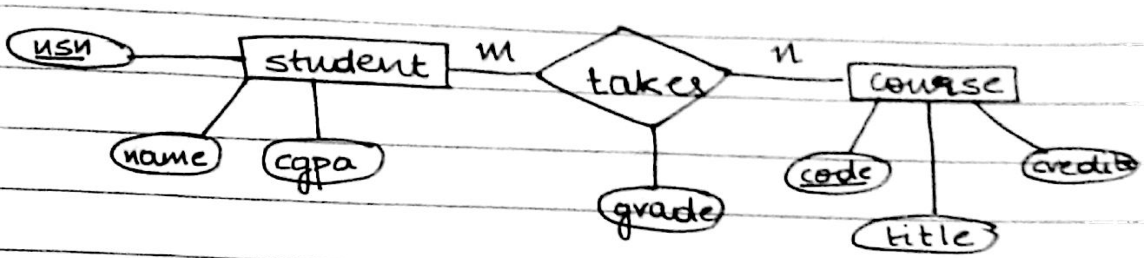
Unit - 4

Database Design



DENORMALIZED FORM → Joins are expensive → This could be used

student_course

usn	name	cgpa	grade	code	title	credits
1	abc	9	A	C1	T1	4
1	abc	9	S	C2	T2	4
2	dc	9.5	A	C1	T1	4
3	xy			C3	T3	4

Redundancy

Problematic insertions

Problems with Redundancy (Anomalies)

↳ insert

* insertion of a new course not yet taken
* insertion of a new student who hasn't yet taken any course

→ deletion

200	abcd	7.1	'C'	C10	T10	5

↳ On deletion, course details are also lost

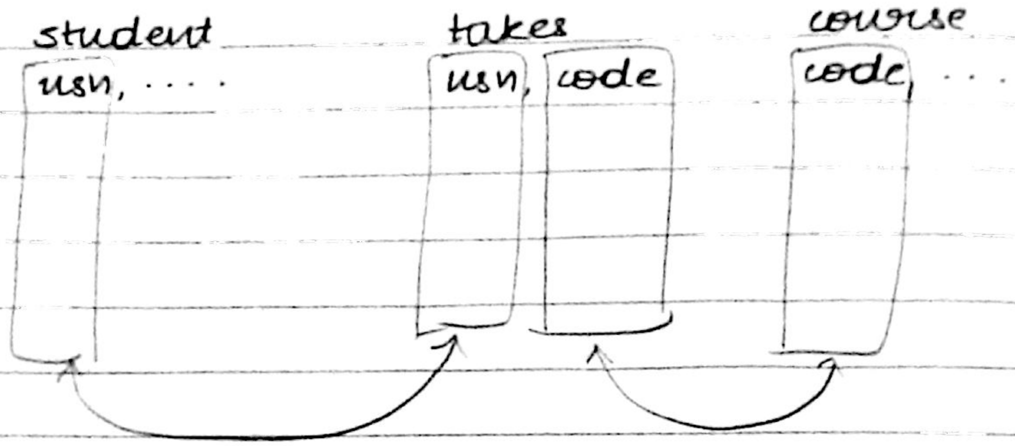
* problem if only 1 student has taken a course

(ER diagrams guide to create a good database)

∴ Informal guidelines → 3 table design is better than 1

→ updation

Inconsistency might persist - updation needs to be done with care.



NOT REDUNDANCY

⟶ Required for relation of data.

Informal Guidelines

1. No Redundancy
2. Reducing null values
3. No spurious tuples (due to joins)
4. clear semantics of attribute

Functional Dependencies

$X \longrightarrow Y$

$\Downarrow$

Y (set) is functionally derivable from X (set)

* constraints applicable to table

$t1(X) = t2(X)$

$t1(Y) = t2(Y)$

wherever the value of X repeats in tuple 1, it has to repeat in tuple 2.

