Atomicity, Consistency
ACID
Isolation
Durability

DBMS

DBMS v/s File system

(1) Unstructured — handled by Operating system → heirarcial
(2) File Location
(3) Allow pre-determined access to data
(4) Data Redundancy → duplicate
(5) Inconsistency
(6) Concurrent access not allowed → RR, RW, WD
(7) Integrity
(8) Security → Role Based

Keys | Superkeys

Attributes / fields →

keys

* set of attributes or attribute that uniquely identify each record of Relation.

max SK = $2^n - 1$

* we add any key to sk → also be a super key

Keys in DBMS

- Candidate key - SK whose proper subset is not a SK. $S_2 \subset S_1$ [$S_2 \notin S_1$]
  - Minimal key - CK is min. SK
  - every CK is SK, vice versa is not true
- Primary key : ~~those attributes which are part of candidate key~~ is req to uniquely identifies all attributes in a table
  - minimal set of attribute that

Q. R(A, B, C), A is CK → no of SK?

Ans → [A, AB, AC, ABC]

Functional Dependencies

FD : $X \rightarrow Y$ (determinant X, determines, dependent Y)

if [$t_1 \cdot x = t_2 \cdot x$]
must then $t_1 \cdot y = t_2 \cdot y$

tuples

x	y
1	5
2	3
1	5

Table

Name	Marks	Roll No.	Dept.	Course
a	78	1	CS	C_1
b	60	2	EE	C_2
a	78	3	CS	C_2
b	60	8	EE	C_3
c	80	5	IT	C_2

Type of functional Dependencies

1. Trivial (Always valid)
   $X \rightarrow X$ → Trivial → X can determine itself
   $X \rightarrow Y$ → (R. No., Name) → Name
   if ($Y \subseteq X$)
   then it is Trivial function dependencies
   • No need to check always valid
2. Non-Trivial
   $X \rightarrow Y$ → Roll No. → Name
   if ($X \cap Y = \Phi$)
   → Nothing is common
   • Semi - Trivial → RollNo, Name → Name, Marks
   Need to check
3. ~~Superkey~~ Multi-valued
4. Transitive.

DBMS

Armstrong's Axioms / Inference Rules :—

1. Reflexivity $[X \rightarrow X]$, $[X \rightarrow Y$ if $(Y \subseteq X)]$
2. Transitivity If $(X \rightarrow Y$ & $Y \rightarrow Z)$ then $X \rightarrow Z$ → if this is not true we can't say anything about $X \rightarrow Z$
3. Augumentation : if $X \rightarrow Y$ then $XA \rightarrow YA$
4. Union : if $X \rightarrow Y$ & $X \rightarrow Z$ then $X \rightarrow YZ$
5. Decomposition / splitting : If $X \rightarrow YZ$ then $X \rightarrow Y$ & $X \rightarrow Z$

   If $XY \rightarrow Z$ then $X \rightarrow Z$, $Y \rightarrow Z$
6. Pseudo transitivity : If $X \rightarrow Y$ & $YZ \rightarrow A$ then $XZ \rightarrow A$
7. Composition : if $X \rightarrow Y$ & $A \rightarrow B$ then $XA \rightarrow YB$

Attribute Closure / closure set :—

X^+ → contains set of attributes determined by X

SK: Set of attributes where closure contains all attributes of given relation

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

SK ← CK $ACD^+ = \{A, C, D, B, E\}$

Prime attributes → Part of CK

$\{A, C, D\}$ → any on rhs? If No → No more CK

Q. $R(A, B, C, D, E)$ F.D: $\{A \rightarrow B, D \rightarrow E\}$

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

Normalization in DBMS → Reduce Redundancy / duplicate data

Problems → ① Insertion Anomaly
② Update Anomaly
③ Delete Anomaly

Solution → decompose tables

→ Related to each other through foreign key.

