where an attribute in a table depends on only a part of P.K & not on whole key.

Now, to remove Partial dependency remove teacher column from score table & add in ~~teacher~~ subject table

Sub_id	Sub_name	teacher
1	Java	Java teacher
2	C++	C++ teacher
3	Php	php teacher

Now Score table is in 2NF with no partial dependency

Score_id	Stud_id	Sub_id	marks
1	10	1	70
2	10	2	75
3	11	1	80

Third Normal form

For a table in third normal form

1) It should be in 2NF
2) It should not have transitive dependency.

Consider a table score

score_id	stud_id	sub_id	marks	exam_name	total_marks

P.K → stud_id + sub_id

exam_name depends on stud_id & sub_id

e.g - mech engg. student have workshop
but
comp. sci student dont have

some subject have practical exam some dont have.
so,
exam_name is dependent on both stud_id & sub_id

Total_marks depends on exam_name as exam changes total_marks changes (practical - 40/30 theory - 60/70)

exam_name is a non-prime attribute (attribute not in PK)

& total_marks depends on exam_name which is ~~not~~ a non-prime attribute.

i.e the defn of transitive dependency

when non-prime attribute depends on other non-prime attribute rather than depending upon prime attribute or primary key then its a T.D.

To remove transitive dependency

remove the column exam_name, total marks & put them in seperate Exam table & use exam_id when required

exam_id	exam_name	total_marks

Boyce - Codd Normal Form

For table in boyce codd normal form

1) It should be in Third normal form
2) for any dependency A → B,
A should be super key.

(2 means in A → B, A cannot be non-prime attribute, if B is a prime attribute)

Consider a example

stud_id	subject	professor
101	Java	P.Java
101	C++	P. C++
102	Java	P. Java2
103	C#	P. C#
104	Java	P. Java

Primary key → studid + subject

One ~~stu~~ professor teaches only one subject but one subject is taught by two different prof.

hence there is dependency in both subject & professor.
subject depends on prof. name.

Above table satisfy 1NF as ~~there~~ ~~its~~ all values are atomic.
It also satisfy 2NF as there is no partial dependency.

There is no transitive dependency
hence table is also in 3NF

But is is not in BCNF, lets
check how,
P.K is stud_id + Subject
means stud id & subject is
prime attribute

in table we have dependency
proff. → Subject

Subject is Prime attribute &
proff. is non prime attribute
which is not allowed in BCNF.

To make this table in BCNF
we can divide table is two table
Student table & Professor table

student table (stud_id , p_id)
professor table (p_id, prof, Subject)

Fourth Normal Form (4NF)

For a table to be in 4NF
it should be
1) It should be in BCNF
2) Table should not have any
multi-valued dependency.

Multi valued dependency
A table to have multivalued
dependency, folln condn are true

1) for A → B, if for single value of A
multiple value of B exists

2) A table should have atleast 3
column.

3) For relation R(A,B,C), if there is a
multi valued dependency beth A & B
then B & C should be independent
of each other.

example

S_id	course	hobby
1	Sci	cricket
1	Math	Hockey
2	C#	Cricket
2	Php	Hockey

here student with S_id 1 will give two
more records as one student have
two hobbies hence along with both course
hobbies should be specified

s_id	course	hobby
1	sci	cricket
1	Math	Hockey
1	sci	Hockey
1	Math	cricket

course & hobby are independent of each other.

To remove multivalue dependency & convert table in 4NF we can divide table in two.

course table

s_id	course
1	Sci
1	Math
2	C#
2	Php

Hobby table

s_id	hobby
1	Cricket
1	Hockey
2	Cricket
2	Hockey

Fifth Normal form 5NF

For a table in 5NF it should be

1) It should be in 4NF
2) It we decompose table to eliminate redundancy & anomaly & when we rejoin the decomposed tables, we should not loose original data or new record should not added/arise.

OR

2) Joining two or more decomposed table should not lose records nor create new records.

consider a table

Seller	Company	Product
A	Coca Cola	Thumbs Up
B	Unilever	Ponds
B	Unilever	Axe
B	Unilever	lakme
C	P4G	Vicks
C	pepsico	pepsi

Now decompose above table in two

1.

Seller	Product
A	ThumbsUp
B	Ponds
B	Axe
B	Lakme
C	Vicks
C	pepsi

2.

seller	company
A	coca cola
B	Unilever
C	P4G
C	pepsico

Page:
Date: / /

if we rejoin the table (1) & (2) on seller, we get the following table

Seller	company	product
A	coca cola	thumbs up
B	Unilever	Ponds
B	Unilever	Axe
B	Unilever	lakme
C	P & G	vicks
C	P & G	Pepsi
C	pepsico	Vicks
C	pepsico	pepsi

Above two rows are getting added extra while joining so we need to add one more table in 1st decomposition as

company	product
coca cola	Thumbs up
Unilever	Ponds
Unilever	Axe
Unilever	lakme
pepsico	pepsi
P & G	Vicks

On the basis of this above table if we rejoin the table there will not be duplicate extra row so that table will be in 5NF

Decomposition

Decomposition in DBMS removes redundancy, anomalies & inconsistencies from a database by dividing table into multiple tables

Types of decomposition

1) lossless join decomposition.
2) Lossy join decomposition.