From student_course,

usn → name
(derives)

usn → cgpa

code → title

code → credits

usn, code → grade

A	B	C
1	2	[illegible]
4	[illegible]	[illegible]
7	[illegible]	

[illegible] Violation

✓ B → C
no value
of B repeated

Attributes ~~clou~~ closure → which attributes are derivable from another attributes (from functional dependency)

$usn^+ = \{name, cgpa, usn\}$

$code^+ = \{code, title, credits\}$

$(usn, code)^+ = \{usn, name, cgpa, code, title, credits, grade\}$

↓
All attributes derivable from each of them + derivable from their combination

→ All attributes of table → Hence (usn, code) is a candidate key

GATE-1999
let R = (A, B, C, D, E, F)

FD: $C \rightarrow F$, $E \rightarrow A$, $EC \rightarrow D$, $A \rightarrow B$

which is a key for R?

a) CD b) EC c) AE d) AC

Solⁿ: Attributes not in RHS must by in key.

$C^+ \Rightarrow \{C, F\}$

$E^+ = \{A, B, E\}$

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

key → EC

GATE - 2000

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

which functional dependency is satisfied?

a) $XY \rightarrow Z$ & $Z \rightarrow Y$

✓ b) $YZ \rightarrow X$ & $Y \rightarrow Z$

c) $YZ \rightarrow X$ & $X \rightarrow Z$

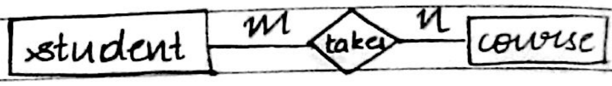
d) $XZ \rightarrow Y$ & $Y \rightarrow Z$

GATE - 2005

R = (A, B, C, D, E, H) Key?

FD → $\{A \rightarrow B,\ BC \rightarrow D,\ E \rightarrow C,\ D \rightarrow A\}$

sol^n: AEH / BEH / DEH



student-course (usn, name, cgpa, code, title, credits, grade)

FDs
- usn → name, cgpa
- code → title, credits
- usn, code → grade

candidate key : (usn, code)

Normal form: Based on keys & FD
- 1NF
- 2NF
- 3NF
- BCNF

For measurement of goodness of database structure → Normal forms

1 Normal form : ~~Rule 1~~ All values must be atomic ~~atomic~~

{Nested columns, Nested tables, car → color ↓ (red, blue, black)} ⇒ violation

2NF → No partial key dependencies
(All functional key dependencies must not have a part of the key on the LHS)

↓ Solving by Decomposition

usn → name, cgpa

Take out the RHS attributes from the table & make it a new table.

* student(<u>usn</u>, name, cgpa)
student course1 (<u>usn</u>, <u>code</u>, title, credits, grade)

↓

* course(<u>code</u>, title, credits)
* studentcourse2 (<u>usn</u>, <u>code</u>, grade)

One of them alone cannot derive grade

3NF → Every FD
$X \to Y$

has 1) X is a superkey
$\boxed{\text{OR}}$ 2) Y should be a prime attribute

Part of any of the candidate key
usn, code → prime attributes
Others are non-prime

candidate key ⊆ Super key

Refer Pg 45
⇓
Distinction b/w candidate key & super key

minimal ← (candidate key)
(super key) → not necessarily minimal

The 3 tables satisfy them.

BCNF (Boyce Codd Normal form) → Stricter form of 3NF

Every FD.
$X \to Y$

has

X is a super key (only)
↓
No 2nd cond. present in 3NF.

The 3 tables satisfy it

↑

All non-keys should be derivable from keys.

Example:

R = (A, B, C, D, E, F)

$C \to F$

$E \to A$

$EC \to D$

$A \to B$ → lost

↓

To avoid loss, create R_4(A, B)

Key : EC

1NF ✓

2NF ✗

$\boxed{C \to F}$, $\boxed{E \to A}$ Violation

Take out F from R & new table.