- Online Transaction and Analytical processing

Need → (1) Reduce Redundancy
(2) Save the space
(3) Remove anomalies
(4) Normalise NULL values
(5) Simplify queries
(6) Used in OLAP & OLTP → denormalised
(7) Searching / sorting will be easy
(8)

(OLAP) Online analytical processing

OLTP) Transaction

(1) First Normal form ~~Operate~~

No multivalued attributes

• ER diagram → to Relation schema
• Entity → attributes

Each column should contain atomic values ← First Normal form → must not contain any multi-value attributes or composite attributes

Rules for 1NF:

1. A col should contain value from same domain
2. Each column should have unique name
3. No ordering to rows and col.
4. No duplicates

* 123 → atomic

  1, 2, 3 → not atomic

  CS001 → Not atomic

E.R → relational schema [always in 1NF]

• Separate table for multivalued attributes, and separate column for composite attributes.

(2) Second Normal Form

Rules to check whether R.D is in 1NF.

[1. If it is in 1st normal form

2. No partial Dependency in the relation

$AB \rightarrow C$

$A \rightarrow C$, $B \rightarrow C$ X

[P.D. Proper-subset of CK → determine non-prime attribute]

non-prime attribute → attr. which are not part of ck

{to remove do full dependencies remove P.D.

eg. R(A, B, C, D, E, F)

F.D = {A → B, B → C, C → D, D → E}

$AB\{DF^+ = \{ABCDEF\}$

$\boxed{AF^+} = \{\ \} \Rightarrow A^+ = \{A\ B\ C\ D\ E\}$, $F^+ = \{F\}$] X SK

→ CK

{A, F} → not in rhs

↳ unique ck

prime attr = {A, F}

AF = non-prime attr. = {B, C, D, E}

for check P.D

proper subset of AF = {A, F} → {B, C, D, E}

A → B

⇓

P.D

So, Not in 2nd Normal form

* If CK is in single attr → that relation is in 2nd Normal form.

means no proper subset is possible

* If all R is present in prime attributes → 2nd NF.

↳ There is no non-prime attributes

Drawbacks in 2nd Normal form
→ still have some Redundancy. [there is update anamoly present]

Third Normal form → Remove all anamoly (insert, update & delete)

(1) It is in 2nd NF
(2) No Transitive dependency for non-prime attr.

Transitive depend. → $\boxed{NPA \to NPA}$ $\boxed{PA \to NPA}$ → not in 2NF.

Not able to handle the cases where have multiple CK & all CKs are overlapping

eg. R(A, B, C, D, E, F)
F.D → {AB → CDEF, BD → F}
AB$^+$ = {A, B, C, D, E, F}
SK AB$^+$ = { " }
{A$^+$, B$^+$} ✗→ SK
$\boxed{AB^+ \to CK}$

X → Y → Z
CK/PK — non unique non-prime — non-prime

P.A → {A, B}
NPA → {C, D, E, F}

AB → CDEF
PA → ✗ T.D.
BD → F → T.D
NPA   NPA

~~Hence~~ ~~Not~~
Hence, Not in 3rd Normal form

2 Method:
A table is in 3NF if & only if for each of its non-trivial functional dependency at least one of the following cond. holds:—
1. LHS is SK
2. RHS is prime attributes

BCNF (Boyce - Codd Normal form)
→ If it is in 3NF
→ for each non-trival F.D X → Y
X must be super key or CK

Q. find No. of NF?
R(A, B, C, D, E)
F → {AB → CDE, D → A}
AB$^+$ = {A B C D E}

CK ← AB$^+$ = { }
CK ← DB$^+$ = {
{D$^+$, B$^+$} ✗ SK

A → CDE, A
B → CDE, A, B
{A} ✗ SK
{B}

PA → AB {A, B}
DB {D, B}

PA → {A, B, D}
NPA → {C, E}

BCNF ✗ → AB is SK
D ✗ SK

3NF → AB → PA → ~~CDE~~ ✗ T.D.
D → PA → ✗ T.D.

(1) Dependency Preserving Decomposition → for decomposition it should be — (1) Dep. preserving (2) Loss less

eg: check DP Decomposition (will not take trivial dep.)

