$^nC_r = \frac{n!}{r!(n-r)!}$

Day: Tashfeen (Keys in DBMS) 22:-2-24 Date: / /

Table/relation

Attributes / fields	Name	Marks	Dept	Course
{Dept, Course} - key	a	78	CS	c_1
{Name, Marks} - not a key (rows / records / tuple)	b	60	EE	c_4
	a	78	CS	c_2
	b	60	EE	c_3
{Name, marks, Dept, Course} - key	c	80	IT	c_7

1) • **Super key** or key same hote hy (theoretical concept)
- Bs uniquely identify krnay.
- super key kay sath any attribute added it will be a super key.

⇒ Maximum Superkey: $2^{n} - 1$ → no. of attributes

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

$5 + 10 + 10 + 5 + 1 = 31$

2) **Candidate keys**:- • Uniquely identify a tuple.
• Strong as primary keys
minimum SK or • SK whose proper subset is not a SK.
• Minimal SK.

⇒ SK ⇒ A, AB, AC, ABC, BC

⇒ if proper subset than

$S_1 = \{1,2,3\}$ $S_2 = \{1,2\}$

$S_2 \subset S_1$

than • $S_2 \subseteq S_1$ ✓
• $S_1 \not\subseteq S_2$ ✓

$S_1 = \{1,2,3\}$ $S_2 = \{1,2,3\}$

$S_2 \subset S_1$

than $S_2 \subseteq S_1$ ✓

$S_1 \subseteq S_2$ ✗

A	B	C
1	1	1
2	1	2
3	2	1
4	2	2

In proper subset, elements are less.

Day: Date: / /

Example:-

	SK	CK
ABC	✓	X
AC	✓	X
A	✓	✓
BC	✓	✓

ABC

A, B, C, AB, AC, BC

AC

A, C

BC

B, C

Minimals:-

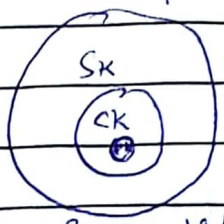
ABC — BC

AB — or C } But both are not SK.

A — or B

So

A, BC



CK

Simple CK

Composite CK.

Prime attributes → part of candidate key.

non-Prime attributes → not part of CK.

3) Primary Keys :-

↓ maybe Composite key

- One CK that has no null values
- short, numeric (alpha), fixed length and never changes

Q:- R (A, B, C), A is CK

No. of SK?

A, AB, AC, ABC

Example:- IBAN

- creates clustered index
- Koi bhi 2 rows mein PK are not same.

Ownership of Enterprise

4) Alternate key

candidate key which is not part of primary key

5) Surrogate key (artificial column added to serve as PK)

- Normally hidden in forms and reports.
- new record add kyo tou we can use surrogate key
- Jb natural primary key tou combining sey no content nai rehti.
- Has artificial values that are meaningless to users.

Day: Date: / /

6) **Unique Keys :-**
- Similar to PK but can accept null value.
- Creates non-clustered index.

7) **Foreign keys :-**
- column of table used to point primary key of another table

Q:- When FK is null?
- Either value in parent table not defined
- or value in parent table is deleted.

Q:- Can a foreign key be an attribute that refers to PK of same relation?
- For recursive relationship

⇒ Attribute Closure / closure set.

Example 1:-

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

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

$E \to E$ $A \to A$ $B \to B$

$A \to C$ $B \to D$

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

$A \to E$ $B \to BCDE$

$A \to ABCDE$ → SK / CK

SK → Jis key closure me saarey attributes aajaye

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

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

$AD^+ = \{A, D, B, C, E\}$ → SK.

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

Example 2:-

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

FD: $\{A \to B, D \to E\}$

$A \to A$ $D \to D$

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

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

$D^+ = \{D, E\}$

Day: Date: / /

Functional Dependency

(attribute of table uniquely identifies another attribute of same table.)

FD: x → y (x determines y)

(determinant) (dependent)

PK → non-key

x = 2 y = 1

- We search value of y in table

x	y
1	1
2	1
3	2
4	3
2	5

⇒ if $t_1.x = t_2.x$ then $t_1.y = t_2.y$

If repitition

- FD are constraints
- FD & key are used to define normal forms for relation

⇒ **Example**

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

R.No → Name ✓

Name → R.No ✗

R.No → Marks ✓

Dept → Course ✗

Marks → Dept ✗

RollNo, Name → marks ✓

Name → Marks ✓

Name, Marks → Dept. ✓

Name, Marks → Dept, Course ✗

⇒ **Good Database design**

- no redundancy (copies of same data)
- no inconsistency (different information in different places)
- no deletion, insertion or update anomalies
- no information lost
- no dependency loss.

