(123) Lec-1 04/11/2020

(8-10 marks)

Syllabus

2m (1) Integrity Constraints and ER Diagrams

(2-4)m (2) Normalization

4m (3) Queries → Relational Algebra
→ SQL
→ TRC

2m (4) File Organization and indexing

(2-4)m (5) Transactions and concurrency control

20 hours
→ ERD and IC
→ Normalization
→ Queries → RA/TRC, SQL
→ Transactions and Concurrency control
→ Indexing Dynamic, MLI

* **ERD and IC**
  → Super keys → Candidate Keys → Alternate Keys
  → foreign keys → ~~Foreign keys~~
  → 1:M, M:1, M:N } RDBMS Design → ERD questions
  → 1:1 RDBMS Design

* **Normalization**
  (1) → Total number of trivial / non-trivial / semi non-trivial functional dependencies for n attribute relation
  (2) → FD set equality test
  (3) → FD set closure
  ** (4) → Candidate Keys – Finding
  (5) → lossless join and dependency preserving decomposition testing
  * (6) → 1NF, 2NF, 3NF, BCNF Definitions
  * (7) → Finding highest normal form

* Queries

* Relational Algebra → and SQL and TRC queries

① π and σ associativity | commutative

② min/max tuples

$\pi, \times, \bowtie, \cup, \cap, -, \bowtie, \ltimes, \times, /$ operators

③ Queries using join :

⇒ some ($\bowtie_c$) 2/3 queries

⇒ every 2 queries

⇒ set difference

⇒ intersection / union

④ TRC some $\exists$ and every $\forall$

* Transaction and Concurrency control

⇒ ACID properties

*** ⇒ Serializability Testing

conflict ~~serializability~~ serializable schedule

view serializable schedule

* ⇒ Recoverability classification

irrecoverable schedule

recoverable schedule

Cascadeless rollback schedule

Strict recoverable schedule

* ⇒ locking protocols

2PL locking protocol

Time stamp ordering protocol

* Indexing

⇒ Sparse and Dense index

⇒ Primary index

Clustering index

Secondary index

⇒ multi level index

*** ⇒ dynamic multilevel index

→ B Tree index

→ B+ Tree index

Q1 How many superkeys are possible for the relational schema R with n attributes ~~and~~ $R(A_1, A_2, A_3, \ldots A_n)$ and candidate keys $\{A_1, A_2A_3\}$?

Concepts 27/11/2020

(CK) Candidate Key : minimal set of attributes which can differentiate the records of relational schema uniquely

(i) minimal attribute set

(ii) able to differentiate records uniquely

sir's defn :* minimal set of attributes which can differentiate records of relational schema uniquely

example

Emp (eid, ename dob) Candidate Key : eid (minimal attrib set)

eid ename
eid dob
eid ename dob

all sets differentiate the records uniquely but eid alone is the minimal set which can diff. the records uniquely.

Works_For (eid pid since)

project id → pid

date attribute → since

eid	pid	since
e1	p1	2015
e1	p2	2018
e2	p2	2015

eid and pid are not unique for all the records

Candidate Key : {eid pid} combination of keys forms CK

* if single key is CK it is called Simple CK

* => CK with only one attribute => Simple CK

* if two or more attributes are combined to form a CK, such a key is called Compound CK

* => CK with two or more attributes => Compound CK

Primary Key (PK)

* any one candidate key of Relational schema which is always NOT NULL
* At most one PK for any relational schema
* Not allowed NULL values

example

R

A	B	C	D	E
a1	4			
a2	Null			
a3	6			
a4	Null			
a5	8			

Alternative Keys (AK)

* all candidate keys of relational Schema except primary key
* Many AKs are allowed for a relational Schema
* Allowed NULL Values

{A, B, CD} ⇒ Candidate Keys

A ⇓ PK

B, CD ⇓ AK

Candidate Keys ⇓ Can be NULL

Super Key (SK)

attribute set which can differentiate records uniquely

attribute set ⇓ may or may not be minimal

Emp (eid ename DOB)

eid	ename	DOB
2	A	1990
3	A	1985
4	B	1985
6	B	1990

eid : Candidate Key

Super Keys : {eid, eid ename, eid DOB, eid ename DOB}

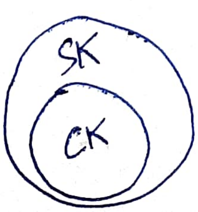
eid: minimal attribute set Super Key ⇒ CK

* CK is a subset of SK

CK ⇓ min. attrib set

SK ⇓ Just attrib set

For some Relational Schema R



* Candidate Key : minimal SK

every CK is a SK but not all SKs are CKs

$$CK \subseteq SK$$

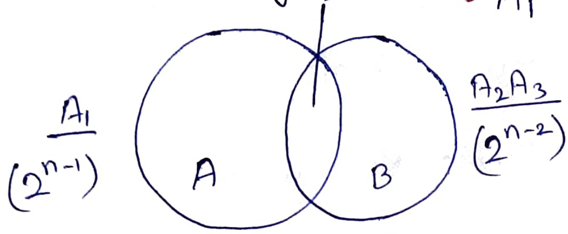
Solution for Q1

CKs = $\{A_1, A_2A_3\}$ $\qquad$ $R = \{A_1, A_2, A_3, \ldots, A_n\}$

Super Key : CK · {0 or more of other attributes}

Two ways of solving this :

1) Set Theory 2^{n-3}



A_1 · {any subset of $\underbrace{\text{other attributes}}$} ⇒ SKs

$\Downarrow$

$(n-1) \Leftarrow \{A_2, A_3, \ldots, A_n\}$
elements
or
attributes

Total possible subsets for the set $\{A_2, A_3, \ldots A_n\}$ with cardinality $(n-1)$ is the power set of this set

if $Q = \{A_2, A_3, A_4, \ldots A_n\}$ and $|Q| = n-1$

Power set of $Q = 2^Q$ and $|2^Q| = 2^{n-1}$ subsets possible

* Any of the 2^{n-1} subsets are combined with A_1 will be a SK
* Total possible super keys because of $A_1 = 2^{n-1}$

Now,

A_2A_3 · {any subset of $\underbrace{\text{remaining attributes}}$}

$\Downarrow$

$(n-2) \Leftarrow \{A_1, A_4, A_5, \ldots A_n\} = P$
elements
or attributes

$|\text{Power Set of } P| = 2^{n-2}$

* Total possible super keys because of $\underline{A_2A_3} = 2^{n-2}$

Common Super Keys : $A_1A_2A_3$, $A_1A_2A_3A_4$, ...

⇒ $A_1A_2A_3$ · {any subset of remaining attributes}

$\Rightarrow$ $A_1 A_2 A_3 \cdot \{$ any subset of $\underbrace{\text{remaining attributes}}$ $\}$

$\Downarrow$

$(n-3)$ attributes $\Leftarrow \{A_4, A_5, \dots A_n\} = T$

Power set of $T = 2^T$ $\quad |2^T| = 2^{n-3}$

* Total possible super keys which are common are 2^{n-3}

Total Super Keys : $n(A \cup B) = n(A) + n(B) - n(A \cap B)$

$n(A \cup B) = 2^{n-1} + 2^{n-2} - 2^{n-3} = 2^{n-3}\{2^2 + 2 - 1\}$

$\boxed{\text{Total SKs} = 5 \times 2^{n-3}}$

2) Another method

Total # of SK = $\{$ # of SKs among prime attributes $\} \times 2^{\{\#\text{ of Non-Prime attributes}\}}$

This can be written as :

Total # of SKs $= x \times 2^y$ where

x : # of SKs among prime attributes

y : # of Non-prime attributes

* Prime Attribute : attributes belonging to any one CK is prime

Example $R(A, B, C, D, E)$ $\quad CK = \{\underline{A}, \underline{B}, \underline{CD}\}$

Prime Attributes = $\{A, B, C, D\}$ all belong to some CK

Non-prime attribute = $\{E\}$

* Attributes not related to or not belonging to any CK are non-prime attributes

* In our question

$R(A_1, A_2, A_3, \ldots, A_n)$ $\qquad CK = \{\underline{A_1}, \underline{A_2 A_3}\}$

$$\text{Total SK} = x \times 2^y$$

x : # of SKs among prime attributes
y : # of non-prime attributes

Prime attributes $= \{A_1, A_2, A_3\}$ $\quad 3$
Non-Prime attrib $= \{A_4, A_5, A_6, \ldots A_n\}$ $\quad n-3$

this is one of the SKs

SK among prime attributes $= \{A_1, A_1A_2, A_1A_3, A_1A_2A_3, A_2A_3\}$

→ Since the only CK in set of prime attributes is A_1 $= A_1 \cdot \{\text{any subset of } \underbrace{\text{remaining keys}}_{\{A_2, A_3\}}\}$

Power set of $\{A_2A_3\} = \{\{\}, \{A_2\}, \{A_3\}, \{A_2A_3\}\}$

⇒ x : # of SKs among prime attributes
$= \{A_1, A_1A_2, A_1A_3, A_1A_2A_3, A_2A_3\} = 5$

$x = 5$

y : # of non-prime attributes
$= \{A_4, A_5, A_6, \ldots A_n\} = n-3$

$y = n-3$

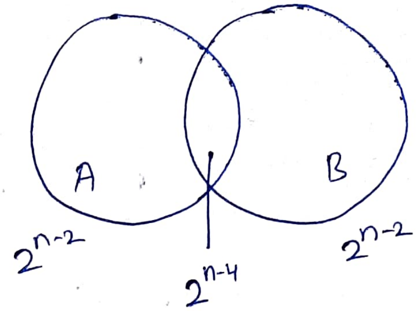
$$\text{Total SKs} = x \times 2^y$$

$$\boxed{\text{Total SKs} = 5 \times 2^{n-3}}$$

Q2 How many super keys for Relational Schema R with n attributes $R(A_1, A_2, A_3, \ldots, A_n)$ and Candidate Keys $\{A_1A_2, A_3A_4\}$?

Method 1 — Set Theory

Lec-2
27/11



$A_1A_2 \cdot \{$any subset of remaining attributes$\}$

$\Downarrow$

$n-2 \Leftarrow \{A_3, A_4, \ldots A_n\}$

$\Rightarrow 2^{n-2}$

$A_3A_4 \cdot \{$any subset of remaining attributes$\} \Rightarrow \{A_1, A_2, A_5, \ldots A_n\} \Rightarrow n-2$

$\Rightarrow 2^{n-2}$

Common SKs : $A_1A_2A_3A_4 \cdot \{$any subset of remaining attributes$\}$

$\Downarrow$

$\Rightarrow 2^{n-4}$ $\qquad (n-4) \Leftarrow \{A_5, A_6, \ldots A_n\}$

$n(A \cup B) = n(A) + n(B) - n(A \cap B)$

$= 2^{n-2} + 2^{n-2} - 2^{n-4}$

$= 2^{n-4}(2^2 + 2^2 - 1)$

$\boxed{\text{Total SKs} = 7 \times 2^{n-4}}$

* Proper subset of a CK is never a SK

Method 2
Faster method

Total SKs $= x \times 2^y$

x : # of SK among prime attributes
y : # of non-prime attributes

* prime attributes : $\{A_1, A_2, A_3, A_4\}$ 4
* Non-prime attributes : $\{A_5, A_6, \ldots A_n\}$ $n-4$

$\Rightarrow y = n-4$

SK among Prime attributes

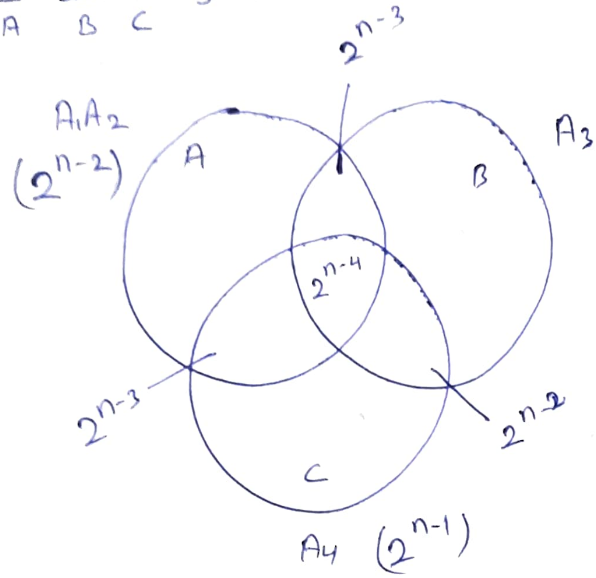
$\left\{ \begin{array}{ll} A_1A_2 & A_3A_4A_1 \\ A_3A_4 & A_3A_4A_2 \\ A_1A_2A_3 & \\ A_1A_2A_4 & \\ A_1A_2A_3A_4 & \end{array} \right\} \Rightarrow x = 7$

Total SKs $= x \times 2^y$

$\boxed{\text{Total SKs} = 7 \times 2^{n-4}}$

(131) Q3 How many Super keys possible for Relational Schema R with n attributes $R(A_1, A_2, A_3, \dots A_n)$ and Candidate Keys $\{\underbrace{A_1A_2}_{A}, \underbrace{A_3}_{B}, \underbrace{A_4}_{C}\}$?

Method 1



$n(A \cup B \cup C) = n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n(A \cap B \cap C)$

$n(A \cap B) = A_1A_2A_3 \cdot \{\text{any other subset of Remaining Keys}\}$

$n(A \cap B) = 2^{n-3}$

$n(B \cap C) = A_3A_4 \cdot \{\underbrace{\text{any other subset}}_{\Downarrow \; n-2}\}$

$n(B \cap C) = 2^{n-2}$

$n(A \cap C) = 2^{n-3}$

$n(A \cap B \cap C) = 2^{n-4}$

$\Rightarrow n(A \cup B \cup C) = n(A) + n(B) + n(C) - n(A \cap B) - n(B \cap C) - n(A \cap C) + n(A \cap B \cap C)$

$= 2^{n-2} + 2^{n-1} + 2^{n-1} - (2^{n-3} + 2^{n-2} + 2^{n-3}) + 2^{n-4}$

$= 2^{n-2}(1+2+2) - 2^{n-3}(1+2+1) + 2^{n-4}$

$= 5 \times 2^{n-2} - 4 \times 2^{n-3} + 2^{n-4}$

$= 2^{n-4}(5 \times 2^2 - 4 \times 2^1 + 1) = \boxed{13 \times 2^{n-4}}$

Method 2

Prime attributes = $\{A_1, A_2, A_3, A_4\}$

SKs among prime attributes = $\{A_1A_2, A_1A_2A_3, A_1A_2A_4, A_1A_2A_3A_4, A_3, A_3A_1, A_3A_2, A_3A_4, A_3A_1A_4, A_3A_2A_4, A_4, A_4A_1, A_4A_2\}$

$\boxed{x = 13}$

Non-prime attributes = $\{A_5, A_6, A_7, \ldots A_n\}$ = $n-4$ $\boxed{y = n-4}$

Total SKs = $x \times 2^y$

$\boxed{\text{Total SKs} = 13 \times 2^{n-4}}$

* more than 3 CKs in ques → method 2 is better
* CK ≤ 3 → method 1 is better

* Overall method 2 is better

Q4 How many superkeys possible for relational schema R with n attributes $R(A_1 A_2 A_3 \ldots A_n)$ and m simple CKs [assume m≤n]?

