DBMS

MODULE 5
Relational Database Design

Written & Compiled by,
Prof.Priyanka Bhilare

5. Relational DB Design

Redundancy:-

Redundancy means duplication of data i.e. same piece of information is stored in tables for multiple times. (When same data is stored multiple times unnecessarily in DB).

eg. Consider following table student.

Roll.no	name	age	branchid	brname	hodcnct	HoD name
1	A	18	101	Comp	123	ABC
2	B	20	102	I.T	456	XYZ
3	C	19	103	EXTC	8910	PQR
4	D	20	104	mech	569	LMN
6	E	21	104	mech	569	LMN
7	F	19	105	M.E.I.T.	223	EFG
8	G	18	105	M.E.I.T.	223	EFG
9	H	19	101	Comp	123	ABC

All the info. related to students of the college is collected in one table. The above table is not normalized.

Anomalies in DBMS:-

There are three types of anomalies occur when the DB is not normalized. The problem with above table is that as new student gets added in the table for same branch all the info. related to the same branch (branchid, brname, hod cnct, hodname) is repeated. This is called redundancy.

Scanned by CamScanner

1) Insertion Anomaly :-

Insertion anomaly says certain data cannot be inserted into the DB without the presence of other data.

eg. new student admitted to the college and currently not assigned to any branch, then we would not be able to insert the data into the table if branchid field doesnot allow the null values.

2) Deletion Anomaly :-

It says that, if we delete some unwanted data it causes deletion of some wanted data.

eg:- Suppose, if at any point of time the college closes the dept. M.E.I.T. then deleting rows that are having branchid as 105 would also delete the info. of students as students are assigned to this dept.

3) updation / modification Anomaly :-

It says, if we want to update the single piece of data, then changes must be reflected to all the other places of data.

eg:- If hod is changed from 'LMN' to 'UVW' for mech dept. we have to update the name of the hod in all the places. If we forget to change the name in any of the places, conflict may arise having two HOD's for the mech dept.

Scanned by CamScanner

To avoid above anomalies, normalization is used.

Normalization :-

Normalization is the process of organizing data in DB to avoid data redundancy, and three types of anomalies.

Functional Dependency :-

Functional Dependency in the DB serves as the constraint between two sets of attributes.

In a particular relation (R) attribute (Y) is functionally dependent on attribute (X), if every value in X of relation has exactly one value of Y in the given relation. The functional dependency is represented by,

$$x \longrightarrow y$$

known as determinant — functionally dependent

which implies that Y is functionally dependent on x or we say that x attribute functionally determines the attribute Y.

Functional Dependency Chart :-

It is the graphical representation of function dependencies among attributes in any relation.

1) Find out Primary key attributes.
2) Make a rectangle & write all primary key attributes inside it.

Scanned by CamScanner

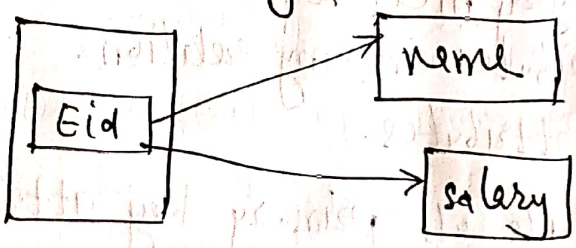
3) Write all non-prime key attributes outside the rectangle.
4) Use arrows to show functional dependencies among attributes.

eg: Consider the eg. employee

Eid	name	Salary
1	Aadi	16000
2	Manoj	16000
3	Sandip	10000
4	Vikas	10000
5	Manoj	16000

In above relation eid is P.K. Suppose you want to know the name & salary of any emp, if you have eid of that emp, then you can easily find info. of that emp. So name & salary attributes depend upon eid attribute.
Here, x is (Eid) and y is (name, salary)
The determinant is Eid.
Suppose x has value 5 then y has (Manoj, 16000) which gives us unique value. therefore x uniquely determines y.
∴ $x \rightarrow y$ where x is Eid & y is (name, salary)
on other hand, if we say salary → name this cannot be F.D. as for salary 16000 we have 2 diff^n values (Aadi, Manoj).

Functional dependency chart is as below:-



Employee relation.

Scanned by CamScanner

Consider the below example and find out functional dependencies.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_2	c_2	d_3
a_3	b_3	c_2	d_4

$A \rightarrow C$, $D \rightarrow B$ are the functional dependencies.

Types of Functional Dependency :-

1) Partial Dependency and fully functional dependency :-

- **Partial Dependency :-**

Suppose you have more than one attribute in Primary key, let A be the non-prime key attribute and if A is not dependent upon all key prime key attributes then partial dependency exists.