- **Insert Anomaly**
- **Update anomaly**
- **delete anomaly**

Exclusive Falcon

A good E-R converted to relational schemas, by default it is 1NF.

Day: Date: / /

Normalization (to ensure good design)

1NF → (1) if every column has **atomic domains/values** i.e.
- only one value in each cell.

Sid	Sname	S_address	PNO
1	Jenny	Haryana, Ind	P_1, P_2
2	Jiya	Punjab, Ind	P_3
3	Payal	Raj, india	P_4, P_5
4	Shanvi	haryana, india	P_7

⇒ This table not in 1NF.
- No composite values
- No multiple values

Convert

(2) Every column should have same domain (datatype change nai krna)

(3) Each column should have unique name.

(4) No ordering to rows and columns

(5) No duplicate rows.

(1)

Sid	SName	State	Country	Phone_No
1	Jenny	Haryana	Ind	P_1
1	Jenny	Haryana	Ind	P_2
2	Jiya	Punj	Ind	P_3
3	Payal	Raj	Ind	P_4

(2)

Sid	Sname	State	Country
1	Jenny	HR	Ind
2	Jiya	Pun	Ind
3	Payal	Raj	ind

Sid	PNO
1	P_1
1	P_2
2	P_3
3	P_4
3	P_5

(3)

not good →

Sid	Sname	State	Country	Ph.1	Ph.2
1				P_1	P_2
2				P_3	Null
3				P_4	P_5

Day: Date: / /

2NF (If CK having only 1 attribute then it is 2NF) ; i.e. !

1. if it is in 1NF
2. No Partial dependency (PA → NPA)

P.D ⇒ Proper subset of CK → non-prime attribute.

a) Sab sy pehly write all attributes

b) Start to discard by seeing relation Prime attributes.

c) Find CK

d) Than see prime attributes are present on right side if yes than more CK.

e) See proper subset of CK

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

F.D = {A→B, B→C, C→D, D→E}

(1) Attribute closer.

$AB\cancel{CDEF}^{+}$ = {ABCDEF}

CK ← AF^{+}

A^{+} = ABCDE X

F^{+} = F X

⇒ if A, F are present on the right side then there can be more candidate key

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

FD: {AB→CD, C→A, D→B}

$AB\cancel{CD}^{+}$ = {ABCD}

SK/CK ← AB^{+} = {ABCD}

A^{+} = {A} X

B^{+} = {B} X

Exclusive Falcon

Day: Boyce/Codd Normal form Date: / /

3NF/BCNF

1) 2nd NF ✓
2) It does not contain any transitive dependency for Non-Prime attributes. (NPA → NPA)

2NF →

Sid	Sname	DOB	State	Country	Pin	Hot
1	A	-	HR	PK	122001	-
2	B	-	HR	PK	122001	-
3	C	-	HR	PK	122001	-
4	D	-	Punjab	PK	123456	-

CK = Sid

Pin → State, Country

Update anomaly

NP → NP

non-prime.

Transitive dependency.

OR

- A table is in 3NF if and only if for each of its non-trivial functional dependency at least one of the following conditions holds:-
  1. LHS is SK
  2. RHS is Prime attribute.

① Example:- R(A, B, C, D)

FD: (A → B, B → C, C → D)

$ABCD^+ = \{ABCD\}$

CK / SK ← $A^+ = \{A, B, C, D\}$

Prime attribute → A

Result:- Now there should be no NPA → NPA

Day: Date: / /

(2) Example:-

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

FD: (AB → CDEF , BD → F)

$ABCDEF^+ = \{ABCDEF\}$

CK / SK ⇒ $AB^+ = \{A, B, C, D, E, F\}$

$A^+ = A$ ×

$B^+ = B$ ×

See PA on right side.

- Prime attributes are → A, B
- Non-Prime → C, D, E, F

TYPES OF FUNCTIONAL DEPENDENCIES

1) Trivial — FD: $X \to Y$ or $X \to X$ if $y \subseteq x$

↓ are always Valid.

2) Non-Trivial — FD: $X \to Y$ (Y is not subset of X)

3) Completely Non-Trivial — $X \cap Y = \phi$ (kuch bhi common nai ho)

4) Transitive — $X \to Y$, $Y \to Z$ than $X \to Z$.