~~* meth~~

method 2

Total SKs = $x \times 2^y$

x : # of SKs among prime attributes

y : # of non prime attributes

Prime attributes = {set of CKs} m # of prime attributes

example 3 simple CKs

{A B C}

$\left.\begin{matrix}\{A, B, C \\ AB, BC, AC \\ ABC\}\end{matrix}\right\} \Rightarrow 2^3 - 1 = 7$

⇒ Super Keys are all subsets of set {A,B,C} except the empty set

Similarly for a set of m simple CKs the total possible SKs = $2^m - 1$

$\Rightarrow \boxed{x = 2^m - 1}$ excluding empty set

# of Non-prime attributes = total attributes − # of prime attributes

$\boxed{y = n - m}$

⇒ Total SKs = $x \times 2^y$

$\boxed{\text{Total SKs} = (2^m - 1) \times 2^{n-m}}$

only for m simple CKs

(133) Q5 How many super keys are possible for relational schema R with n attributes R(A_1, A_2, A_3... A_n) ?

* No CK set is given
* Total Possible SKs is asked => think of max possible CKs

if each attribute of relation R is a CK

{$\underline{A_1}$, $\underline{A_2}$, $\underline{A_3}$, ..., $\underline{A_n}$} => then we will get max possible SKs

Total SKs = $2^n - 1$ → exclude the empty set [Max Possible SKs]

↑ every possible subset of n attributes

example

R(A, B, C)

max possible SKs ?

=> all attributes are CKs

=> then max SKs possibles

=> R($\underline{A}$, $\underline{B}$, $\underline{C}$)

SKs = {A, B, C,
AB, BC, AC,
ABC}

=> $2^3 - 1$ => 7 SKeys

* if each attribute is a CK then max possible SKs is possible

---

Q6 How many maximum possible candidate keys in Relational Schema R with 4 attributes R(A, B, C, D) ?

(i) if one attribute forms a CK => $^4C_1 = 4$ CKs — A B C D

(ii) if two attributes forms a CK => $^4C_2 = 6$ CKs — AB AC AD BC BD CD

(iii) if three attributes forms a CK => $^4C_3 = 4$ CKs — ABC BCD ACD ABD

(iv) if all four attributes together form a CK => $^4C_4 = 1$ CK — ABCD

if we take set (i) as CK then we can't take set (ii) as CK

if we take any one set as CK set rest 3 will not be CK set

max possible set of CK = 6 CKs

Foreign Key (Referential Key)

* Used to relate data b/w tables (to establish relationship b/w tables)
* it is defined over two relations (tables)
  (i) Referenced Relation
  (ii) Referencing Relation

CK
PK
AK
SK
} ↳ Constraints of these keys are within one table

FK } Constraints are over two tables

Definition: Foreign Key is a set of attributes that references to Primary Key or Alternate Key of same relation or other relation.

→ X can be in same table as Y or in some other table

X ← references — Y (set of attributes)

PK/AK — Referenced attributes

FK — Referencing attributes

References: Enroll.Sid (FK) → Student.Sid (PK)

Student

Sid	Sname	DOB
S1	A	1990
S2	A	1985
S3	B	1985
S4	C	1990
S5	C	1995

Referenced Relation

Enroll

	Sid	Cid	Fee
×	S10	C1	5000
✓	S1	C1	5000
✓	S2	C1	4000
✓	S2	C2	5000
✓	S5	C3	3000
Not allowed →	S10	C1	3000

Referencing Relation

← this should not happen because this leads to data inconsistency

→ Sid of Enroll relation should always belong to Sid of Student

⇓

To make sure that this doesn't happen, we can declare Sid as FK which references Sid of Student

135

Employee

Eid (PK)	Ename	SupID
e1	A	Null
e2	A	e1
e3	B	e1
e4	C	e2
e5	B	Null
e6	A	e10 → Restricted

SupID: Supervisor of empolyee should also be an employee and their data should be present in Employee Table

SupID - FK
References
PK - Eid

Imp

Referencial Key Integrity Constraints
[Foreign Key IC]

(I) Referenced Relation

(II) Referencing Relation

Enroll References Student

Student

PK→ Sid	Sname	DOB
S1	A	1990
S2	A	1985 ← Deletion causes violation of IC
S3	B	1985 ← Deletion of this record will not cause violation
S4	C	1990
S5	C	1995
New → S6	D	1990

Enroll

FK→ Sid	Cid	Fee
S1	C1	5000
S2	C1	4000
S2	C2	5000
S5	C3	3000

* insertion of new record in Student relation does not cause any violation to the FK conditions of referencing relation

a) Insertion : No Violation

b) Deletion : May cause violation

(1) Do not allow the records being used in referencing table to be deleted from the referenced table

(2) if allowing to delete those records from referenced relation which are being used in referencing relation then DBMS should also delete those records from referencing table alongwith records from referenced table

To preserve Integrity constraints:
(i) Do not allow to delete ⇒ on delete no action
(ii) on delete cascade
(iii) on delete set null

c) Updation : May cause violation

③ Set null values to those records ~~when~~ get in referencing table whose corresponding records gets deleted from the referenced table

→ this can only be done when the FK of referencing table is not the PK of that table, in case FK is PK of referencing table then setting values as NULL is not allowed

references

R(A B) S(C D)

PK (under A); FK (Not PK) (under D)

Here, set null is allowed on delete as FK of S can be NULL

b) Deletion : may cause violation

(i) on delete no action

⟹ Deletion of referenced record is restricted if FK violation occurs

(ii) on delete cascade

⟹ Forced to delete referencing data

(iii) on delete set null

⟹ on deletion of referenced record sets NULL in related FK values

c) Updation : may cause violation

(i) ON UPDATE NO ACTION

⟹ Updation of referenced record is restricted if FK violation occurs

(ii) ON UPDATE CASCADE

⟹ Forced to update referencing data

(iii) ON UPDATE SET NULL

⟹ on updation of referenced record sets NULL in related FK values

(II) Referencing Relation

* Insertion may cause violation if inserted data not in referenced relation
* Deletion of referencing data does not cause any violation
* Updation may cause violation if updated data not in referenced relation

a) Insertion : may cause violation
=> if violation is caused on insert then restrict insertion

b) Deletion : No violation

c) Updation : may cause violation
=> restrict updation if violation caused

Q1 Consider the following tables

R

A	B
2	2 → Null
3	2 ×
4	4
5	4
6	10
7	12
8	12

S

C	D
× 2	3
× 3	3
4	4
5	4
6	7
10	8
12	8

In table R, A is PK and B is FK referencing to S with on delete set null.
In table S, C is PK and D is FK referencing to R with on delete cascade.
In order to delete record (3, 2) from R, number of additional records that need to be deleted ?

R (A, B) —refers, on delete set null→ S (C, D)

S (C, D) —refers, on delete cascade→ R (A, B)

Total # of additional Records deleted = 2
(2,3) and (3,3) of S

→ on deleting record (3, 2) from R
→ 3 is one of the PK
→ on delete of PK => cascade
→ 3 will also be deleted from referencing D FK
=> (2,3) and (3,3) from S will be deleted

→ 2 and 3 are PK of S (on delete set null)
→ corresponding values in R will be set null (2,2) → (2, Null)

Q2 Which is true regarding foreign key? **

a] each record of referencing relation ~~exactly~~ related to exactly one record of referenced relation
False

b] each record of referencing relation related to atmost one record of referenced relation
True

c] each record of referenced relation related to one or more records of referencing relation
False

d] each record of referenced relation related to many (0 or more) records of referencing relation
True

Referenced Relation

R(A B D)
PK AK

R

A	B	D
a1	2	
a2	4	
a3	Null	
a4	6	
a5	Null	
a6	8	

1:M

Referencing Relation

S

C	D
C1	Null
C2	2
C3	2
C4	4
C5	6

S(C D)
PK FK

Not referring to any value of B

FK whose value is Null is out of referential integrity constraint

* one record of referenced table is related to many records of referencing table

* some records of R are not related to any record of S

1 Record of R $\xrightarrow{\text{related to}}$ 0 or more records of S

at most one record of R $\xleftarrow{\text{related to}}$ 1 record of S

either not related to any record of R, as in:
$S \rightarrow (C_1, Null)$

related to exactly one record of R, as in:
$S \rightarrow (C_2, 2)$

Q How many non-trivial FDs are possible for given instance of record set?

R

A	B	C
6	4	7
6	4	9
6	6	7
6	6	9
8	4	7

✓ a) 0   b) 1

c) 2   d) 3

No Non-Trivial FDs

Functional Dependency   FD → (arrow)

R

	X	Y
t_1	x_1	y_4
t_2	x_1	y_4
⋮		

$X \rightarrow Y$ (X Determines Y)

(for each X there is one Y)

* if one record has x_1 and x_1 determines y_4 then if x_1 occurs again then it should determine y_4 only

$\left.\begin{matrix} x_1 & y_4 \\ x_1 & y_4 \\ \text{same } x & \text{same } y \end{matrix}\right\} \Rightarrow$ FD $x \rightarrow y$

$\left.\begin{matrix} x_1 & y_2 \\ x_1 & y_4 \end{matrix}\right]$ diff y $\Big\} \Rightarrow$ No FD b/w x and y

same x

* if $X \rightarrow Y$ in R then

if $(t_1.x = t_2.x)$ then $(t_1.y = t_2.y)$

Student

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S1	A	C3
S2	B	C3
S3	B	C3
S4	C	C3
S2	B	C2

Sid → Sname (Sid determines Sname)

~~if two or more students~~

[S1 A]
[S1 A]

Same Sid   Same Sname ⇒ FD Sid → Sname

* if same sid then same Sname — if this constraint is not violated then FD exists

* if Sid → Sname then Sname does not determine Sid

Trivial FD: if attribute determines themselves

Sid → Sid

Sname → Sname

* Sid Sname → Sname (subset of Sid Sname)

Sid Sname → Sid Sname

* $X \rightarrow Y$ is trivial FD if $X \supseteq Y$ → attribute sets

* R (A, B, C)

A → A	AB → B	ABC → A
B → B	AB → A	BC → B
C → C	ABC → AB	AC → C

All are Trivial FDs

Non-Trivial FD: if attribute determines some other attribute

Sid → Sname

Sid Cid → Sname

Non-Trivial FD, No common attribute

A → B

AB → C

141 Semi-Non Trivial FD : combination of Trivial and Non-Trivial

Sid $\rightarrow$ Sid Sname $\Longrightarrow$ Sid $\rightarrow$ Sid (Trivial)

Sid $\rightarrow$ Sname (Non Trivial)

$X \rightarrow YZ \Rightarrow X \rightarrow Y$

$X \rightarrow Z$

Sid Cid $\rightarrow$ Cid Sname (Semi non trivial)

→ Sid Cid $\rightarrow$ Cid (Trivial)

→ Sid Cid $\rightarrow$ Sname (Non-Trivial)

Solution

Possible Non-Trivial FDs

$A \rightarrow B^{\times}$	$B \rightarrow A^{\times}$	$A \rightarrow BC^{\times}$	~~A → CB~~
$B \rightarrow C^{\times}$	$C \rightarrow B^{\times}$	$B \rightarrow CA^{\times}$	
$A \rightarrow C^{\times}$	$C \rightarrow A^{\times}$	$C \rightarrow AB^{\times}$	
$AB \rightarrow C^{\times}$	$BC \rightarrow A^{\times}$	$AC \rightarrow B^{\times}$	

Non-Trivial FDs present in R = 0

Q Consider given relational schema R(A, B, C, D) and FD set $F = \{AB \rightarrow CD, C \rightarrow A, D \rightarrow B\}$. How many candidate key are possible for given relational schema?

* Finding # of Candidate Keys — most important ques

35:12 Lec4

~~* With Respect To FDs CKs can be defined as :~~

30/12/20

Super Key : X is some set of attribute of R

X is Super Key of R iff X^{+} = {determine all attributes of R}

↓

X closure

$R(ABCDEF)$ $\quad FD = \{AB \to C,\ C \to D,\ D \to E,\ E \to F,\ F \to A\}$

$(AB)^+ = \underbrace{\{A, B, C, D, E, F\}}_{\text{Determines all attributes of } R} \Rightarrow AB$ is SK of R

* if AB is SK then any superset of AB is also SK

$(BC)^+ = \{B, C, D, E, F, A\} \Rightarrow BC$ is SK

$(AC)^+ = \{A, C, D, E, F\} \Rightarrow AC$ is not SK

Candidate Key (minimal SK) $\quad CK \in SK$

* X is CK iff ① X must be SK $\Rightarrow X^+ = \{$all attrib of $R\}$
 ② No proper subset of X is SK
 $\Rightarrow \forall\ Y \subset X$ such that Y^+ doesn't determine all attributes of R

* $R(ABCDE)$ $\quad FD = \{AB \to C,\ C \to D,\ B \to E\}$

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

* Proper subsets of $AB = \{A, B\}$
* A is not SK $\quad$ * B is not SK

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

* AB is SK as well as CK of R (AB is minimal SK)

Solution $\quad R(A, B, C, D)$ $\quad FD = \{AB \to CD,\ C \to A,\ D \to B\}$

$(AB)^+ = \{C, D, A, B\} \Rightarrow AB$ is SK

$A^+ = \{A\}$ $\quad B^+ = \{B\}$ $\Rightarrow AB$ is CK

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

$C^+ = \{C, A\}$ $\quad D^+ = \{B, D\}$ $\Rightarrow CD$ is CK

(143)

~~CK = {AB, CD, ABC, ABD, CDA, CDB, ABCD}~~

~~Prime Attrib = {A, B, C, D}~~

~~Non Prime Attrib = { } = ϕ~~

CK = {AB, AD, BC, CD}

$\boxed{\text{4 CKs possible}}$

* if some <u>non-trivial FD</u> $X \longrightarrow Y$ with Y is prime attribute of R then R consists of more than one CK

* AB is a CK and $D \longrightarrow B$ is of $X \longrightarrow Y$ form where Y is prime

prime

$\Rightarrow$ in place of B in AB we take D to form AD

$(AD)^+ = \{A, D, B, C\}$ AD is also SK (guaranteed)

$A^+ = \{A\}$ $D^+ = \{D, B\}$ $\Rightarrow$ AD is CK

* in $C \longrightarrow A$ FD A is prime

in place of A in AB we can take C $\Rightarrow$ CB

in place of A in AD we can take C $\Rightarrow$ CD

$(CB)^+ = \{A, B, C, D\}$

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

} both are SK

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

$\Downarrow$

(CB) and (CD) are CKs

30/12/2020 Lec-5

Q Consider given relational schema R(A B C D E F) and FD set $F = \{AB \to C,\ C \to DE,\ E \to F,\ F \to A\}$ How many CKs are possible for given relational schema?

Soln

$$F = \{AB \to C,\ C \to DE,\ E \to F,\ F \to A\}$$

* $(AB)^+ = \{A, B, C, D, E, F\}$ AB is SK

  $A^+ = \{A\}$ $B^+ = \{B\}$ $\Rightarrow$ $\boxed{AB \text{ is CK}}$

* prime = $\{A, B\}$

* $\underbrace{F \to A}_{\text{Non-T}}$, A is prime , A can be replaced by F in (AB)

  $(FB)^+ = \{A B C D E F\}$ FB is SK

  $F^+ = \{F, A\}$ $B^+ = \{B\}$ $\Rightarrow$ $\boxed{FB \text{ is CK}}$

* prime = $\{A, B, F\}$

* $\underbrace{E \to F}_{\text{Non-T}}$, F is prime , F can be replaced by E in (FB)

  $(EB)^+ = \{A B C D E F\}$ EB is SK

  $E^+ = \{E, F, A\}$ $B^+ = \{B\}$ $\Rightarrow$ $\boxed{EB \text{ is CK}}$

* prime = $\{A B F E\}$

* $\underbrace{C \to E}_{\text{Non-T}}$, E is prime , E can pe replaced with C in EB

  $(CB)^+ = \{A B C D E F\}$ CB is SK

  $C^+ = \{C, D, E, F, A\}$ $B^+ = \{B\}$ $\Rightarrow$ $\boxed{CB \text{ is CK}}$

* prime = $\{A B F E C\}$

$\Rightarrow$ CK = $\{AB, FB, EB, BC\}$

$\boxed{4 \text{ CKs}}$

* NP-Complete problem – finding CKs of R [Exponential Time Complexity]

Q Consider given relational schema R(ABCDEF)

and FD set F = {AB → C, C → D, CD → AE, DE → F,
EF → B}

How many CKs are possible for given relational schema?