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

F : {A → B, B → C, C → D, D → A}

R₁ (A, B, C) | R₂ (C, D, E)

$A^+ = (\cancel{A}BC\cancel{D})$ | A → BC | $C^+ = \cancel{C}DAB$ | C → D
$B^+ = (\cancel{B}CDA)$ | B → CA | $D^+ = D\cancel{A}\cancel{B}C$ | D → C
$C^+ = (\cancel{C}DAB)$ | C → AB | $E^+ = \cancel{E}$ | duplicate
$AB^+ = \cancel{AB}CD$ | ~~AB → C~~ | $CD^+ = \cancel{CD}AB$ ~~(DE → C)~~
$BC^+ = \cancel{BC}DA$ | ~~BC → A~~ | $DE^+ = \cancel{DE}ABC$ ~~(CE → D)~~
 | | $CE^+ = \cancel{CE}DAB$

Trivial

not part of R₂

same

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

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

$\equiv F$ → every FD of F is member of G and vice-versa

check $F_1 \cup F_2$

Mandatory Property

(2) Lossless Join Decomposition (Mandatory)

R = (A, B, C)
1 1 1
2 1 2
3 2 1
4 3 2

R₁ (A, B)
1 1
2 1
3 2
4 3

R₂ (B, C)
1 1
1 2
2 1
3 2

where B is common

R₁ × R₂

R₁ (A, B) R₂ (B, C)
1 2 — 1, 1
2 3 — 2, 3
3 4 — 3, 1

Natural join operation (⋈)

R₁ ⋈ R₂ → R₁.B = R₂.B

R₁ ⋈ R₂ → A B C
1 1 1
1 1 2
2 1 1
2 1 2
3 2 1
4 3 2

✗ not lossless
we must get exactly same relation as R

1. attr. (R₁) ∪ attr (R₂) = attr (R)
2. att (R₁) ∩ att (R₂) ≠ ϕ
not lossless
3. If common attr is SK of atleast one relation then that would be lossless join decomposition.

* Loss less is compulsory but Dependency preserving decomposition is not necessary.
* Sometimes, it is not possible to get dependency preserving decomposition in case BCNF decomp.
* upto 3NF is always possible to get dependency preserving decomposition.

DB
OS → memory mapping, virtual memory
OPPS
CN
COA

→ If R is decomposed into R₁ & R₂ then decomposition is lossless if —
1. att (R₁) ∪ att (R₂) = att (R)
2. att (R₁) ∩ att (R₂) ≠ ϕ
3. att (R₁) ∩ att (R₂) = att (R₁)
or
att (R₁) ∩ att (R₂) = att (R₂)

Canonical Cover / Minimal Cover / Irreducible set of FDs.

F

F' if F' don't have (i) extraneous attr. / redundant attr.

(ii) Redundent FD.

$F_1 = \{A\cancel{B} \to C, \; A \to B\}$

$(A\cancel{B} \to C$ $\quad B \to C\}$

(iii) Remove Redundant FD

$F : \{AB \to C, \; C \to AB, \; B \to C, \; A\cancel{B}C \to A\cancel{C}, \; A \to C$

$AC \to B\}$

Trival

step 1: $\{AB \to C, \; C \to A, \; C \to B, \; B \to C, \; \cancel{ABC \to A}, \; A \to C, \; A\cancel{C} \to B\}$

$AB\cancel{C} \to C$, trivial

step 2: $\{B \to C, \; C \to A, \; \cancel{C \to B}, \; B \to C, \; A \to C, \; C \to B\}$

Removing Redundant

step 3: $\{C \to A, \; B \to C, \; A \to C, \; C \to B\}$

$\{C \to A, \; B \to C, \; A \to C\}$

Convert a relation from 1NF to 2NF

Steps:-

1. Spliting Rule so that in every FD right hand side has single attributes
2. Remove extraneous attr

$F\{A\cancel{B} \to C \quad A \to B\}$
$\hookrightarrow$ B is extraneous bcz $A \to B$

3. Remove redundant FD. / duplicate

* In spliting property we can only split right hand side not left side

Database system → compromised of 2 things

Database
↓
collection of Related data
- Structured
  - ↳ RDBMS → Relation
  - eg. IRCTC university
- Unstructured → eg. Webpage,

DBMS
↓
collection of operations
eg. SQL, MySQL

- OODB
- Rational DB
- Distributed DB
- Hierarchical DB

File System Vs DBMS

File System → Path is necessary to access the file system

DBMS → client-server
1KB ↑ 25GB
- when more than one clients are accessing the data

Key To access each data uniquely.

key / Superkey is an attribute or set of attributes that uniquely identify each record of Relation.