4th normal form → multi valued dependencies

Day: 5th normal form → Join dependencies Date: / /

Integrity Constraints / Rules

1) Entity Integrity Constraints
- No null values in primary keys
- Uniqueness

2) Referential Integrity Constraint
- Data linked b/w two tables
- Foreign key values parent aur child tables ma same honi chachiy

3) Domain Constraint
- Same datatype in the columns

4) Attribute Constraint

5) Table Constraint (constraint on an individual table)
e.g., PK
Unique
FK
~~check~~

Day: Date: / /

Relational Algebra

Operators

- Basic operator
  - Projection (π)
  - Selection (σ)
  - Cross Product ($\times$)
  - Union ($\cup$)
  - Rename (ρ)
  - Set difference ($-$)
- Derived operator
  - Join ($\bowtie$)
  - Intersect ($\cap$)
  - Division ($/, \div$)

1) **Projection (π)** (Used at last) (Retrieves the data) (works on column)

By default ⇒ Unique (duplicates remove)

RollNo	Name	Age
1	A	20
2	B	21
3	A	19

Q: Retrieve rollno from table Stu?

$\pi_{rollno}(Stu)$

$\pi_{Name}(Stu)$

Name
A
B.

2) **Selection (σ)** (works on tuple)

Q: Retrieve the name of student whose roll no = '2'

$$\pi_{Name}\left(\sigma_{rollno=2}(Stu)\right)$$

If you wanna join, 1 column should be same.

Day: Date: / /

3) Cross - Product (x)

T_1

A	B	C
1	2	3
2	1	4

T_2

C	D	E
3	4	5
2	1	2

Cross Product (Total columns = T_1 columns + T_2 columns)

(Total rows = T_1 rows X T_2 rows)

A	B	C	C	D	E
1	2	3	3	4	5
1	2	3	2	1	2
2	1	4	2	1	2
2	1	4	2	1	2

4) Set difference (-) (A-B) = A but not B

A - B ≠ B - A

1, 2, 3 (S_1) 3, 4 (S_2)

- No. of columns must be same
- Domain or data in column should be same.

$S_1 - S_2$ = 1, 2

$S_2 - S_1$ = 4

Day: Date: / /

Roll no	Name
1	A
2	B
3	C

Student

Emp.No	Name
7	E
1	A

Employee

Student - Employee

Roll no	Name
2	B
3	C

Q: Name of person who are students but not emp.

$$\Pi_{name}(Stu) - \Pi_{name}(Emp)$$

5) Union (U)

- No. of columns must be same in tables.
- Domain should be same.

Roll no	Name
1	A
2	B
3	C
7	E

6) Rename (ρ) $\rho_{id/sid}(Student)$

new attribute / old attribute.

Student	grade

$$\Pi_{name}\left[\sigma_{grade='A'}(Student \times Grade)\right]$$

if rename

$$\rho_{Stugrade}\left[\sigma \ldots\right]$$

Day: Date: / /

1) Division Method :- (every, all) A(X,Y) / B(Y)

Q:- Retrieve sid of students who enrolled in all course?

done by Ravi & Isha

$E(sid, cid) / C(cid) = S_1$

sid	cid
S_1	C_1
S_2	C_1
S_1	C_2
S_3	C_2

Enrolled 'E'

cid
C_1
C_2

Course 'C'

2) Joins cross product + Condition
- Inner Join: Theta join, equi join, natural join
- Outer Join: Left, right, full

a) Natural Join (⋈)

Select E_name from Emp, Dept where Emp.no = Dept.no

↓ same as

Select E_name from Emp natural join Dept.

- Common column between tables.
- Name and type of attribute must be same.

b) Theta Join (θ) (other than equal condition, if we join R_1 and R_2)

↓ general case of join operation

R

A_1	A_2
20	25
80	40

S

B_1
50
35

A_1	A_2	B_1
20	25	50
20	25	35
80	40	50
80	40	35

$R \bowtie_{A_2 > B_1} S$

Day: Date: / /

c) Equi join

Where theta join uses only = Condition.

d) Left outer join (⟕)

- Set of tuples from left
- left say har cheez aye gi q aur agr wo right ma ni hy tu waha null aye jae gi.

e) Right outer join (⟖)

- Right table say sarri rows.

f) Full outer join (⟗)

- Right aur left sb ka sb combine kro.
- Jaha kuch na aye column ma waha NULL likh do.