Sol^n

CK = { AB, AEF

prime = { A, B, E, F

non-prime = {

* ~~$(AB)^+ = \{A B C D\}$ ×~~
* ~~$(CD)^+ = \{A, E, C, D\}$ ×~~

* $(AB)^+ = \{ABCDEF\}$ AB is SK

  $A^+ = \{A\}$ $B^+ = \{B\}$ AB is CK

* CD → A, A is prime

  $(CDB)^+ = \{ABCDEF\}$ CDB is SK Not CK

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

* EF → B, B is prime

  $(AEF)^+ = \{ABCDEF\}$ AEF is SK and CK

  $A^+ = \{A\}$ $E^+ = \{E\}$ $F^+ = \{F\}$

  $(EF)^+ = \{E, F, B\}$ $(AE)^+ = \{A, E\}$ $(AF)^+ = \{A, F\}$

19:50

Sir's Solution

$(AB)^+ = \{A, B, C, D, E, F\}$ SK ✓

$A^+ = \{A\}$ $B^+ = \{B\}$ CK ✓

Prime = {A, B}

CK = {AB}

are there any productions having either A or B (primes) on RHS ?

(i) CD ⟶ A | (ii) EF ⟶ B

let's put EF in place of B in AB

Prime = {A, B, E, F}

$(AEF)^+ = \{A, B, C, D, E, F\}$ SK ✓

$A^+ = \{A\}$ AE = {A, E} CK ✓

$E^+ = \{E\}$ AF = {A, F}

$F^+ = \{F\}$ EF = {E, F, B}

CK = {AB, AEF}

let's put CD in place of A in AB

$(CDB)^+ = \{A, B, C, D, E, F\}$ SK ✓

$C^+ = \{A B C D E F\}$ CK ✗

This thinking is wrong because we have to consider AE as one prime unit

are there any productions having prime attributes on RHS ?

(i) DE ⟶ F

let's put DE in place of F in (AEF)

Primes = {A, B, E, F, D}

$(ADE)^+ = \{A, B, C, D, E, F\}$ SK ✓

$A^+ = \{A\}$ AD = {A, D} CK ✓

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

$E^+ = \{E\}$ DE = {D, E, F, B}

CK = {AB, AEF, ADE}

AE can be substituted by CD in either (i) AEF or (ii) ADE

(i) $(CDF)^+ = \{A B C D E F\}$ SK ✓ (ii) $(CD)^+ = \{ABCDEF\}$ SK ✓

$C^+ = \{ABCDEF\}$ CK ✗ $C^+ = \{ABCDEF\}$ CK ✗

(147) 02/01/2021

$\Longrightarrow$ C is a CK since it is minimal SK

CK = { AB, AEF, ADE, C }   total (4) CKs

ACE is also not a CK but SK

Q Consider given relational schema R(A, B, C, D, E, F) and FD set F = { $AB \rightarrow C$, $C \rightarrow A$, $BC \rightarrow D$, $ACD \rightarrow B$, $BE \rightarrow C$, $EC \rightarrow FA$, $CF \rightarrow BD$, $D \rightarrow E$ } How many CKs are possible for given relational schema?

Soln

* Finding CKs of R with n attributes is NP-Complete problem (exponential time complexity)

$(AB)^+ = \{ABCDEF\}$ SK ✓

$A^+ = \{A\}$ $B^+ = \{B\}$ CK ✓

CK = { AB, BC, CD, BE, BD, CF, EC,

(i) $C \rightarrow A$ (ii) $ACD \rightarrow B$

$\Downarrow$

$(CB)^+ = \{ABCDEF\}$ SK ✓

$C^+ = \{C, A\}$ $B^+ = \{B\}$ CK ✓

$(ACD)^+ = \{ABCDEF\}$ SK ✓

$A^+ = \{A\}$ AC = {A, C}   CK ✗

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

$D^+ = \{D, E\}$ $\boxed{DC = \{CDEAFB\}} \longrightarrow$ CK

D = {D, E}   C = {A, C}

(i) $BE \rightarrow C$ (ii) $AB \rightarrow C$

four possible combinations

1. (i) in BC
2. (ii) in BC
3. (i) in CD
4. (ii) in CD

1. (i) in BC

$(BE)^+ = \{ABCDEF\}$ SK ✓

$B = \{B\}$ $E^+ = \{E\}$ CK ✓

2. (ii) in BC

$(AB)^+ \rightarrow$ SK, CK

3. (i) in CD

$(BED) \rightarrow$ SK ✓ CK ✗

$\Downarrow$

BE is CK

4. (ii) in CD

$(ABD) \rightarrow$ SK ✓ CK ✗

$\Downarrow$

AB is CK

(i) $ACD \rightarrow B$ (ii) $D \rightarrow E$

1. (i) in BE
2. (ii) in BE

1. ~~ACD~~ (i) in BE

$(ACDE)^+ \Rightarrow$ SK ✓

CD is CK $\leftarrow$ CK ✗

2. (ii) in BE

$(BD)^+ = \{ABCDEF\}$ SK ✓

$B^+ = \{B\}$ $D^+ = \{D, E\}$ CK ✓

(i) $CF \rightarrow BD$

$(CF)^+ = \{ABCDEF\}$ SK ✓

$C^+ = \{AC\}$ $F^+ = \{F\}$ CK ✓

(i) $BE \rightarrow C$

in CF

$(BEF)^+ = \{ABCDEF\}$ SK ✓

BE is CK $\leftarrow$ CK ✗

(ii) $AB \rightarrow C$

in CF

$(ABF) \rightarrow$ SK ✓

CK ✗ $\rightarrow$ AB is CK

(i) EC → F in CF

(ii) EC → A in AB

(EC)$^+$ = {ABCDEF} SK ✓

E^+ = {E} C^+ = {C, A} CK ✓

(ECB) → SK ✓

CK ✗

BC
BE } CK ✓
EC

Q Consider given FD set

{AB → CD, AF → D, DE → F, C → G, F → E, G → A}

Which FD is not member of given FD set?

a) BEG → CF

b) BG → D

c) BCF → A

✓d) AB → E

F = {. . .} (FD set)

~~if you want to prove that~~ X → Y is a member of F

~~then~~ iff X^+ = {. . . Y . . } ~~iff ... possible~~

Compute Closure of LHS if it determines RHS it is a member of FD set

(BEG)$^+$ = {B, E, G, A, C, D, F} BEG → CF ✓

(BCF)$^+$ = {B, C, F, G, A, E, D} ✓

(BG)$^+$ = {B, G, A, C, D} ✓

(AB)$^+$ = {A, B, C, D, G} AB → E ✗

Q Considers given FD sets

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

$G = \{A \to BCD,\ B \to ACD,\ C \to ABD,\ BD \to A,\ AD \to B,$
$AB \to D,\ BC \to A\}$

Which is true for given FD set?

a) $F \subset G$ ✓ b) $F \supset G$ c) $F = G$ d) None

Concept ☆ Expressive Powers of FD Sets ☆

$F \equiv G$

F covers G and G covers F

* every FD of G set must be a member of F set

$\boxed{F \supseteq G}$

* every FD of F set must be a member of G set

$\boxed{F \subseteq G}$

F

$A^+ = \{ABCD\}$ ✓
$B^+ = \{ABCD\}$ ✓
$C^+ = \{ABCD\}$ ✓
$D^+ = \{ABCD\}$ ✗
$(BD)^+ = \{ABCD\}$
$(AD)^+ = \{ABCD\}$
$(AB)^+ = \{ABCD\}$
$(BC)^+ = \{ABCD\}$

G covers F ✗

$D^+ = \{D\}$
in G set

G

$A^+ = \{ABCD\}$ ✓
$B^+ = \{ABCD\}$ ✓
$C^+ = \{ABCD\}$ ✓
$(BD)^+ = \{A, B, D, C\}$ ✓
$(AD)^+ = \{ABCD\}$ ✓
$(AB)^+ = \{ABCD\}$ ✓
$(BC)^+ = \{BCAD\}$ ✓

1:02:51

$\Rightarrow$ F covers G ✓

$\boxed{F \supset G}$

(151)

Q Consider given FD sets

$F = \{A \rightarrow BCDEF, BC \rightarrow ADEF, B \rightarrow F, D \rightarrow E\}$

$G = \{A \rightarrow BCD, BC \rightarrow A, B \rightarrow F, D \rightarrow E\}$

Which of the following is true for the given FD sets?

a) $F \subset G$ b) $F \supset G$ c) $F = G$ d) None

Sol^n

G Covers F	F Covers G
Productions of F :	Productions of G
$A^+ = \{ABCDEF\}$	$A^+ = \{BCDAFE\}$
$B^+ = \{BF\}$	$B^+ = \{B, F\}$
$(BC)^+ = \{ABCDEF\}$	$D^+ = \{D, E\}$
$D^+ = \{D, E\}$	$(BC)^+ = \{ABCFDE\}$

F Covers G	G covers F	Conclusion
T	T	$F = G$
T	F	$F \supset G$
F	T	$F \subset G$

Properties of Decomposition

① loss less Join decomposition

② Dependency Preserving Decomposition

* To reduce / eliminate redundancy in DB Table ⟹ we perform decomposition of relations into subrelations according to some rules

Loss less Join Decomposition

Relational Schema R with instance r decomposed into sub-relations R1 R2 R3 ... Rn

(instance r ↳ record set)

① In General, $\underbrace{R_1 \bowtie R_2 \bowtie \cdots \bowtie R_n}_{\text{Natural Join of Sub-relations}} \supseteq R$

② if $R_1 \bowtie R_2 \bowtie \cdots \bowtie R_n = R$ then lossless join decomposition

③ if $R_1 \bowtie R_2 \bowtie R_3 \bowtie \cdots \bowtie R_n \supset R$ then lossy join decomposition

$\boxed{\bowtie \text{ Natural Join Symbol}}$

example CK : Sid Cid → Combination of Sid & Cid is unique

R

(record set) r

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_3	B	C_3

Decompose R into sub-relations

R_1(Sid Sname)

Sid	Sname
S_1	A
S_2	B
S_3	B

R_2(Sid Cid)

Sid	Cid
S_1	C_1
S_1	C_2
S_2	C_2
S_3	C_3

FD = { Sid → Sname }

$R_1 \bowtie R_2$

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_3	B	C_3

4 records

- Natural join combines both tables' data records Based on constraint equality of common attribute
- every record data of Sid in R_1 should be checked with every record data of Sid in R_2 to find $\bowtie$ b/w R_1 & R_2

$\Rightarrow R_1 \bowtie R_2 = R$ exactly equal
⇓
lossless join decomposition

Decompose R into sub-relations

R_1 (Sid Sname)

S_1	A
S_2	B
S_3	B

R_2 (Sname, Cid)

A	C_1
A	C_2
B	C_2
B	C_3

Common attribute in R_1 and R_2 = Sname

$R_1 \bowtie R_2$

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_2	B	C_3
S_3	B	C_2
S_3	B	C_3

original records

extra records

6 Records

$\underbrace{(R_1 \bowtie R_2) \supset R}$
⇓
lossy join decomposition

Spurious Tuples/Records
Cause inconsistency in data

* if common attribute of both sub-relations is a SK or CK or PK of one of the subrelations then $\bowtie$ is lossless join decomposition

~~May or may not be~~

* if common attribute is not SK/CK for any sub-relation then there is a chance of spurious records (may or may not be lossy decomp)

* Relational Schema R with FD set F decomposed into Sub-Relations R_1, R_2

Given decomposition is lossless Join

iff ① $R_1 \cup R_2 = R$ and ↔ (attributes of R_1 union with attributes of R_2 should be exactly equal to attrib of R)

② $R_1 \cap R_2 \rightarrow R_1$
OR
$R_1 \cap R_2 \rightarrow R_2$

→ ~~Intersection of attributes of R_1 and R_2 should be either equal to all attributes of R_1 or all attributes of R_2~~

↓ Common attribute should be a SK of either R_1 or R_2

Example

① ✗
② ✓

R(ABCDE)

R_1(ABC) R_2(CD)

lossy join → attribute E is missing

① all attributes of R are present in sub-relations or not
② Common attribute set b/w R_1 & R_2 must be a SK of any subrelation

---

Q Which of the following is true if Relational Schema R is decomposed into sub-relations R_1 R_2 R_3 and given decomposition is lossy join decomposition?

a) $R_1 \bowtie R_2 \bowtie R_3 = R$ lossless

b) $R_1 \bowtie R_2 \bowtie R_3 \subset R$ never occurs

c) $R_1 \bowtie R_2 \bowtie R_3 \supset R$ lossy

d) None

lossless join decomposition
① union of attributes of all sub relations must be exactly equal to attributes of R
② Common attribute should be a SK of either of the participating sub-relations

(155) Q Consider relational schema R(A B C D) FD set of relation R and decomposed sub relations are given below

(i) $\{B \to C, D \to A\}$ $\{\overset{R_1}{AB}, \overset{R_2}{BC}\}$ lossy

✓ (ii) $\{AB \to C, C \to A, C \to D\}$ $\{\overset{R_1}{ACD}, \overset{R_2}{BC}\}$ ~~lossy~~ lossless

✓ (iii) $\{A \to BC, C \to AD\}$ $\{\overset{R_1}{ABC}, \overset{R_2}{AD}\}$ lossless

✓ (iv) $\{A \to B, B \to C, C \to BD\}$ $\{\overset{R_1}{AB}, \overset{R_2}{BC}, \overset{R_3}{CD}\}$ lossless

How many decompositions are lossless join decompositions ?

3 are lossless join decompositions

(i) R(A B C D) $F = \{B \to C, D \to A\}$

$R_1 = AB$ (2) Common = B , B is SK of R_2

$R_2 = BC$

(1) $R_1 \cup R_2 = \{A B C\}$ D is missing $\Rightarrow$ Lossy decomposition

(ii) R(A B C D) $F = \{AB \to C, C \to A, C \to D\}$

$R_1 = ACD$ (1) $R_1 \cup R_2 = R$ ✓

$R_2 = BC$ (2) Common = C , SK for R_1

$C^+ = \{C, A, D\}$ ✓

$\Rightarrow$ ~~Lossy decomp~~

lossless decomposition

(iii) R(A B C D) $FD = \{A \to BC, C \to AD\}$

$R_1 = ABC$ (1) $R_1 \cup R_2 = R$ ✓

$R_2 = AD$ (2) Common = A , SK ✓

$A^+ = \{A B C D\}$ ✓

Lossless decomposition

(iv) $R(ABCD)$

$F = \{A \to B,\ B \to C,\ C \to BD\}$

$R_1 = AB$

$R_2 = BC$ ① $R_1 \cup R_2 \cup R_3 = R$ ✓

$R_3 = CD$ for ② →

① let's consider R_1 R_2 first (First Join b/w R_1 R_2 — lossless)

$R_1 = AB$ ~~[illegible]~~ ② common $= B$

$R_2 = BC$ $B^+ = \{B\ C\ D\}$ B is SK for R_2

$R_{12}(ABC)$

② let's consider R_{12} and R_3 (Second Join b/w R_{12} and R_3 — lossless)

$R_{12} = ABC$ ② common $= C$ C is SK for R_3

$R_3 = CD$ $C^+ = \{C, B, D\}$

$R_{123}(ABCD)$ ✓ NO violation L6 | 53:16

Dependency Preserving ~~[illegible]~~ Decomposition

Relational schema R with FD set F decomposed into subrelations $R_1\ R_2\ R_3 \ldots R_n$ with FD sets $F_1\ F_2\ F_3 \ldots F_n$ respectively.