→ R1 (<u>C</u>, F)

→ R2 (<u>E</u>, A)

✗ BCNF ✓ 3NF R3 (B, <u>C</u>, D, <u>E</u>)

Properties of decomposition

→ Lossless join (non-additive join)

→ Functional dependency preservation
↓
violated <u>A → B</u>

check if this is satisfied during decomposition

But if we can go to a higher NF, it's OK

Q. R = { A B C D E F G H I J

Decompose to 3NF.

~~R1 {A, D}~~ R1 {A, D, E}

R2 {B, F} ~~R3' {A,~~

R3 {A, B, C}

R4 {D, I, J}

R5 {F, G, H}

→ ~~123~~ In 3NF

Also in BCNF.

Q. AB → C, CD → E, DE → B

candidate keys: ABD, ADE, ACD

Prime attributes, A, B, C, D, E

R(A, B, $\boxed{C}$, D, E)

X WRONG { R1(A, B, C) / R2(D, E, B)

→ should not be done - keys disturbed

2NF

* { when a functional dependency derives a prime attribute, even if there is partial functional dependency, it is still in 2NF.

∴ 2NF → R(A, B, C, D, E)

→ 3NF also

Q. CARSALE (Car #, Date-sold, Salesperson #, commission %, Discount-amt)

candidate Key derive all attributes

A. 2NF { CARSALE (Car #, Date-sold, salesperson #, Discount-amt)

SALES-COM (salesperson #, commision %)

3NF { DISCOUNT (Date-sold, Discount-amt)

SALES-COM (salesperson #, commision %)

CARSALE (Car #, Salesperson #, Date-sold)

BCNF also

Q. BOOK(<u>Title</u>, <u>Author</u>, Type, List-price, Author_affil, Publisher)

A. 2NF

F ⟶ ~~Removed for 3NF~~

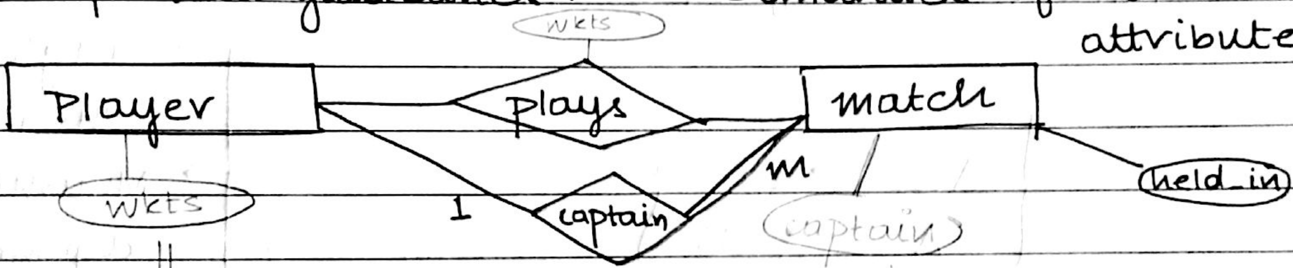
BOOK(<u>Title</u>, <u>Author</u>) ~~List-Price~~)

BOOK-DET(<u>Title</u>, Type, Publisher)

AUTHOR(<u>Author</u>, Author-affil)

3NF: ~~Book~~-Type(<u>Type</u>, List_Price)

<u>Informal guidelines</u>: Semantics of the attribute



⇓ correct on both sides

⟶ Does not belong to match instead relation

<u>Inference Rules for FD</u>

Armstrong's inference rules/ axioms

Given a set of FDs, we can infer additional FDs that hold whenever the FDs in F hold:

(1) Reflexive ⟶ Trivial $Y \subseteq X$

If Y is a subset of X,

$X \rightarrow Y$

(2) Augmentation If $A \rightarrow B$, then

$XA \rightarrow XB$ is also true

(3) Transitive If $X \rightarrow Y$, $Y \rightarrow Z$, then $X \rightarrow Z$

(Used in Attribute closure)