Max superkey → only a theorical concept
(Roll.No, name / dep / course / Marks)

$= {}^5C_1 + {}^5C_2 + {}^5C_3 + {}^5C_4 + {}^5C_5$

$= 31 = \boxed{2^n - 1}$

If (B, C, D) is not a superkey then subset of it cannot be a superkey.

2. Candidate key → SK where proper subsets is not a SK.

eg. If SK - A, AB, AC, ABC, BC

ABC → proper subsets → AB, A, B, C, BC, AC
Here are SK.
so, ABC is not a candidate key.

AC → proper subsets → A, C
is a SK.
so not a candidate key

Minimal key

A → {} so it candidate key.

BC → {B}, {C} ← proper subsets
not a superkey.
so a candidate key.

{A} {BC} → candidate or Minimal key.

functional dependencies
↳ for every same X there should be same Y

X → Y

Roll.No	Name	Marks	Dep.	Course
1	a	78	CS	C_1
2	b	80	EE	C_1
3	a	78	CS	C_2
4	b	60	EE	C_3
5	c	80	IT	C_3
6	d	80	EC	C_2

R.No → Name ✓
Name → Roll.No ✗
R.No → Marks ✓
Dept → course ✗
course → dep. ✗
Marks → dep. ✗
[R.No, Name] → Marks ✓
[Name, Marks] → dep ✓
[Name, Marks] → [Dep., course] ✗

Types →
1. Trivial
2. Non-Trivial
3. Multi-valued
4. Transitive

Attribute closure / closet set (+)

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

FD {A → B, B → C, C → D, D → E}

A → C
A → A
A → D
A → E

A → ABCDE

Similarly,
B → BCDE
C → CDE
D → DE
E → E

X^+ → contains set of attributes determined by X.

↓

set of attributes

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

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

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

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

1. Superkey when FD is given –
set of attributes whose closure contains all relation of given relation

A^+, AD^+, AB^+ → A anything$^+$

So, No. of superkeys possible

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

no. of relation $= 2^4 = 16$

So. 16 → superkeys is possible

2. Candidate key

A^+ → only superkey which is a candidate key

e.g. R(A, B, C, D, E)

FD = {A → B, D → E}

To find sk,
↳ find closure

$A^+ = \{A, B\}$ ×

$BC^+ = \{B, C\}$ ×

SK ← $ABCDE^+ = \{A, B, C, D, E\}$ ✓

CK × $ABDE^+ = \{A, B, D, E\}$ ×

SK ← $ACDE^+ = \{A, C, D, E, B\}$ ✓

CK × contains all attributes ↳ of the relation

SK ← $ACD^+ = \{A, C, D, B, E\}$ ✓

To find is it a CK?

all proper subsets → A, C, D, AC, AD, CD

So it is a candidate key ← None is a SK

Finding of all candidate keys of a relation

Step 1: R(A, B, C, D, E)

F.D → {A → B, D → E}

Step 1: (SK) $ABCDE^+ = \{A, B, C, D, E\}$

Step 2: discard B & E coz we have A & D and with these we can determine B & E

(SK) $A\cancel{B}CD\cancel{E}^+ = \{A, B, C, D, E\}$

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

None is SK: AC^+, CD^+, AD^+, A^+, C^+, D^+ → So, is a candidate key.

Step 4: For check whether there is more CK or not.

Prime Attributes → attributes which are part of CK.