① In General, $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} \subseteq F$

② if $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} = F$ Then dependency preserving decomposition

③ if $\{F_1 \cup F_2 \cup F_3 \ldots \cup F_n\} \subset F$ Then not dependency preserving decomposition

$\{F_1 \cup F_2 \cup \ldots F_n\} \supset F$ NEVER OCCURS

157

Q ~~Ret~~ Consider relational schema $R(ABCD)$ FD set of relation R and decomposed sub relations given below.

(i) $\{B \rightarrow C, D \rightarrow A\}$ $\{AB, BC\}$

(ii) $\{AB \rightarrow C, C \rightarrow A, C \rightarrow D\}$ $\{ACD, BC\}$

(iii) $\{A \rightarrow BC, C \rightarrow AD\}$ $\{ABC, AD\}$

(iv) $\{A \rightarrow B, B \rightarrow C, C \rightarrow BD\}$ $\{AB, BC, CD\}$

How many decompositions are dependency preserving decompositions?

(i) $R(ABCD)$

$F = \{B \rightarrow C, D \rightarrow A\}$

$R_1 = AB$
$R_2 = BC$

$\Rightarrow$ for $A \rightarrow B$ to exist A^+ (A closure) should determine B in R and for $B \rightarrow A$ to exist B^+ should determine A in R.

$F_1 = \{\}$
$F_2 = \{B \rightarrow C\}$

$(F_1 \cup F_2) \subset F$

Not Dependency Preserving Decomposition

(ii) $R_1 = ACD$ $F_1 = \{C \rightarrow A, C \rightarrow D\}$
$R_2 = BC$ $F_2 = \{\}$

Not Dependency Preserving

(iii) $R_1 = ABC$ $F_1 = \{A \rightarrow BC, C \rightarrow AB\}$ $(AB)^+ = \{ABC\}$ $C^+ = \{CADB\}$
$AB \rightarrow C$
$BC \rightarrow A$
$R_2 = AD$ $F_2 = \{A \rightarrow D\}$

Dependency preserving

(iv) $R_1 = AB$ $F_1 = \{A \rightarrow B\}$
$R_2 = BC$ $F_2 = \{B \rightarrow C, C \rightarrow B\}$
$R_3 = CD$ $F_3 = \{C \rightarrow D\}$

Dependency Preserving

Steps

1. Find FD set of Sub Relations
2. take union of FDs of sub-relations and check with FD of R

Normal Forms (60% -7 0% questions of Normalization)

* used to reduce/eliminate redundancy
* based on the normal form satisfied we can understand the degree of Redundancy

* lower normal form — more redundancy
* higher normal form — less redundancy.

Redundancy

Over Single (SVD) valued dependency
FD (X → Y)

over MVD (Multi-Valued Dependency)
FD (X →→ Y)

* Different NFs ⟹ 1NF 2NF 3NF BCNF 4NF

Most Redundancy

1NF — Default Normal Form of RDBMS Tables/Relations

2NF — Reduces Degree of Redundancy

all are defined over SVDs — Redundancy may exist over MVDs

3NF — Further more reduction in redundancy

BCNF — Completely Eliminates Degree of Redundancy

least Redundancy

⇓

* 0% Redundancy over SVD
* Not 0% Redundancy over MVD [may suffer redundancy because of MVD]

4NF
- defined over MVDs (X →→ Y)
- 0% redundancy over SVDs (FDs)
- 0% redundancy over MVDs

⟹ First Normal Form (1NF)

Every RDBMS relation must satisfy this condition

* Default NF of RDBMS relations
* Relation R is in 1NF iff

→ No MVA attributes in R [every attribute of R must be single valued or atomic]

→

example

Emp (eid ename phno)

eid	ename	phno
e1	A	P1, P2
e2	B	P2, P3
e3	B	P4, P1

phno – MVA (Multi-value attrib.)

* Since MVA exist in Emp relation – Not in 1NF
– Not satisfies RDBMS condition

Emp

eid	ename	phno
e1	A	P1
e1	A	P2
e2	B	P2
e2	B	P3
e3	B	P4

Redundancy (due to Partial Dependency)

$F = \{eid \rightarrow ename\}$

* To convert MVA to SVA, design CK properly

~~Old CK: eid~~
~~New CK: eid phno~~

This relation is in 1NF (no MVA, allowed in RDBMS)

if R(ABC)

$F = \{A \rightarrow B\}$

CK = AC minimal SK

Similarly

Emp (eid ename phno)

$F = \{eid \rightarrow ename\}$

CK = eid phno → minimal SK

CK: eid phno

FD = {eid → ename}

⇓

Partial Dependency

⇓

Not in 2NF but in 1NF

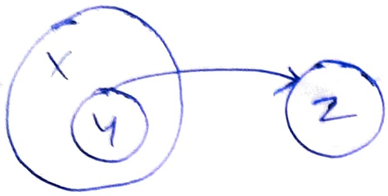
⇓

because No MVA in R

Second Normal Form (2NF)

Relational Schema R is in 2NF iff No Partial Dependency in relation R [Redundancy due to partial dependency should not arise; Relation must already be in 1NF]

Partial Dependency ⇒ proper subset of CK is determining some non-prime attribute



X : Any Candidate Key of R

$Y \subset X$

Y is proper subset of CK

Z is some non-prime attribute of R

$Y \to Z$: partial dependency
(Z: non-prime)

example: R(ABC)

AB is CK X = AB

B is ⊂ AB Y = B

C is non-prime attrib of R(ABC)

if $B \to C$ exists ⇒ then it is called partial dependency

1NF No MVA, default RDBMS NF

2NF No Partial Dependency, already in 1NF

3NF $X \to Y$: Non-trivial FD
X: SK OR Y: Prime

BCNF $X \to Y$ (for all non-trivial FDs)
X: SK

4NF

(61)

Third Normal Form (3NF)

A Relational Schema R is in 3NF iff every non-trivial FD $X \rightarrow Y$ in R with

1. X must be SK of R

OR

2. Y must be prime attribute of R

(determinant) X ⟶ Y (determiner) : Non-Trivial FD

X ↓ SK of R OR Y ↓ Y is prime (both can also be true at a time)

* 3NF fails when neither X is SK of R nor Y is prime

example R(A B C D E) F = { AB → C, C → A, C → D, AB → E }

prime = { A B C } non-prime = { D E }

CK = { AB, BC }

$(AB)^+ = \{A, B, C, D, E\}$ SK ✓

$A^+ = \{A\}$ $B^+ = \{B\}$ CK ✓

$(BC)^+ = \{A, B, C, D, E\}$ ✓

$C^+ = \{C, A\}$ $B^+ = \{B\}$ ✓

allowed in 3NF

F = { AB → C, ✓

C → A, ✓

C → D, ✗

AB → E ✓

}

(AB) → SK, CK

$C^+ = \{C, A, D\}$ Not SK

A → prime

BCNF (Boyce-Codd Normal Form)

* Relational Schema R is in BCNF iff every non-trivial FD X → Y in R with X must be Super Key.

∀ non-trivial FDs (X → Y) in R: {X : Super Key of R}

For All

⇓

This restriction is allowed for trivial FD, this restriction is only meant for Non-trivial FDs

Allowed in BCNF:
- * Trivial FD — No Redundancy, No Restriction
- * Non-Trivial FD ⇒ if X → Y : No Redundancy
  is SK

Not allowed in BCNF:
- * Non-Trivial FD: X → Y with X : Not SK ⇒ Forms Redundancy

⇓

if this kind of non-trivial FD exists then Relation is not in BCNF

* if Relation R is in BCNF
  - ⇒ 0% Redundancy in R over FDs (SVDs)
  - ⇒ May not have 0% redundancy over MVDs

Lec-8 02/01/2021

Q Consider given relational schema R with FD set R(ABCDE)

FD = {A → B, BC → E, ED → A}

What is the highest normal form of R?

a) 1NF b) 2NF c) 3NF d) BCNF

1NF	No MVA	
2NF	No P.Dep.	
3NF	X → Y (X: SK, Y: Prime)	
BCNF	X → Y (X: SK)	

Need to find
1) CK ← SK required
2) Prime
3) Non-prime

① Find all CKs of R
② Find prime & non-prime
③ Check NF

163

$R(ABCDE)$ $\{A \to B, BC \to E, ED \to A\}$

$A^+ = \{A B\}$ SK ✗

$(BC)^+ = \{B C E\}$ SK ✗

$(ED)^+ = \{E D A B\}$ SK ✗ $\Rightarrow$ EDC is SK

$(EDC)^+ = \{A B C D E\}$ SK ✓

$E^+ = \{E\}$ $D^+ = \{D\}$ $C^+ = \{C\}$ CK ✓

$(ED)^+ = \{E D A B\}$ $(EC)^+ = \{E C\}$ $(DC)^+ = \{D C\}$

CK = {EDC, BCD, ACD} 3 CKs

prime = {E, D, C, B, A}

non-prime = { }

(i) $BC \to E$, E is prime

$\Rightarrow$ $(BCD)^+ = \{A B C D E\}$ SK ✓

$B^+ = \{B\}$ $C^+ = \{C\}$ $D^+ = \{D\}$ CK ✓

$(BC)^+ = \{B C E\}$ $(CD)^+ = \{D C\}$ $(BD)^+ = \{B D\}$

(i) $A \to B$, B is prime

$(ACD)^+ = \{A B C D E\}$ SK ✓

$A^+ = \{A B\}$ $C^+ = \{C\}$ $D^+ = \{D\}$ CK ✓

$(AC)^+ = \{A C B E\}$ $(AD)^+ = \{A B D\}$ $(CD)^+ = \{C D\}$

(i) $ED \to A$

(EDC) $\to$ already CK

FD = { A → B , BC → E , ED → A }

BCNF: A → B ↓ Not SK; BC → E ↓ Not SK; ED → A ↓ Not SK

✗ Not BCNF (even if one FD fails then also NOT BCNF)

3NF: A → B (A ✗, B ✓); BC → E (BC ✗, E ✓); ED → A (ED ✗, A ✓)

All are prime

✓ $\boxed{3NF}$ is the highest NF

* A relation R if in a higher NF then it is also in all lower NFs.

Q Consider given relational schema R with FD set R (ABCDE) FD = { ABD → C, BC → D, CD → E } What is highest normal form of relational schema R?

~~a) 1NF~~ b) 2NF c) 3NF d) BCNF

1. Find CKs
2. Find prime & non-primes
3. Check NFs

CK = { ABD, ABC }

Prime = { A B C D }

non-prime = { E }

FD = { ABD → C, BC → D, CD → E }

- ✗ BCNF: ABD → C ✓ SK; BC → D ✗ Not SK; CD → E ✗ Not SK
- ✗ 3NF: ✓ C → prime; ✓ D → prime; ✗
- ✗ 2NF

Condition for 2NF to exist:

{ proper subset of CK of R }$^+$ = { only prime attrib of Relation R }

⇓

if this is satisfied then

⇓

No chance of Partial Dependency

⇓

2NF

* Proper subsets of ABD, ABC

$A^+ = \{A\}$ ✓

$B^+ = \{B\}$ ✓

$D^+ = \{D\}$ ✓

$(AB)^+ = \{AB\}$ ✓

$(BD)^+ = \{BD\}$ ✓

$(AD)^+ = \{AD\}$ ✓

$C^+ = \{C\}$ ✓

$(AC)^+ = \{AC\}$ ✓

$(BC)^+ = \{BCDE\}$ ✗

↑ Non-prime (E)

* The condition for 2NF is not satisfied => 2NF not exist
* Partial Dependency $BC \to E$ exists in R => Not in 2NF
* Check for 1NF is not required as it is default for RDBMS

Q Which of the following is false if relational schema R is in 3NF but not in BCNF?

a) R must consist of at least two compound keys (True)

b) R must consist of overlapped CKs (True)

c) R must have at least one non-trivial FD such that proper subset of CK determines proper subset of some other CK of R (True)

d) R must consist of at ~~least~~ most one compound key (False)

☆ R in 3NF but not in BCNF
(Determinant is non SK) (Determiner is prime)

$\underbrace{\text{Proper subset of CK} \xrightarrow{\text{determines}} \text{Proper subset of some other CK}}$

⇓

To form this kind of FD, R should have at least 2 Compound CKs ⇐ { this kind of FD is allowed in 3NF but not allowed in BCNF

example R(ABC) {AB → C, C → A} CK = {AB, BC} overlapped CKs

BCNF → False for R 3NF ✓

$C \to A$

C: proper subset of CK BC

A: proper subset of CK AB

R is in 3NF but not BCNF

29:19|18

Q Consider the given claims

(i) if every CK of relational schema R is simple CK then R is always in 2NF but may be not in 3NF. $\boxed{\text{True}}$

(ii) if every CK of relational schema R is simple CK then R is always in BCNF. $\boxed{\text{False}}$

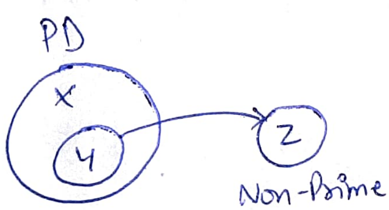
True (iii) if every attribute of relational schema R is prime attribute then relational schema R is always in 3NF but may not be in BCNF.

(iv) if every attribute of relational schema R is prime attribute then relational schema R is always in BCNF $\boxed{\text{False}}$

Which of the given claims are true?

a) (ii), (iii) b) (ii), (iv) c) (i), (iii) d) (i), (iv)

(i) if every CK is simple CK, failure of 2NF never occurs as there won't be any compound CKs to form a partial dependency as shown below



all CKs are simple ⟹ No proper subsets will be formed other than blank / empty subset

↓

empty subset determines non-prime will never occur ⟹ always in 2NF

* Partial Dependy is possible only if there exist at least one compound CK

* 3NF may fail when there are only simple CKs

example R(ABCD)

FD = {	2NF	3NF	
$A \rightarrow B$,	✓	✓	$A \rightarrow SK$
$B \rightarrow CA$,	✓	✓	$B \rightarrow SK$
$C \rightarrow D$	✓	×	
}	Passed	Fails	

CK = {A, B}

prime = {A, B}

non-prime = {C, D}

(67) 03/01/2021

(iii) R(ABC) CK = {AB, BC} FD = {AB → C, C → A}

prime = {A B C} non-prime = { }
(all attrib are prime) (no non-primes)

$X \to Y$, X → SK OR Y → prime → always satisfied in this case, when all attributes are prime

(iv) * this condition may or may not be true for BCNF

03/01/2021 Lec-9

Queries (4 marks) almost all years

** ① Relational (RA) Algebra — Procedural Query

** ② SQL

③ Tuple Relational Calculus (TRC)

② and ③: Non-procedural Queries

Procedural Query: What data to retrieve and How to retrieve it

Non-procedural Queries:
least bothered : How to retrieve
most bothered : What data retrieves from DB Files

* Very frequent questions from RA and SQL come in GATE

Relational Algebra ⇒ uses algebra operators

Basic Operations:

π	projection	ρ	rename
σ	selection	$\cup$	union
$\times$	cross-product	$-$	set difference

Derived operations:

$\cap$	intersection	$\bowtie$	join
$\bowtie_c$	conditional join	⟕	left-outer join
⟖	right-outer join	⟗	full outer join
/ or ÷	division operator		

Relational Calculus < TRC (Tuple RC), DRC (Domain RC)

TRC or RC uses predicate calculus and first order logic formulae

$\wedge, \vee, \sim$ or $\neg, \in, \exists, \forall$ etc

* Prerequisites for SQL : Set theory, Predicate Calculus, First Order logic (all from DM)

Q Consider given relational schemas

Stud (<u>Sid</u> Sname) with <u>Sid</u> as PK and 100 tuples.