1) lossless -
Consider a relation R which is decomposed into sub relations $R_1, R_2, \ldots R_n$, this decomposition is lossless when we join these sub relations result in same relation R.

e.g - R(A,B,C)

A	B	C
1	2	1
2	5	3
3	3	3

$R_1(A,B)$

A	B
1	2
2	5
3	3

$R_2(B,C)$

B	C
2	1
5	3
3	3

if we do natural join $R_1 \bowtie R_2$ then Relation is

A	B	C
1	2	1
2	5	3
3	3	3

relation is same as original then we can say that its a lossless decomposition.

2) lossy decomposition

Consider a relation R which is decomposed into $R_1, R_2 \dots R_n$, this decomposition is lossy when we we join this sub relation, does not result same relation R.

e.g R(A, B, C)

A	B	C
1	2	1
2	5	3
3	3	3

A	C
1	1
2	3
3	3

B	C
2	1
5	3
3	3

A	B	C
1	2	1
2	5	3
2	3	3
3	5	3
3	3	3

relation is not same as original then we can say that its a lossy decomposition

Page:
Date: / /

Dependency Preservation

Functional dependencies for a relational schema R are said to be preserved after a decomposition into some sub relational schema R_1 & R_2 we are able to obtain all the Functional dependencies, as were present initially from these sub rellational schemas

ex - 1

$R(ABCDE)$

$F = \{AB \rightarrow CD,\ C \rightarrow D,\ D \rightarrow E\}$

$D = \{ABC,\ CD,\ DE\}$

$ABC \downarrow AB \rightarrow C$, $CD \downarrow C \rightarrow D$, $DE \downarrow D \rightarrow E$

$AB \rightarrow CD$ (By transitive property)

As, we are able to obtain all f.d then its dependency preserving

ex-2

$R(ABCDEG)$

$F\{AB \rightarrow C,\ AC \rightarrow B,\ AD \rightarrow E,\ B \rightarrow D,\ BC \rightarrow A,\ E \rightarrow G\}$

$D = \{ABC,\ ACDE,\ ADG\}$

$ABC \downarrow$

$AB \rightarrow C$

$AC \rightarrow B$

$BC \rightarrow A$

$ACDE \downarrow$

$AD \rightarrow E$

But we are unable to obtain
$B \to D$ & $E \to G$

hence its not dependency preserving

Scanned by CamScanner

Dependency Preserving Decomposition

let relation R decomposed into $R_1, R_2 \cdots R_n$ & FD set F of subrelation as $F_1, F_2 \cdots F_n$ resp.

If $[F_1 \cup F_2 \cup F_3 \cup \cdots F_n] = F$

then dependency preserving decomposition.

If $[F_1 \cup F_2 \cup \cdots F_n] \subset F$

then not dependency preserving decomposition

R(F)

$R_1(F_1)$ $R_2(F_2)$

R(F)

ex-1 R(ABC)

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

i) $\{AB, BC\}$

$R_1(AB)$	$R_2(BC)$
$A \to B$	$B \to C$
$B \to A$	$C \to B$

$B^+ = BCA$

$C^+ = ABC$

as $C \to B$ & $B \to A$ then by transitive prop $C \to A$ then its a ~~transitive~~ dependency pres.

ex - 2

R(ABCDE)

F = {A → B, B → C, C → D, D → BE}

D → B
D → E

Decomposed in {AB, BC, CD, DE}

R1	R2	R3	R4
AB	BC	CD	DE
F1 A → B to check for B → A make a closure of B	B → C C → B F2	C → D D → C F3	D → E F4

$B^+ = BCDE$ $C^+ = CDBE$ $D^+ = DBEC$

$\{F_1 \cup F_2 \cup F_3 \cup F_4\} = F$

A → B, B → C, C → D, D → E
for D → B consider
C → B & D → C from F_2 & F_3
resp. then with transitive rule
D → B & ~~de~~ ~~D → B~~ then check for
D closure from $F_1 F_2 F_3$ & F_4
$D^+ = BCDE$

then we can say that its dependency preserving

Page:
Date: / /

ex-3

$D \to B$
$D \to E$

R(ABCDE)

$\{A \to B,\ B \to C,\ C \to D,\ D \to BE\}$

$\{AB, BD, DE, BC\}$

AB	BD	DE	BC
$A \to B$	$B \to D$	$D \to E$	$B \to C$
	$D \to B$		$C \to B$

✓ from closure

for $C \to D$

$C \to B \quad B \to D$ then $C \to D$

So its F.D preserving

ex-4

R(ABCD)

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

$AB \to C \quad AB \to D$

Decomposed into $\{ABC, BCD, AD\}$

ABC	BCD	AD
$AB \to C$	$BD \to C$	$D \to A$
F_1	F_2	F_3

$B^+ = B$

$C^+ = C$

$D^+ = AD$

$BC^+ = BC$

$CD^+ = CDA$

$BD^+ = BCDA$

$AB \to C$ ✓, $AB \to D$ ✗, $D \to A$ ✓

So its not D.P