↳ {A, C, D}

If any of these attributes are present on the right side of f. Relation

↓

then there is more CK

↓

else if none of prime attributes are present on right side of func dep. then there is No more CK.

eg 2: R(A, B, C, D)

FD → {A → B, B → C, C → A}

1. (SK) $ABCD^+ = \{A, B, C, D\}$

(A → C) $ACD^+ = \{A, C, D, B\}$

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

$\{A\}^+$, $\{D^+\}$ × SK

↓

{A, D} AD^+ → CK

↓

present on the right side (C → A)

CD^+ → SK

{C}, {D}

CK × SK

prime attributes {A, D, C, B}
↓ on the right side {B → C}

CD⁺
↓
BD⁺ → SK
so, ↳ {B, D} → ✗SK
↓
CK

AD⁺

eg. 3. R(A, B, C, D, E, F)
F.D = {AB → C, C → DE, E → F, D → A, C → B}

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

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

AB → F $AB^+ = \{A B C D E F\}$ (SK)

for (CK) find $\{A^+\} = \{A\}$ → ✗SK
$\{B^+\} = \{B\}$ → ✗SK
so AB → CK

prime attributes = {A, B}

D → A → present on right side

DB^+ → SK
↓
=> {D} → ✗SK
{B} → ✗SK

prime A. = {A, B, D}

for C → B
AC^+ = SK
↳ {A} => ✗SK
{C} = {D, E, A, B, C, F} → ✓SK

AC^+ = ✗CK

$\boxed{C \to SK \to CK}$

P.A = {A, B, D, C}
↓
C → D

$CB^+ = \{C\}$ → SK
✗ CK

CK = {AB, BD, C}

P.A = {A, B, D, C}

Non-P.A = {E, F}

Normalisation
↳ Process of making the table free from insert, update and delete anomaly and save space from redundant data.

Q. check NF:
R(A, B, C, D, E)
FD - {A → BCDE, BC → ACE, D → E}
$ABCDE^+ = \{\ \}$ => $A^+ = \{\ \}$
SK → CK
{} → ✗SK

BC = {ACE, B, D}
↳ SK
↳ {B} → ✗SK
{C} → ✗SK

$\boxed{A = CK, BC = CK}$

BCNF: X must be super key
A → SK
BC → SK
D → ✗SK

No in BCNF

3NF: A ✓
BC ✓
D ✗

2NF: A ✓
BC ✓
D ✓

$\boxed{\text{lecture - 14 \& 15}}$ examples

Q1 → R(A, B, C, D, E)

F [AB → CDE, D → BE] → CK = AB, AD
PA = (A, B, D)
(✗) NPA = (C, E)

	AB	D	
BCNF:	✓	✗	(X is SK)
3NF:	✓	✗	(X is SK (or) Y is PA
(PA) ∪ (NPA) = NPA 2NF:	✓	✗	(

∴ 1NF

check dependency preserving decomposition

ques on lossless join decompos
L19: [Jenny]

R(A B C D E)

F = {A → BCD, B → AE, BC → AED
D → E, C → DE

R_1(A, B) R_2(B, C) R_3(C, D, E)

for R_1

→ A^+ → check in F

A^+ = {~~A~~BCDE} ⇒ $\boxed{A \to B}$

trivial so discard

not part of R_1

B^+ = {~~B~~AECD}

Trivial

not part of R_1

B^+ = {A} → $\boxed{B \to A}$

pick 2 → put here it will become trivial (AB)

for R_2 (BC)

B^+ = {~~BAE~~CD} ⇒ $\boxed{B \to C}$

C^+ = {~~C~~D~~E~~} → { }

2(BC) → will become trivial

for R_3 (C, DE)

C^+ = {~~C~~DE} → $\boxed{C \to DE}$

D^+ = {~~D~~E} → $\boxed{D \to E}$

E^+ = E

duplicate

CD^+ = {~~CD~~E} → ~~CD → E~~

$F_1 \cup F_2 \cup F_3$ = {A → B, B → A, B → C, C → DE
D → E}

Now will check each F.D of F is member of G

for A → BCD ✓ → A^+ = {ABCDE}

B → AE ✓ → B^+ = {ACDE}

BC → AED ✓ → BC^+ = {BCADE}

D → E ✓

C → DE ✓