Enroll (<u>Sid Cid</u>) with <u>Sid Cid</u> as PK and 50 tuples

How many (max, min) in result of Stud ⋈ Enroll ?

a) (100, 50) b) (100, 0) ✓c) (50, 50) d) (5000, 0)

Concepts

$$R \bowtie S = \underset{\text{all distinct attributes}}{\pi}\left(\underset{\text{equality of common attribute}}{\sigma}(R \times S)\right)$$

example R(ABC) S(BCD)   (R.B = S.B, R.C = S.C)

$$R \bowtie S = \pi_{ABCD}\left(\sigma_{R.B=S.B,\ R.C=S.C}(R \times S)\right)$$

Solution

Stud (<u>Sid</u> Sname) — PK

Sid	Sname
S1	A
S2	B
S3	C
S4	D
S5	B

1 : M

Enroll (<u>Sid Cid</u>) — Not PK (Not unique)

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C3

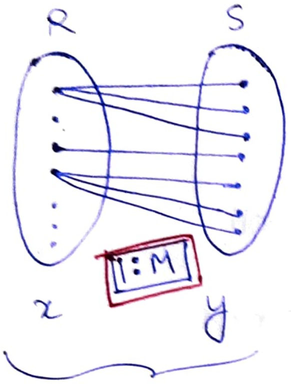
Common = Sid

* every record of Enroll is related to only one record of Stud
* one record of Stud can be related to multiple records of Enroll

Natural Join of Stud and Enroll

$$\text{Stud} \bowtie \text{Enroll} = \underset{\text{projection}}{\pi}\left(\underset{\text{selection}}{\sigma}\left(\underset{\text{cross-product}}{\text{Stud} \times \text{Enroll}}\right)\right)$$

* Selection constraint : equality of common attribute
* projection of all distinct attributes of R



at most y

R has x records

S has y records

R:S have a 1:M Relationship

$R \bowtie S$ produces at most y records

$$\Rightarrow \underset{100 \text{ tuples}}{\text{Stud}} \overset{1 : M}{\bowtie} \underset{50 \text{ tuples}}{\text{Enroll}} \text{ produces } \underbrace{\text{at most 50 records}}_{\text{max} = 50}$$

* if Sid of Enroll is FK referencing Sid of Stud

references

Stud (<u>Sid</u> Sname) — referenced relation

Enroll (<u>Sid Cid</u>) — Referencing Relation

⇒ Then min tuples in Stud ⋈ Enroll = 50 (Assumption : Sid of Enroll is FK)

(because every record of Enroll will be mapped with exactly one record of Stud as Sid is FK in Enroll referencing Sid of Stud)

if no FK present then min = 0

FK		NO FK	
min	max	min	max
50	50	0	50

* By default we consider FK exists unless otherwise stated.

Q $R(\underline{A}BC)$ $A \rightarrow PK$ with n-tuples
$S(\underline{B}DE)$ $B \rightarrow PK$ with m-tuples

Assume every attribute of R is not null attribute. How many (max, min) tuples in result of $R \bowtie S$?

Soln Common Attrib = B, PK for S

PK R(A	B	C)
a1	2	
a2	2	
a3	2	
a4	4	

PK S(B	D	E)
2		
3		
4		

B is not key $\boxed{M:1}$ B is Key

$\boxed{n \text{ tuples}}$ $\boxed{m \text{ tuples}}$

if R.B is FK ref. S.B
$R \bowtie S$
max = n
min = n

No FK Constraint
max = n
min = 0

max records after join $R \bowtie S$ = # of tuples in the relation for which common attribute B is not key

Q $R(\underline{A}BC)$ $A \rightarrow PK$ n tuples
$S(\underline{A}DE)$ $A \rightarrow PK$ m tuples

How many (max, min) tuples in result of $R \bowtie S$?

common attribute is key in both

Soln $R(\underline{A}\ B\ C)$ $S(\underline{A}\ B\ C)$

Common = A key for both tables

R(A	B	C)
2		
3		
4		
6		
⋮		

S(A	B	C)
2		
3		
4		

$\boxed{1:1}$ mapping

Records in $R \bowtie S$
max = min (m, n)
min = min (m, n) with FK
min = 0 if no FK

FK : if S.A is FK ref. R.A

(171)

Q R(<u>A</u>BC) A → PK n tuples

S(<u>D</u>BE) D → PK m tuples

Common attribute is not key in both

How many (max, min) tuples in R ⋈ S ?

Soln

R(A B C) (PK: A)		S(D B E) (PK: D)	
a1	2	d1	2
a2	2	d2	2
a3	2	d3	2

Common = B
not key for any relation

$\boxed{M : M}$ (Many to Many)

Records in R ⋈ S

maximum = mn tuples
minimum = 0 tuples

No FK required here

Q Consider given relational schema R(A B) S(C D) and RA queries

How many of the given queries results in same records set ?

(i) R ∩ S
(ii) R − (R − S)
(iii) R ⋈ S
(iv) R ⋈ $\rho(S)$
$C \to A$ C is renamed to A
$D \to B$ D is renamed to B

✓ (i) (ii) (iv) are same

(i) R ∩ S results in schema having name of first relation

(ii) R − (R − S) = R ∩ S [same as (i)]

~~(iii)~~ R ⋈ S *No common attribute in R, S ⇒

* if no common attrib then ⋈ is equal to cross product

$\boxed{R \bowtie S = R \times S}$
cross-product

(iv) R ⋈ $\rho(S)$ = R ∩ S *AB of R should have same values as AB of S
$C \to A$
$D \to B$

Q Consider given relational schema and RA query

Emp (<u>eid</u> , sal , dno , gen)

$$\pi_{eid}\left(Emp \bowtie_{Sal>S \wedge gen=Female \wedge G=Male} \rho_{I,S,D,G}(Emp) \right)$$

Conditional Join

Which of the following statement is correct for result of given RA Query?

× a) Retrieves eid(s) of female employees whose salary is more than salary of every male employees.

✓ b) Retrieves eid(s) of female employees whose salary is more than ~~some~~ the salary of some male employees.

× c) Retrieves eid(s) of female employees whose salary is more than the salary of every employee.

× d) Retrieves eid(s) of female employees whose salary is more than the salaries of some employees. [correction : some male employees]

Concept ☆ Conditional Join ($\bowtie_c$) OR Natural Join ($\bowtie$)

Condition satisfied for <u>some/any/at least one</u> record(s) of other relation

Solution Conditions : Sal > S $\wedge$ gen = Female $\wedge$ G = Male

Instance 1 > Instance 2

Emp (<u>eid</u> , sal , dno , gen) Emp (I , S , D , G)

Projection ⇑ female ⇑ Male

(173)

a) Emp

	eid	Sal	gen
✗	e1	20	F
✓	e3	40	F
✓	e5	60	F

Emp

I	S	G
e2	30	M
e4	40	M

Projection

eid
e3
e5

* condition given is not meant for every male employee, but at least one male employee

---

Q Consider the given relational schema and RA query.

Emp (eid , Sal, dno, gen)

Conditional join works for some/any/at least one

$$\underbrace{\pi_{eid}\left(\sigma_{gen=female}(Emp)\right)}_{\text{all females (universal set of Females)}} - \pi_{eid}\left(Emp \bowtie_{Sal <= S \wedge gen = Female \wedge G = Male} \rho_{I,S,D,G}(Emp)\right)$$

$\boxed{U - A = A^c}$

→ females whose salary is less than or equal to some male employees

Which of the following statement is correct for the result of given RA query ?

✓ a) Retrieves eid(s) of female employees whose salary is more than the Salary of every male employee ⇒ A^c (answer)

b) Retrieves eid(s) of female employees whose salary is more than the Salary of some male employees

c) Retrieves eid(s) of female employees whose salary is less than or equal to salary of every male employee ⇒ A

d) Retrieves eid(s) of female employees whose salary is less than or equal to salary of some male employees.

R(A...) S(B...)

$\pi_A (R \bowtie_{R.A > S.B} S)$: A values of R which are more than some values of B of relation S

$\pi_A (R) - \pi_A (R \bowtie_{R.A <= S.B} S)$: A values of R which are more than every value of B of relation S

Q Consider given relational schemas and RA query

MadeEasyFaculty (fid rating)

Subject_Allocation (SubjectName fid)

Subject taught by some faculty whose rating is less than or equal to 9

$\pi_{SubjectName} (SubjectAllocation) - \pi_{Subject Name} (Subject Allocation \bowtie \sigma_{rating \leq 9} (MadeEasyFaculty))$

Which is correct statement for result of given RA query

a) Retrieve Subject Names which are taught by only such faculties whose rating is more than 9.

b) Retrieve Subject Names which are taught by some faculty whose rating is more than 9.

c) Retrieve Subject Names which are taught by only faculties whose rating is less than or equal to 9.

d) Retrieve Subject Names which are taught by some faculties whose rating is less than or equal to 9.

∩ (intersection) Records belonging to both relations (common records)

∪ (union) Records from both relations combined

− (set diff) Records of First relation but not second relation

(175)

Q Consider given relational schema and RA query

MadeEasyFaculty (<u>Fid</u> , Fname , Rating)

conditional join

$$\pi_{Fid}(\text{MadeEasyFaculty}) - \pi_{Fid}\left(\text{MadeEasyFaculty} \bowtie_{\text{rating}<R} \rho_{F,N,R}(\text{MadeEasyFaculty})\right)$$

Which statement is correct for the result of given RA query?

a) Retrieves faculties whose rating is more than or equal to rating of some faculties. (this can be directly expressed by $\bowtie_c$)

✓ b) Retrieves faculties whose rating is more than or equal to rating of every faculty. (set difference required to express this)

c) Retrieves faculties whose rating is more than rating of some faculties.

d) Retrieves faculties whose rating is more than the rating of every faculty. (modification in condition like rating ≤ R required)

less than

Instance 1

MEF 1

	Fid	Fname	Rating
✓	f1		8
✓	f2		10
✓	f3		12
✗	f4		14

Instance 2

MEF 2

Fid	Fname	Rating
f1		8
f2		10
f3		12
f4		14

Cond. Fid = {f1 f2 f3}

Fid rating less than at least one FID of MEF 2 of MEF 1

all Fid − Cond. Fid = {f4}

all Fid whose rating is more than or equal to rating of every faculty

R(A...)

What to Retrieve

★ Retrieve maximum value of attribute A : values of A which are ≥ every A of R ≡ {all A values of relation R} − {A values of relation R which are < some A values of R}

(Set difference; some → $\bowtie$ or $\bowtie_c$)

In SQL,

max() function → What to Retrieve

$$\equiv \boxed{\pi_A(R) - \pi_A\left(R \underset{A<A1}{\bowtie} \rho_{A\to A1}(R)\right)}$$

} How to Retrieve

★ Retrieve minimum value of A attribute : values of A which are ≤ every A of R ≡ {all A values of relation R} − {A values of relation R which are > some A values of R}

(some → $\bowtie_c$)

In SQL,

min() function

$$\equiv \boxed{\pi_A(R) - \pi_A\left(R \underset{A>A1}{\bowtie} \rho_{A\to A1}(R)\right)}$$

Lec-11 03/01/2021

Q Consider given relational schema R(AB) and following queries "To retreive maximum values of attribute A"?

my answer = C

Query (i) $\pi_A(R) - \pi_A\left(R \underset{A<A1}{\bowtie} \rho_{A\to A1, B\to B1}(R)\right)$ ✓

Query (ii)

```
SELECT DISTINCT A
FROM R
WHERE A = (SELECT max(A)
           FROM R);
```

✓

Which query is correct representation for given specification?

a) only query (i) correct b) only query (ii) correct

c) both query (i) and (ii) correct d) both are wrong

177

RA $\pi_{A}(R) - \pi_{A}\left(R \bowtie_{A<A_1} \rho_{\substack{A \to A_1 \\ B \to B_1}}(R)\right)$

equivalent SQL

```
SELECT A
FROM R
EXCEPT
SELECT T1.A
FROM R T1, R T2
WHERE T1.A < T2.A;
```

} All A values

EXCEPT ⟶ Set Difference

} $A < A_1$ Values

] SQL Query to retrieve max Value without using aggregate functions

---

Q Consider the given record sets of relations R and S

R

A	B	
10	2	4
20	3	3
60	4	1
80	5	0
40	6	2

S

C	D
80	7
10	8
40	9
60	10
20	11

```
Select *
FROM R
WHERE (SELECT COUNT(*)
       FROM S
       WHERE R.A < S.C) < 2;
```

Corelated Nested Query

if query executes over R and S data, how many tuples are present in result of given SQL query?

Concepts

Queries Used

FROM
WHERE
HAVING

Nested Query without Correlation

⟹ Inner query is independent of outer query

eg

```
SELECT eid
FROM Emp
WHERE Sal = (SELECT max(sal)
             FROM Emp);
```

computed first ⇓ (computes only once)

outer query uses result of inner query for its computation

(Costly to Execute)

Nested Query with correlation

inner query is not independent

⟹ Inner query's where clause uses attribute of outer query

R(A B) S(C D)

```
SELECT A
FROM R
WHERE (SELECT Count(*)
       FROM S
       WHERE R.A < S.C) = 1;
```

Computes for each record of outer query ⇓

Solution

For each record of R inner query is executed based on the result of this execution, the outer query is executed, this is repeated till are records of R are exhausted.

32:18

R	A	B	For each record of R count(*)			
×	10	2	4	<	2	×
×	20	3	3	<	2	×
✓	60	4	1	<	2	✓
✓	80	5	0	<	2	✓
×	40	6	2	<	2	×

} 2 tuples present in Result

Q Consider given relational schema. Emp(Eid, Sal)

Which of the following queries are correct representations to retrieve eid(s) who gets second highest salary?