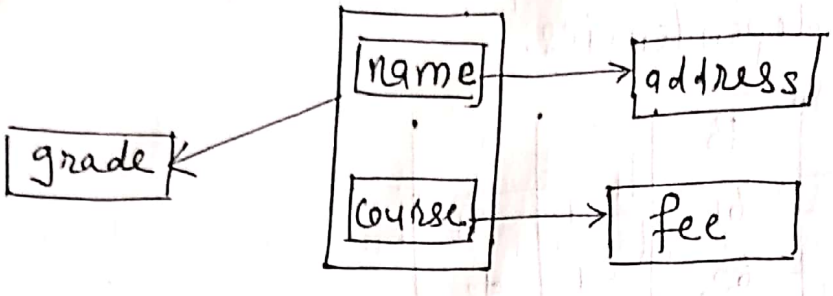
- **fully functional dependency :-**

Let A be the non-prime key attribute and value of A is dependent upon all prime key attributes. Then A is said to be fully functional dependent.

Consider a relation student having prime key attributes (name, course) and non-prime key attributes (grade, fee, address)

name	address	course	fee	grade
p	Andheri	ADBMS	700	A
q	Parel	DBMS	800	B
r	Dadar	UP	500	A
s	Thane	DBMS	800	A
t	Andheri	TCS	500	B
T	Virar	TCS	500	B

Scanned by CamScanner



As shown in fig. address, fee are partially dependent because you can find the address of the student by his name and fee of any course by name of the course.

Grade is fully functionally dependent because you can find the grade of any student in a particular course if you know name of the student and course name.

2) Transitive and non-transitive dependency :-

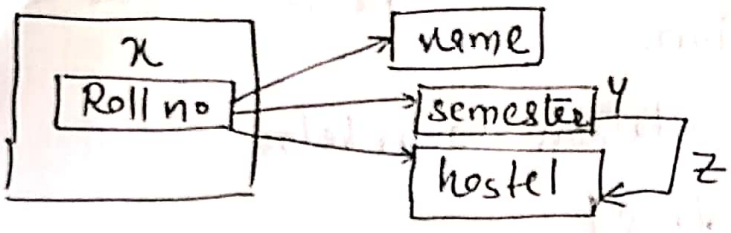
- **Transitive Dependency :-**

It occurs due to dependency between non-prime key. Suppose in a relation (R), $x \rightarrow y$ (y depends upon x), $y \rightarrow z$ (z depends upon y) then $x \rightarrow z$ (z depends upon x). Therefore, z is said to be transitively dependent upon x.

- **Non-Transitive Dependency :-**

Any functional dependency which is not transitive is known as non-transitive dependency. consider, a relation student having prime key attribute (roll no) & non-prime key attribute (name, semester, hostel). for each semester there is diff^n hostel

Scanned by CamScanner



Here, hostel is transitively dependent upon rollno. semester of any student can be find by his roll no. Hostel can be find out by semester of student. Here name is non-transitively dependent upon roll-no.

3) Single valued - multivalued dependency :-

Single value: In any relation R, if for particular value of x, y has single value then it is known as single valued dependency.

multivalued dependency :-

In any relation R, if for a particular value of x, y has more than one value, then it is known as multivalued dependency. It is denoted by $X \rightarrow\rightarrow Y$.

Rules :-

1. A table should have atleast 3 columns for it to have a multidependency.
2. For a relation R(A, B, C), if there is a multivalued dependency between $A \rightarrow\rightarrow B$ and $A \rightarrow\rightarrow C$, then B & C should be independent of each other.

Scanned by CamScanner

Consider, the relation shown below :-

ID	Teacher	class	days
11	Sam	Comp	3
12	John	Comp	6
11	Sam	EXTC	3
12	John	Mech	5
13	Mick	Mech	2

Table prof.

There is a multi-valued dependency (mvd) between Teacher and class because a ~~teacher~~ class can be taken by more than one ~~class~~ teacher. There is another (MVD) betn class and days because a class can be on more than one day.

There is a single valued dependency between ID and teacher because each teacher has a unique ID

✓ ~~Teacher~~ → ~~class~~ class →→ Teacher

✗ ~~class~~ → ~~days~~ class →→ days

✓ ID → ~~Teacher~~

fig. FD chart.

Scanned by CamScanner

Normalization :-

⊛ Prime Attribute :-

An attribute which is a part of the candidate key is known as prime attribute.

⊛ Non-prime Attribute :-

An attribute which is not a part of the prime key is said to be a non-prime attribute.

1) First Normal Form :- (1NF).

As per the rule of the first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values (single)

2) Second Normal Form :- (2NF)

A relation is in 2NF if it is in 1NF and all non-primary key attributes must be fully functionally dependent upon primary key attribute.

A relation is in 2NF iff it has no partial dependency.