Scanned by CamScanner

IR4 Union If $X \rightarrow A$ & $X \rightarrow B$
then $X \rightarrow AB$

IR5 Decomposing If $X \rightarrow AB$,
then $X \rightarrow A$, $X \rightarrow B$

IR6 Pseudo transitive
If $A \rightarrow B$,
$WB \rightarrow Y$, then

$WA \rightarrow Y$

WRONG $\left\{ \begin{array}{l} AB \rightarrow C \text{ then} \\ A \rightarrow C \\ B \rightarrow C \end{array} \right.$ X

Minimal set of FDs (minimal cover)

$$F = \left\{ \begin{array}{l} AB \longrightarrow C \\ A \longrightarrow B \end{array} \right\}$$

→ single attribute in RHS
→ Minimal set of dependencies required
→

Step 1 → canonical form of dependencies
RHS must have single attribute
If not, decompose

Step 2 → LHS has multiple attributes, reduce them.

For $AB \rightarrow C$ can we have
$A \rightarrow C$ or $B \rightarrow C$?
By Pseudo-transitivity. $AA \rightarrow C$
$\Rightarrow A \longrightarrow C \equiv AB \rightarrow C$

Scanned by CamScanner

Final minimal cover : $A \rightarrow B$

$A \rightarrow C$

Step 3 → Removal of redundant FDs

Q_2

$$F = \begin{Bmatrix} A \rightarrow BC \\ B \rightarrow C \end{Bmatrix}$$

Step 1 → canonical form

$A \rightarrow B$ ①

$A \rightarrow C$ ②

$B \rightarrow C$ ③

Step 2 → No LHS with multiple attributes

Step 3 → Redundancy

From ① & ③, $A \rightarrow C$ (inferable)

∴ Minimal cover : $A \rightarrow B$

$B \rightarrow C$

$A \rightarrow C$ is inferable from these 2. All FDs must be inferable from minimal cover.

Q3. $A \rightarrow BCE$

$AB \rightarrow DE$

$BI \rightarrow J$

Step 1:

$$\left.\begin{array}{l} A \rightarrow B \\ A \rightarrow C \\ A \rightarrow E \\ AB \rightarrow D \\ BI \rightarrow J \\ AB \rightarrow E \end{array}\right\} \text{Decomposition}$$

Step 2:

$A \rightarrow B$

$$\left.\begin{array}{l} AB \rightarrow D \equiv A \rightarrow D \\ A \rightarrow B \ \& \ AB \rightarrow E \equiv A \rightarrow E \end{array}\right\} \text{Pseudo transitivity}$$

$$\left.\begin{array}{l} A \rightarrow B \\ A \rightarrow C \\ A \rightarrow E \\ A \rightarrow D \\ BI \rightarrow J \end{array}\right\} \text{Minimal cover}$$

Step 3 → x

$A \rightarrow BCDE$

$BI \rightarrow J$

Q4. $B \rightarrow A$

$D \rightarrow A$

$AB \rightarrow D$

A. Step 1 ✓

Step 2: $B \rightarrow A$

$AB \rightarrow D$

$\Rightarrow B \rightarrow D$

Step 3: $B \rightarrow A$

$$\left.\begin{array}{l} D \rightarrow A \\ B \rightarrow D \end{array}\right\} \text{infers } B \rightarrow A$$

∴ Minimal cover: $D \rightarrow A$, $B \rightarrow D$

FD → Equivalence

Closure F^+

$F = \{A \to BC,\ B \to A,\ AD \to E\}$

$G = \{A \to ABC,\ B \to A,\ BD \to E\}$

$A \to C$ → Member in the closure of F

Gives exhaustive set of dependencies implied by F.