✓ Query 1]

```
③ SELECT eid
  FROM Emp
  WHERE Sal = (SELECT max(Sal)
            ② FROM Emp
              WHERE Sal < (SELECT max(Sal)
                        ① From Emp));
```

2nd max salary less than the max salary

max salary

Bottom-up Approach

Nested Query without correlation

✓ Query 2]

```
SELECT eid
FROM Emp T1
WHERE (SELECT COUNT(DISTINCT SAL)
       FROM Emp T2
       WHERE T1.Sal < T2.Sal) = 1;
```

Nested Query with correlation

by changing these conditions we can find Kth highest salary.

a) Only query 1 is correct

b) Only query 2 is correct

c) both query 1 and 2 are correct ✓

d) both query 1 and 2 are incorrect

query 1]

eid	sal
e1	10
e2	80
e3	30
e4	20
e5	80
e6	30

$\Rightarrow$

e1	10
e3	30
e4	20
e6	30

$\Rightarrow$ max = 30

Second max salary
↓

e3	30
e6	30

] Result

query 2]

T1

eid	sal
e1	10
e2	80
✓ e3	30
e4	20
e5	80
✓ e6	30

T2

eid	sal
e1	10
e2	80
e3	30
e4	20
e5	80
e6	30

× I. T1.sal < T2.sal

$10 < \underbrace{80, 30, 20, 80, 30}_{5}$

⇓

Distinct (sal)

⇓

$\underbrace{80, 30, 20}$

⇓ Count

× $3 \neq 1$

× II. 80 < No record

⇓ count

× $0 \neq 1$

✓ III. $30 < \underbrace{80, 80}$

⇓ distinct (sal)

80

⇓ count

✓ $1 = 1$

× IV. $20 < \underbrace{80, 30, 80, 30}$

⇓ distinct(sal)

$\underbrace{80, 30}$

⇓ count

× $2 \neq 1$

× V. 80 < No Record same as II

✓ VI. same as III

Q Consider given relational schema Emp(Eid, Ename, SupID)
which of the following queries are correct representation to retrieve names of the supervisors

Query 1] $\pi_{Ename}(Emp \bowtie_{Eid = S} \rho_{I,N,S}(Emp))$

Query 2]
```
SELECT DISTINCT T1.Ename
FROM Emp T1, Emp T2
WHERE T1.SupID = T2.Eid;
```

a) only query 1 is correct
b) only query 2 is correct
c) Both query 1 and 2 are correct
d) both queries are wrong

Solution

Emp Instance 1

eid	Ename	SupID
e1	A	NULL
e2	B	e1
e3	C	e1
e4	D	e2

Required Output

A
B

Both are names of supervisors

Instance 2

eid (I)	ename (N)	supID (S)
e1	A	NULL
e2	B	e1
e3	C	e1
e4	D	e2

For query 1]

Eid = S

AB

Correct

For query 2]

Instance 1 (T1)

eid	ename	supID
e1	A	Null
e2	B	e1
e3	C	e1
e4	D	e2

Instance 2 (T2)

eid	ename	supID
e1	A	Null
e2	B	e1
e3	C	e1
e4	D	e2

BCD

Incorrect

T1.SupID = T2.eid

 Lec-12 03/01/2021

Q Which of the following SQL query is valid?
Assume relational schema R(ABC)

a) SELECT B
FROM R
GROUP BY (A); ✗

b) SELECT A,B
FROM R
GROUP BY (A); ✗

✓ c) SELECT A, Count(C)
FROM R
GROUP BY A;

aggregate function is allowed

d) SELECT *
FROM R
GROUP BY A; ✗

Concept ☆

if $\boxed{\text{GROUP BY}}$ clause is used in the query then there are some restrictions on the SELECT clause

① * every attribute of GROUP BY clause must be in the SELECT clause

* if GROUP BY clause is used in a query

I. Every attribute of GROUP BY clause must be in SELECT clause

II. Can use aggregate functions in SELECT clause and this aggregate function computes aggregation of each group

III. Not allowed to select any other attribute in select clause

Q Consider the given DB Tables

R

A	B
2	4
4	6
6	5
8	9

S

C	D
7	4
4	8
2	3
5	7

(i) SELECT *
FROM R
WHERE B > ANY (SELECT C
FROM S
WHERE D > 10);

4 → False
6 → F
5 → F
9 → F

⇒ No tuples in result (empty result)

Normal Nested Query

(ii) SELECT *
FROM R
WHERE B > ALL (SELECT C
FROM S
WHERE D > 10);

4 > T
6 > T
5 > T
9 > T

non-empty result

$\boxed{\text{4 tuples in result}}$

(iii) SELECT *
FROM R
~~WHERE B > ALL~~
WHERE EXISTS (SELECT COUNT(*)
FROM S
WHERE R.B > S.C and S.D > 10);

$\boxed{\text{4 tuples}}$

Correlated Query

returns true for all records

No record of S > 10 → 0 output

Count(*)
0

non-empty result
⇑
inner query result

How many of the given SQL queries result in a non-empty record set if executed over the given data of R & S tables? Ans: (ii) (iii) returns non-empty results

(i) inner query result is empty B > ANY (empty)
(ii) inner query result is empty B > ALL (empty)

Concepts

ANY () ALL ()

☆ ANY

comparison
↓
X > ANY

Y
4
6
8
10

} inner query result

↑
this condition should be true for any value of Y set

* if the comparison of X values with any inner query result values is true then ANY () function returns true

if X = 2 2 > ANY

Y
4
6
8
10

returns False as X is smaller than all Y values

if X = 5 5 > ANY

Y
4
6
8
10

returns True as 5 > 4, at least one value of Y satisfies the condition

☆ $\boxed{\text{EXISTS Function returns true if inner query result is non-empty, false otherwise}}$

(183)

X > ANY $\boxed{Y}$ — empty

ANY( ) behaves as ∃ → there exists at least one

ANY( ) → returns true if at least one value satifies the given predicate condition

* because the set is empty no value satisfies the given condition, hence, it returns False as answer

ANY Function expects at least one true case for returning true, otherwise it's false

---

ALL

X > ALL

Y
4
6
8
10

} inner query result

* ALL function expects at least one false to return false, otherwise it's true

ALL( ) behaves as ∀ → for all values

ALL( ) → returns True if all values satisfy the given condition, even if one value fails for the given condition, ALL returns False

X > ALL $\boxed{Y}$ — empty

* ALL Function with empty set returns True (not even one value fails for the predicate condition) [as there are no values at all]

if at least one value fails the condition => returns False

if all values pass the condition => returns True

Q Consider given relational schema WorksFor (<u>eid pid</u>)
Which of the following queries is the correct representation of "to retrieve eid(s) who work for at least 2 projects"?

✓ Query 1]

```
SELECT DISTINCT T1.eid
From WorksFor T1, WorksFor T2
WHERE T1.eid = T2.eid and
      T1.pid <> T2.pid ;
```

↑ not equal to

✓ Query 2]

```
SELECT eid
FROM Emp T1
GROUP BY eid
HAVING COUNT(*) >= 2 ;
```

a) only query 1 is correct

b) only query 2 is correct

✓ c) both queries are correct

d) both queries are incorrect

Query 1]

	Eid	Pid
✓	e1	p1
✓	e1	p2
✓	e1	p3
✓	e2	p1
✓	e2	p2
×	e3	p1

Eid	Pid
e1	p1
e1	p2
e1	p3
e2	p1
e2	p2
e3	p1

~~8 Records in result of query~~

Distinct T1.eid = {e1, e2}   query 1 is correct for given specification

Query 2]

	Eid	Pid	
Group 1	e1	p1	count(*) = 3
	e1	p2	
	e1	p3	
Group 2	e2	p1	count(*) = 2
	e2	p2	
Group 3	e3	p1	count(*) = 1

Groups 1 and 2: output / result

(185) Q Consider the given DB table and SQL query

R(ABC)

R

A	B	C
5	Null	6
4	8	4
10	8	Null
Null	6	3

```
SELECT *
FROM R
WHERE (A>=5) and (B>C OR B>5);
```

How many records in result of given SQL query?

Concepts

* NULL ⟹ used to represent unknown values or ~~unexis~~ non-existing values

* Null is non-zero, not empty string
* RDBMS guideline : Null is set of ASCII values assigned by DBMS such that No two Null values are same in one DB table.

R

A	B
4	8
8	Null
10	Null
15	5

* a DB file is allocated for this
* Some space is allocated for each record

→ Set of ASCII values are stored in the file ~~...~~ to denote Null

Values for both Nulls are different

* Comparison with Null Value is treated as unknown (unknown means neither True nor False)

example

```
SELECT *
FROM R
WHERE B>6;
```

8 > 6 True

Null > 6 Unknown

Null > 6 Unknown

5 > 6 False

(Unknown, Unknown, False) discarded

Result

A	B
4	8

~~WHERE clause~~

WHERE Clause discards a record if result of condition following where clause is False or Unknown

example where clause with 2 conditions

```
SELECT *
FROM R
WHERE condition_one and/or condition_two
```

		AND	OR
p	q	$p \wedge q$	$p \vee q$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

* For and/or operations with unknown

~~True and unknown~~

OR	T	F	U
T	T	T	T
F	T	F	U
U	T	U	U

AND	T	F	U
T	T	F	U
F	F	F	F
U	U	F	U

True ≡ 1 False ≡ 0 Unknown ≡ 1/2 (assumption)

OR

x	y	$x \vee y \equiv \max(x,y)$
0	0	0
0	1	1
1	0	1
1	1	1
1/2	0	1/2
1/2	1	1
1/2	1/2	1/2

AND

x	y	$x \wedge y \equiv \min(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1
1/2	0	0
1/2	1	1/2
1/2	1/2	1/2

(187)

Solution

X I. A B C

5 Null 6

$A \geq 5$ T

$B > C \Rightarrow Null > 6$ U

$B > 5 \Rightarrow U$

$\Rightarrow$ T and (U or U)

$\Rightarrow$ T and U

$\Rightarrow$ U

$\Downarrow$

discarded

OR = max(x,y)

AND = min(x,y)

where clause discards the result which is False or Unknown

X

II. A B C

4 8 4

$A \geq 5$ F

$B > C$ T

$B > 5$ T

$\Rightarrow$ F and (T or T)

$\Rightarrow$ F and T

$\Rightarrow$ F $\Rightarrow$ discarded

✓ III. A B C

10 8 Null

$A \geq 5$ T

$B > C$ U

$B > 5$ T

$\Rightarrow$ T and (U or T)

$\Rightarrow$ T and T

$\Rightarrow$ T $\Rightarrow$ accepted by WHERE clause

X

IV. A B C

Null 6 3

$A \geq 5$ U

$B > C$ T

$B > 5$ T

$\Rightarrow$ U and (T or T)

$\Rightarrow$ U and T

$\Rightarrow$ U $\Rightarrow$ discarded

Lec-13 03/01/2021

Q Consider given relational schema R(X,Y), which is used to store graph edges. Each directed edge (x,y) of graph is a record in relation R.

Which of the given RA query cannot express in constant length

a) Retrieve vertices which forms self loop (T)

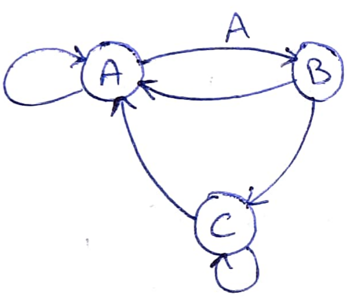
b) Retrieve vertices which forms loop with at most two vertices (T)

c) Retrieve all adjacent vertices of given vertex

✓ d) Retrieve all vertices which ~~can~~ are reachable from given vertex.

Solution

Sample Graph



X	Y
A	A
A	B
B	A
B	C
C	C
C	A

Vertices with

✓ a) self loop $X = Y$

can be retrieved ✓

$$\pi_X\left(\sigma_{X=Y}(R)\right)$$

This query retrieves all vertices having self-loops

✓ b) loops with at most two vertices

at most two vertices → Selfloop, 1, 2

	X	Y
t_1	A	B
t_2	B	A

$t_1.X = t_2.Y$ for loop with 2 vertices

$\underbrace{\hphantom{xxxxxxxxxxxxxxxx}}$ 2 instances of R needed

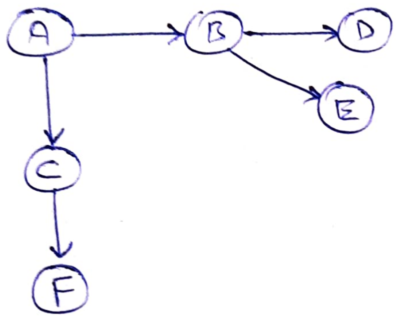
$$\pi_X\left(R \underset{\substack{X=Y1 \\ \wedge Y=X1}}{\bowtie} \rho_{\substack{X\to X1 \\ Y\to Y1}}(R)\right)$$

(189) c) Adjacent vertices of vertex B

$$\pi_{Y}\left(\sigma_{X=B}(R)\right)$$ B → given vertex

d) vertices reachable from given vertex

example



B, C are reachable from A at one edge length

D E F are reachable from A at 2 edge lengths

and so on... has n edge length reachability

* Reachability can be of any number of edge length

* reachable by one edge length ⇒ adjacent vertices of given vertex

1 instance — one edge length — $\pi_{Y}\left(\sigma_{X=A}(R)\right) \cup$

2 instances — two edge length — $\pi_{T_2.Y}\left(\rho(R,T_1) \bowtie_{T_1.X=A \wedge T_1.Y=T_2.X} \rho(R,T_2)\right) \cup$

3 instances — three edge length

⋮

n instances — n edge length — $\pi_{T_n.Y}\left(\sigma_{T_1.X=A \wedge T_1.Y=T_2.X \wedge T_2.Y=T_3.X \wedge \cdots \wedge T_{n-1}.Y=T_n.X}(T_1 \times T_2 \times \cdots \times T_n)\right)$

* length is not known and not fixed
* fails to express reachable vertices because it requires length of reachability
* Such queries are called as Not constant length RA query

Q Emp(Eid , Ename , rating) Works (Eid Pid) Projects (Pid Pname)

Consider given TRC query

$\{ T / \exists T_1 \in \text{Works} \ \exists T_2 \in \text{Project}$

$(T_1.Pid = T_2.Pid \text{ and } (T_2.Pname = DB \vee T_2.Pname = OS)$

$\wedge T = T_1.eid) \}$

What is the result of TRC query?

Concepts ☆ TRC uses First Order Logic

TRC is not very frequent in exam

$\exists$ in TRC is equivalent to $\bowtie_c$ in RA

Solution $\underline{\exists T_1 \in \text{Works}} \ \underline{\exists T_2 \in \text{Project}}$

$\Downarrow$

$\sigma(\text{Works} \times \text{Project})$

let's say we rename works to T_1 and Project to T_2

$$\pi_{T_1.eid}\left(\sigma_{\substack{T_1.pid = T_2.pid \\ \wedge (T_2.Pname = DB \vee T_2.Pname = OS)}}\left(\rho(\text{Works}, T_1) \overset{\text{cross product}}{\times} \rho(\text{Project}, T_2)\right)\right)$$

$\cancel{\wedge (T = T_1.eid)}$

Retrieves (employee ids) eid(s) of employees who works for some DB project or some OS project

Free Variable / Result Variable

* TRC Query Format ☆ $\{ T \mid P(T) \}$ This T is used for projection

retrieving set of records T which satisfy P(T)

(91)

Q Consider given relational schema WorksFor (eid pid). Write TRC query to retrieve eid(s) of employees who work for at least two projects?