3) Third Normal Form :- (3NF)

A relation must be in 2NF, and there is no transitive dependency for non-prime key attributes.

Scanned by CamScanner

(4) BCNF (Boyce-Codd Normal Form)

BCNF is a strict format of 3NF. A relation is in BCNF if and only if all determinants are candidate keys.

(5) Fourth Normal Form (4NF) :-

A relation will be in 4NF if it is in BCNF and has no multivalued dependency.

Consider the following eg.

name	address	course	grade	fee	sem	hostel	duration
XYZ	dadar	C1	A+	9000	VI	H7	10
		C3	B1	8000	III	H3	11
PQR	Parel	C2	B1	5000	V	H4	06
ABC	~~mumbai~~ Andheri	C2	B1	5000	V	H5	09
		C1	A1	9000	VI	H3	09

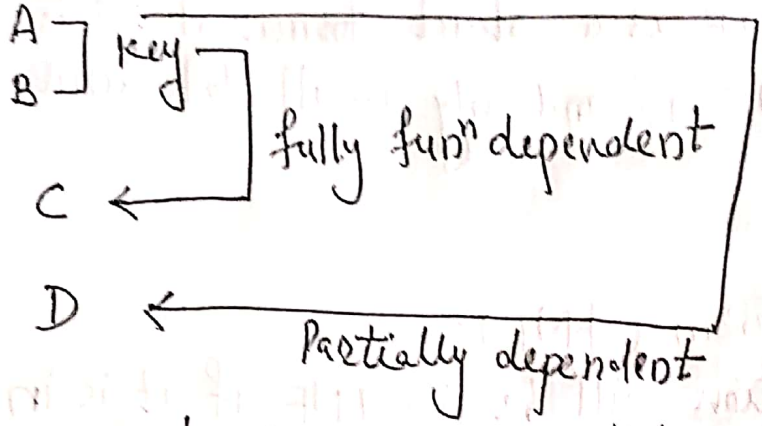
Table: student

(1) 1NF:- Candidate key {course, name}

name	address	course	grade	fee	sem	hostel	Duration
XYZ	dadar	C1	A+	9000	VI	H7	10
XYZ	dadar	C3	B1	8000	III	H3	11
PQR	parel	C2	B1	5000	V	H4	06
ABC	Andheri	C2	B1	5000	V	H5	09
ABC	Andheri	C1	A1	9000	VI	H3	09

Scanned by CamScanner

② 2NF :- (A,B) is a key.



then create two separate tables (A*,B*,C) and (A*,D)
* represents Primary key.

Here,

{name → address}, {course → fee, sem} show Partial dependency where

{name, course} → {grade, duration, hostel} represents fully F.D.

name	address
XYZ	dadar
PQR	parel
ABC	andheri

C.K. {name} ①

Course	Fee	sem
C1	9000	VI
C2	5000	V
C3	8000	III

C.K. {course} ②

name	course	grade	duration	hostel
XYZ	C1	A+	10	H7
XYZ	C3	B1	11	H3
PQR	C2	B1	06	H4
ABC	C2	B1	09	H5
ABC	C1	B1	09	H3

C.K. {name, course}

③

Scanned by CamScanner

3 3NF :-

$x^* \to y \to z$ converted to $\{x^*, y\}$, $\{y^*, z\}$

* shows Primary key.

Here, in first & third table we don't have transitivity. but in second table we have. (keep 1,3 table as it is)

$\therefore$ {course → sem}
{sem → fee}

we have two separate tables,

course	sem
C1	VI
C2	V
C3	III

4

C.K. {course}

sem	fee
VI	9000
V	5000
III	8000

5

C.K. {sem}

4 BCNF :-

if $x \to y$, then x must be a primary key.

now in 1st, 4th, 5th tables we satisfy BCNF criteria

as {name → address} ⇒ name is P.K.
{course → sem} → course is P.K
{sem → fee} — sem is P.K

but in the 3rd table, we have hostel → grade F.D. where hostel is not P.K.

Scanned by CamScanner

∴ create separate table,

hostel	grade
H7	A+
H3	B1
H4	B1
H5	B1

C.K. {hostel} (6)

name	course	duration
XYZ	C1	10
XYZ	C3	11
PQR	C2	06
ABC	C2	09
ABC	C1	09

C.K. {name, course} (7)

(5) 4NF :-

{1,4,5,6,7} tables will be as it is, in 7th table

We have multivalued dependency.

{name →→ course}
{name →→ duration}

∴

name	course
XYZ	C1
XYZ	C3
PQR	C2
ABC	C2
ABC	C1

C.K. {name, course} (8)

course	duration
C1	10
C3	11
C2	06
C2	09
C1	09

(9) C.K. {course, duration}

∴ final tables are {1, 4, 5, 6, 8, 9}.

Scanned by CamScanner