If dependencies in F ~~is~~ can be inferred from G → G covers F

If F covers G & G covers F → Equivalent

Finding Equivalence :

→ Minimal cover of F & G (OR)

→ Find out attribute closures

Method 1 :

MC for F

(1) $A \to B$

$A \to C$

$B \to A$

$AD \to E$

(2) —

(3) $A \to BC$

$B \to A$

$AD \to E$

↑

Equivalent

↓

MC for G

(1) $A \to B$ — 1

$A \to C$

$B \to A$

$BD \to E$ — 2

(2) $BD \to E$,

But in F,

we have $AD \to E$

From 1 & 2,

$AD \to E$

(3) $A \to BC$

$B \to A$

$AD \to E$

Q. $F = \{A \rightarrow B,\ AB \rightarrow C,\ D \rightarrow AC,\ D \rightarrow E\}$
$G = \{A \rightarrow BC,\ D \rightarrow AE\}$

Method 2:

From F, there is no C on LHS. Hence, C^+ is not required.
ly. for G, we need only A^+ & D^+

For F,

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

$B^+ = \{B\} \longrightarrow$ Trivial

$D^+ = \{D, A, C, E, B\}$ — B: From A

For G,

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

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

$$\left.\begin{array}{l} A^+ \text{ of } F \equiv A^+ \text{ of } G \\ D^+ \text{ of } G \equiv D^+ \text{ of } F \end{array}\right\} \text{Equivalent}$$

Q. $F = \left\{ \begin{array}{l} A \longrightarrow ABC \\ B \longrightarrow A \quad BD \longrightarrow E \\ AB \longrightarrow D \end{array} \right.$

(i) Is $A \longrightarrow D$ in F^+?

Sol^n: $B \rightarrow A$

$AB \longrightarrow D$

$\therefore B \longrightarrow D$

$B \longrightarrow D$

$A \longrightarrow ABC$

$\therefore A \longrightarrow D$

(ii) $A \to BE$?

Sol^n

$B \to D$	$B \to E$
$BD \to E$	$A \to B$
$\therefore B \to E$	$\therefore A \to E$

4NF

Multivalued dependencies

emp(empid, dependent, project_no)

Multiple dependents & multiple projects possible for one employee.

id	dependent	p_no
1	shankar	p1
1	suma	p2

(id $\xrightarrow{1:n}$ dependent, id $\xrightarrow{1:n}$ p_no)

Multivalued attributes

1	shankar	p2
1	suma	p1

when we combine many 1:n relationships in one relation → not in ~~4F~~ 4NF.

id $\twoheadrightarrow$ dependent

id $\twoheadrightarrow$ proj_no

4NF → For each MVD, $X \twoheadrightarrow Y$

(1) LHS is a superkey

(OR) (2) $X \cup Y = R$

In table, all 3 columns are the key. $X \cup Y \neq R$, Hence not in 4NF.

Remedy:

Decompose the MVDs into tables.

emp-dep(empid, dependent)
emp-proj(empid, proj-no)
> 2nd condition satisfied.

5NF (Project Join Normal Form)

Join dependency $J(R_1, R_2, \cdots R_n) = R$ (non-additive (lossless) join)

S

sid	pid	pjid
1	p1	pj1
1	p2	pj2
2	p1	pj1 (pj2)

S1

sid	pid
1	p1
1	p2
2	p1

S2

sid	pjid
1	pj1
1	pj2
2	pj1 (pj2)

S3

pid	pjid
p1	pj1
p2	pj2
p1	pj2

↳ spurious tuples generated (1 p1 pj2)

On applying natural join on S1, S2 & S3, we need to get back original table.

S1 * S2 * S3

Joining any number of relations less than the actual number must not give original table.

S1 * S2

sid	pid	pjid	* S3
1	p1	pj1	✓
1	p1	pj2	X
1	p2	pj1	X
1	p2	pj2	✓
2	p1	pj1	✓

= S

Lossless join $(R1 \cap R2)$

$\longrightarrow (R2 - R1)$ OR

$\longrightarrow (R1 - R2)$

$S1 - S2 \Rightarrow$ pid

$S2 - S1 \Rightarrow$ pjid

$\therefore$ sid $\rightarrow$ pid OR

$\underbrace{\text{sid} \rightarrow \text{pjid}}$

this should hold true for table to be in 5NF.

5NF depends on decomposition proposed.