Solution

Works

	eid	pid
t_1	e_1	P_1
t_2	e_1	P_2

Self cross-product

= ≠

T_1

eid	pid
e_1	P_1
e_1	P_2
e_2	P_2

T_2

eid	pid
e_1	P_1
e_1	P_2
e_2	P_2

Using RA:

$$\pi_{T_1.eid}\left(\sigma_{T_1.eid = T_2.eid \wedge T_2.Pid \neq T_1.Pid}\left(\rho(T_1, Works) \times \rho(T_2, Works)\right)\right)$$

TRC Query:

$$\{T \mid \exists\, T_1 \in Works\_For\ \exists\, T_2 \in Works\_For\ (T_1.eid = T_2.eid \text{ and } T_2.Pid \neq T_1.Pid \wedge T = T_1.eid)\}$$

TRC Queries for $\cup, \cap, -$

$R \cup S = \{T \mid T \in R \vee T \in S\}$

$R \cap S = \{T \mid T \in R \wedge T \in S\}$

$R - S = \{T \mid T \in R \wedge \neg(T \in S)\}$

Important Points

* Unsafe TRC query: a TRC query which results in infinite record set

example $\{T \mid \neg T \in R\}$ Retrieving set of records which do not belong to R ⇒ infinite set of records

Expressive Power Comparison

$$\left\{ \begin{array}{l} \text{basic RA} \\ \text{queries} \end{array} \right\} = \left\{ \begin{array}{l} \text{Safe TRC} \\ \text{queries} \end{array} \right\}$$

The expressive power of basic RA queries is equivalent to expressive power of safe TRC queries

* every query in safe TRC can be expressed as basic RA query and vice-versa

* Basic RA query : queries which can be expressed using
$$\{ ⟗, \pi, \sigma, \times, \rho, \cup, -, \cap, \bowtie, \bowtie_c, ⟕, ⟖, / \}$$

* RA always provides result with distinct tuples
* Queries which can't be expressed using basic RA/ Safe TRC
  * Count of vertices reachable from given vertex (count of tuples)
    OR
    count of attribute values
  * Sum of attribute values
  * Average of attribute values
  * Ordering of records
  * Unable to represent RA query to prove duplicate records

Lec-14 04/01/2020 ER-Model (almost guaranteed 2m question every year)

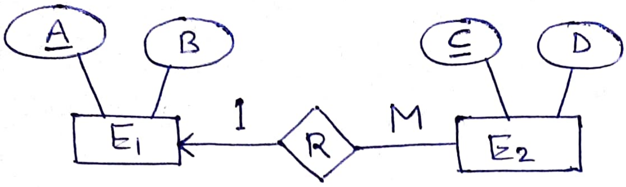
1. Finding CK set of relationship set
2. Find minimum RDBMS Tables required for given ER Diagram

* 1:M, M:1, M:N, 1:1 } Binary Mapping OR Binary Relationship Set
* Self-Referential
* Weak Entity Set

Default: INF

3. Find minimum RDBMS Tables for given ER Diagram which satisfy some NF

(193) Q Consider the following ERD

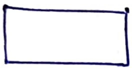


What is the CK of relationship R? if Relationship set R is related b/w E_1 and E_2 with 1:M mapping?

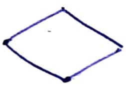
a) {A} b) {C} c) {AC} d) {A, C}

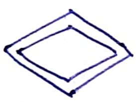
Concepts

* ER Diagram Symbols

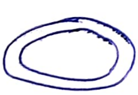

Entity or Strong Entity


Weak Entity


Relationship


Weak Relationship


attribute


Multi-Valued attribute

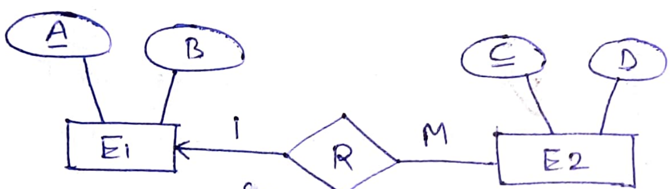
* Types of attributes

1. Simple attribute – can't be divided further (roll number)
2. Composite attribute – can be divided into components (addr)
3. SVA (Single Valued Attribute) – takes up only a single value (enrollment #)
4. MVA (Multi-Valued Attibute) – takes up more than one value (phone #)
5. Derived attribute – can be derived from other attributes (age)
6. Key attribute – uniquely identifies each entity (student id)

Terminology

* Entity - an object with either physical existence or conceptual existence.
* Entity Set - group of similar kind of entities
* Relationship - association among two or more entities
* Relationship Set - a set of relationships of same type
* Degree of Relationship Set - number of different entity sets participating in a relationship, it can be unary, binary or n-ary type.
* Total Participation - each entity in entity set must participate
* Partial Participation - each entity in entity set may or may not participate
* Weak Entity - some entity for which key attribute can't be defined

Solution



* Attributes of relationship are the key attributes of participating relations
* Key attributes of related entities are the attributes of the relationship

refers   CKey   refers

E1

A	B
a1	b1
a2	b1
a3	b2
a4	b3

R

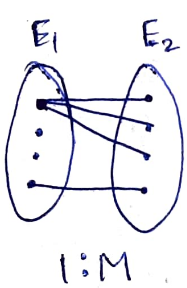
A	C
a1	c1
a1	c2
a2	c3

E2

C	D
c1	d1
c2	d1
c3	d2
c4	d3

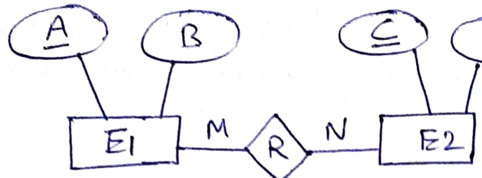
1:M

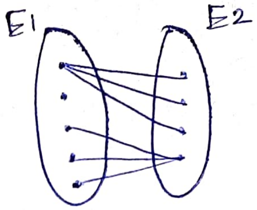
* Here, C should be key of R



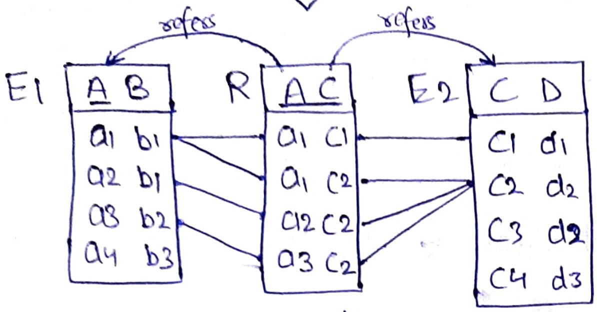
* each element of E1 can pair with many (M) elements of E2 but each element of E2 can pair with at most one (1) element of E1
* For 1:M mapping CK of relationship set R which has 2 attributes A, C is ⊆ (Key of Relationship Set is same as key of RHS Entity's key)

* Based on required mapping b/w entity E1 and E2 we need to design CK of relationship set



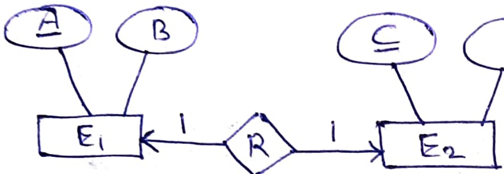


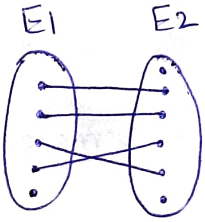
M:N



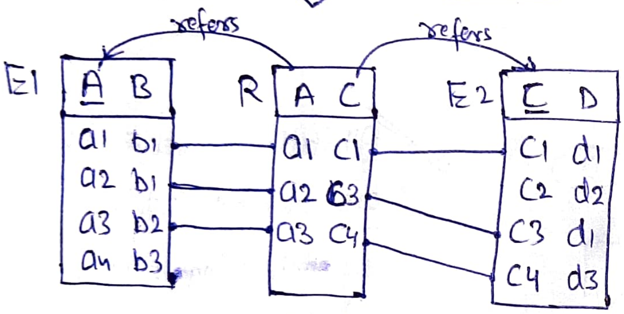
M:N

* For M:N mapping CK of relationship R is combination of {AC} i.e. Combination of keys of both E1 and E2





1:1



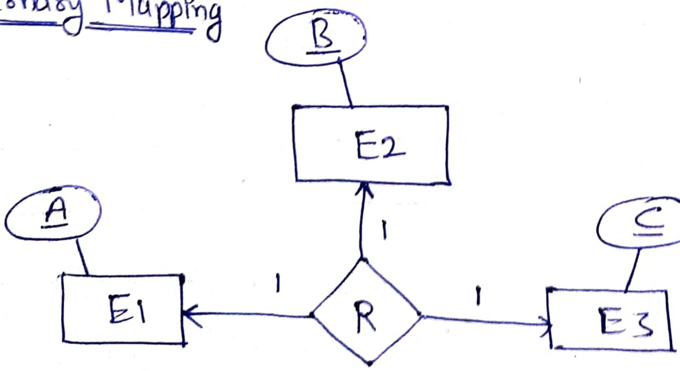
1:1

* each element of E1 and E2 is allowed to map with atmost one element of other entity.

* No two records with same a or same c in R

* For 1:1 mapping between E1 and E2, CKs of Relationship R(AC) are {A, C}

Ternary Mapping



R (ABC)

CKs of R = {A, B, C}

Binary Mapping

4 possible mappings

(i) 1:M (ii) M:1

(iii) M:N (iv) 1:1

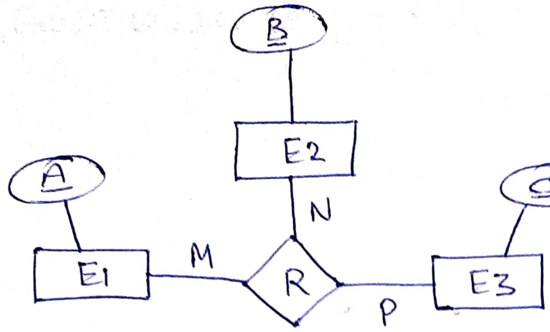
Ternary Mapping

(i) 1:1:1 (ii) 1:1:M

(iii) 1:M:1 (iv) 1:M:N

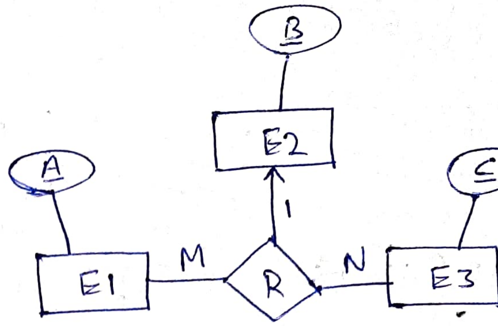
(v) M:1:1 (vi) M:1:N

(vii) M:N:1 (viii) M:N:P



R(A B C)

CK of R = {ABC}



R (ABC)

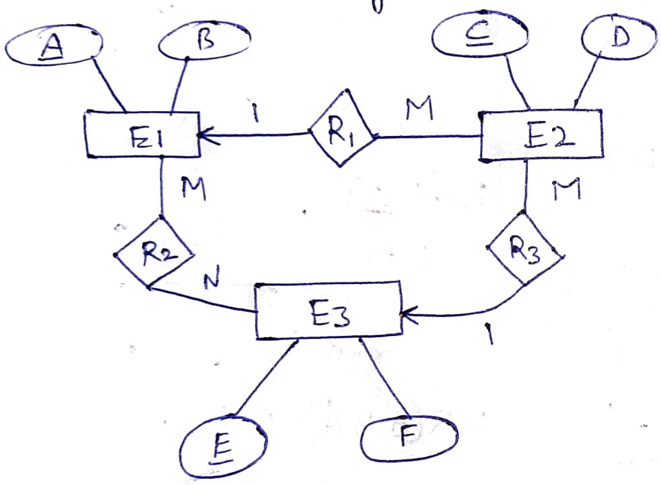
CK of R = {AC}

* one side many ⇒ CK is key of that entity
* two side many ⇒ CK is combination of keys of those two entities

---

Q Consider the following ERD [every year 2m question comes]

How many minimum RDBMS tables are required for given ER Diagram ?

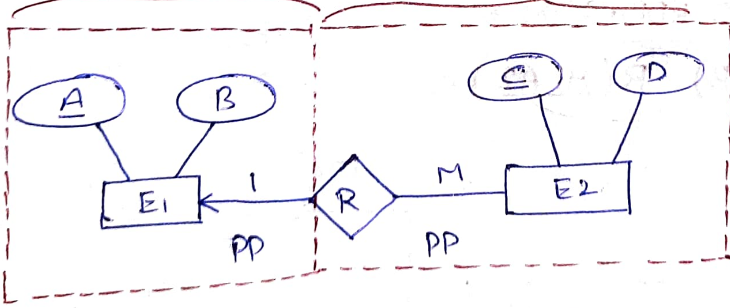


one Table | one Table (R can combine with E2 as both of them have same key C)

Concepts

CK = {C}

2 Tables



PP - Partial Participation

TP - Total Participation

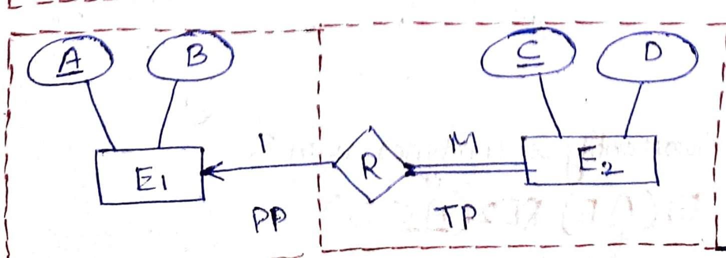
→ minimum 2 RDBMS tables

E1 (A B) E2R (C D A)

FK refers

CK = {C}

2 Tables



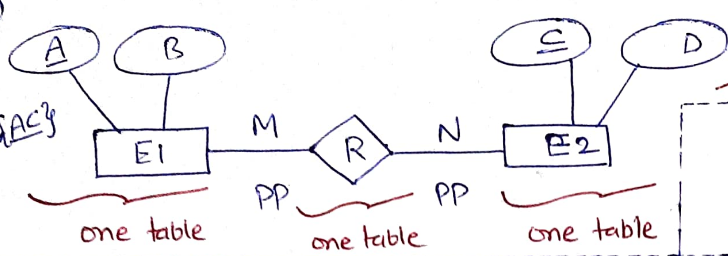
one table | one table

→ E1 (AB)

E2R (C D A)

FK (Not Null because of total P.)

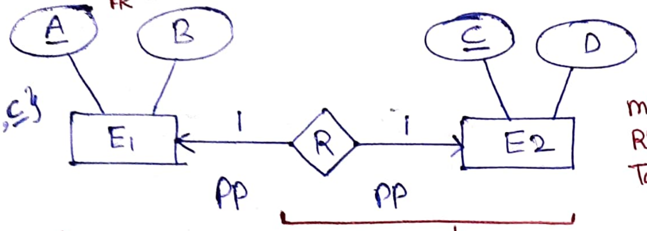
CK = {AC}



minimum 3 RDBMS tables
E1(AB) E2(CD) R(AC)

① E1R(ABC) E2(CD) ② E1(AB) E2(CDA)
FK FK

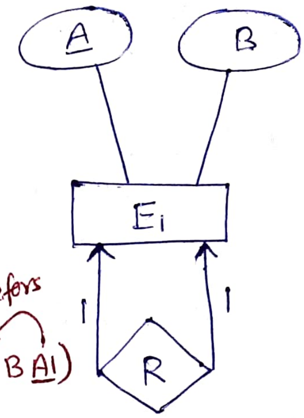
CK = {A, C}



min 2 RDBMS Tables

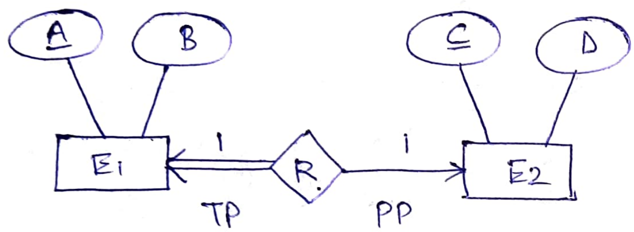
or combine R, E1

either combine R, E2



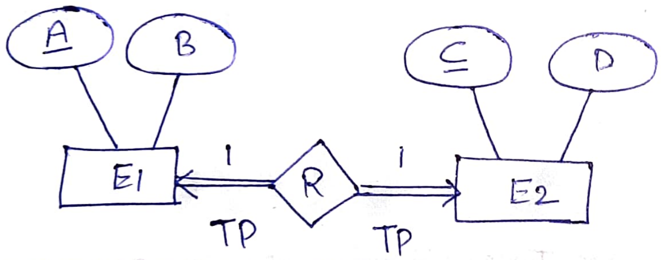
refers
E1R(ABA1)

Self Referential
(Recursive Entity set)

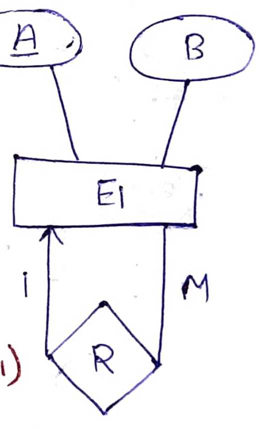


min one RDBMS Table for both

E1E2R(ABCD)



min one RDBMS Table for both

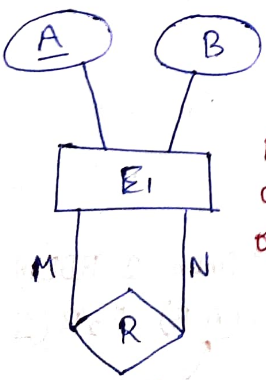


R(A1 A2)
1 M

refers
E1R(ABA1)

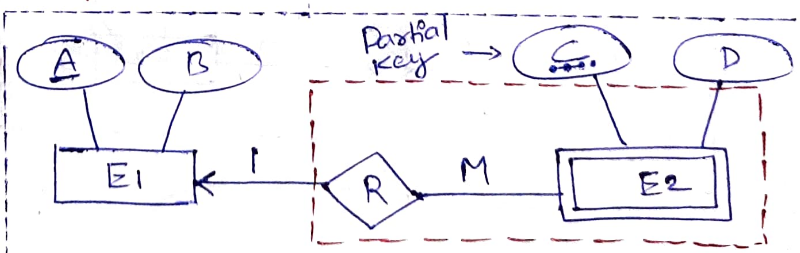
Self Referential
(Recursive entity set)

min 2 RDBMS Tables
E1(AB) R(A1 A2) both FKs
refers refers



A1 A2 are subset of A attrib

Self Referential
(Recursive Entity Set)

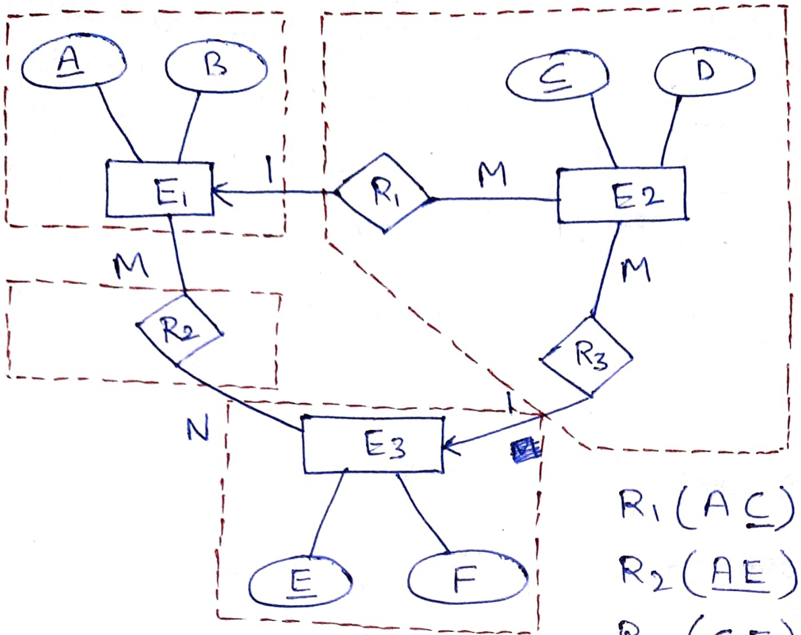


Weak entity set combines with R
E1(AB) RE2(ACD)
FK
refers

min 2 Tables

*Weak Entity set always have 1:M mapping

Solution



E1 (A B)

refers

E2R1R3 (C D A E)

refers

E3 (E F)

refers

R2 (A E)

refers

R1 (A C)

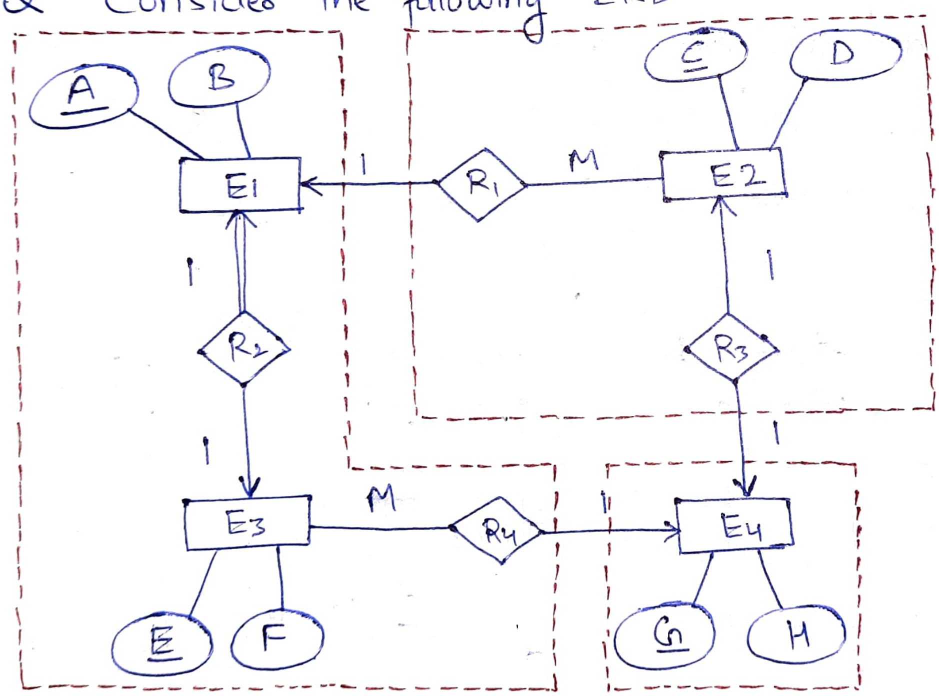
R2 (A E)

R3 (C E)

4 Tables

CS21GBHA619

Q Consider the following ERD



R1 (A C)

R2 (A E)

R3 (C G)

R4 (E G)

3 Tables

E1R2E3R4 (A B E F G)

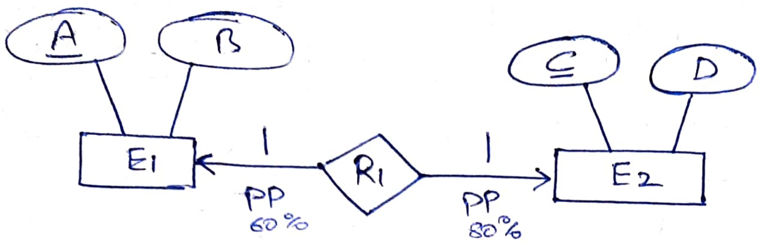
refers

E2R1R3 (C D A G)

E4 (G H)

refers

Q Consider the following ERD



if 60% participation at E1 end and 80% participation at E2 end. Which is the best possible RDBMS design for given ERD?

a) E1 E2 kept separate with FK at E1 } feasible

~~b)~~ E1 E2 " " " " at E2 } feasible

× c) E1 E2 merged into a single table with No FK

× d) E1 E2 kept separate with FK. at both. E1 E2

Solution * PP at both sides => No chance to merge into single table

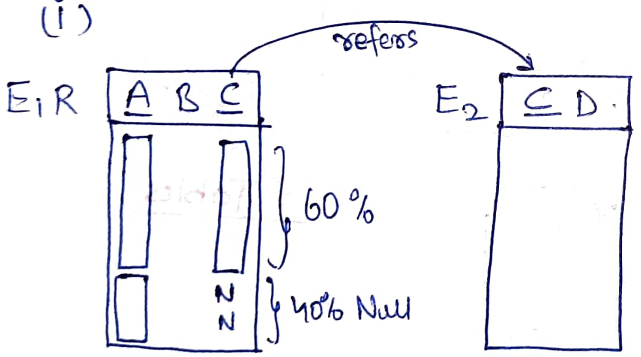
* No chance of FK on both tables

2 possibilities in case of 1:1 with PP on both ends

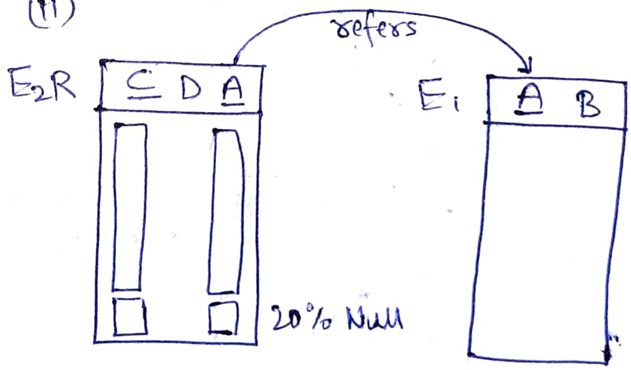
(i) Combine E1 and R with FK at E1 referencing E2

(ii) Combine E2 and R with FK at E2 referencing E1

(i)

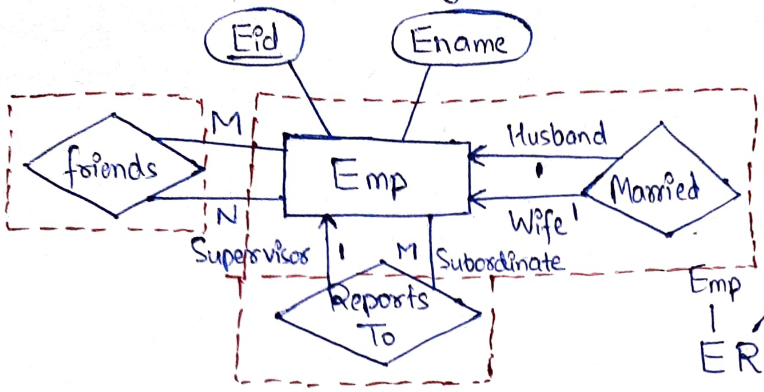


(ii)



* better option as it has less null values

Q Consider the following ERD



How many minimum relational tables are required for given ERD?

Emp ReportsTo Married

ERM (<u>Eid</u> Ename Sup W)

refers

refers

friend (<u>X Y</u>)

~~Emp (Eid Ename)~~ refers

~~Emp Married (Eid Ename Spouse)~~

refers

~~Emp ReportsTo (Eid Ename Sup)~~

* let X and Y be attributes of friends ~~table~~ relationship
* let H and W be attributes of Married relationship
* let Sup and Sub be attributes of ReportsTo relationship

Friends (<u>X Y</u>) M:N with Emp itself

Married (<u>H</u> <u>W</u>) 1:1 with PP on both ends with Emp itself

ReportsTo (Sup <u>Sub</u>) 1:M with Emp itself

Emp (<u>Eid</u> Ename) will combine with Married and Reports To
1:1 1:M

Friends (<u>XY</u>) is a separate table

refers

EMR (<u>Eid</u> Ename <u>W</u> Sup)

refers

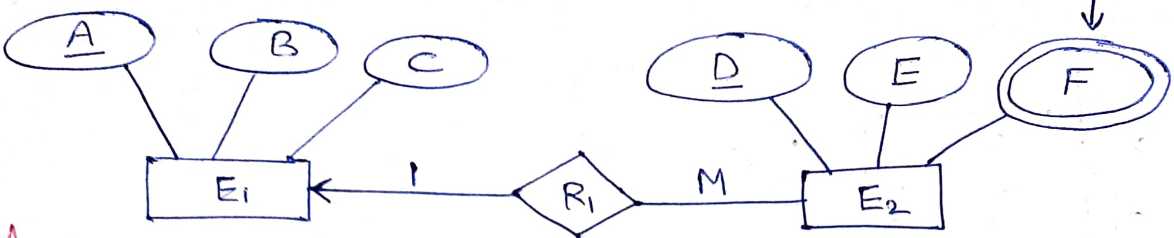
Friends (<u>XY</u>)

* one Supervisor can have multiple Subordinate
* one wife can have exactly one husband

$\boxed{\text{Minimum 2 Tables}}$

(201) Q Consider the following ERD

Multi Valued Attribute



How many minimum relational tables are required for the given ERD which satisfy BCNF?

Solution

FD Sets

$E_1(\underline{A}\ B\ C)$ $\quad$ $R_1(A\ \underline{D})$ $\quad$ $E_2(\underline{D}\ E\ F)$

$FD = \{A \rightarrow BC, \quad D \rightarrow A, \quad D \rightarrow E\}$

$D \rightarrow F$ is invalid because F is MVA

* Because of 1:M R_1 and E_2 combines

$R_1E_2(\underline{D}\ E\ F\ A)$ $\xrightarrow{\text{refers}}$ $E_1(\underline{A}\ B\ C)$

$D \rightarrow E$

$D \rightarrow A$

} min 2 Relational Tables satisfies only 1NF

to convert MVA to SVA

design CK properly

if $\underline{\underline{DF}}$ is CK

then $\left.\begin{array}{l} D \rightarrow E \\ D \rightarrow A \end{array}\right\}$ Partial Dependencies present $\Rightarrow$ Not 2NF

* to bring in 2NF, remove PDs by decomposition of E_2R_1

(i) PD attributes in one table $E_2R_1(\underline{D}\ E\ A)$

$D \rightarrow EA$

(ii) Left attributes in other table $E_2(\underline{DF})$

$E_1(\underline{A}\ B\ C)$

} 3 Tables

2NF ✓

3NF ✓

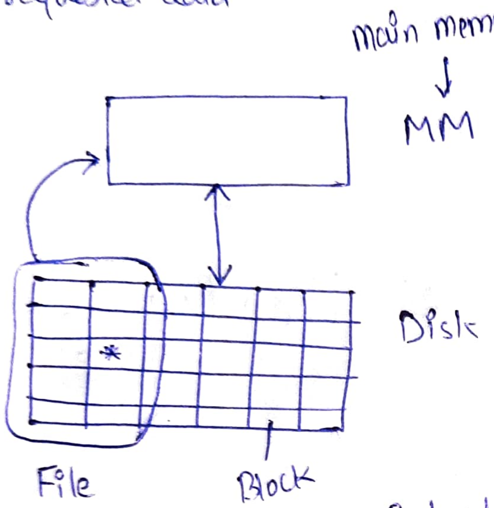
BCNF ✓

Lec-15 04/01/2021 INDEXING

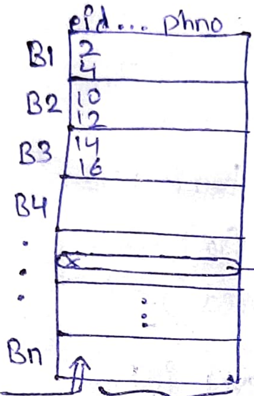
NAT type question comes from this section

* I/O Cost OR Access Cost

of disk blocks which travel from disk to main memory in order to retrieve requested data



* I/O Cost without index



we need to access this record based on ordered field

Ordered field (Data Records are physically ordered based on this field)

unordered fields (all fields/attributes other than ordered field/attribute).

DB File Stored in Disk having n number of Blocks

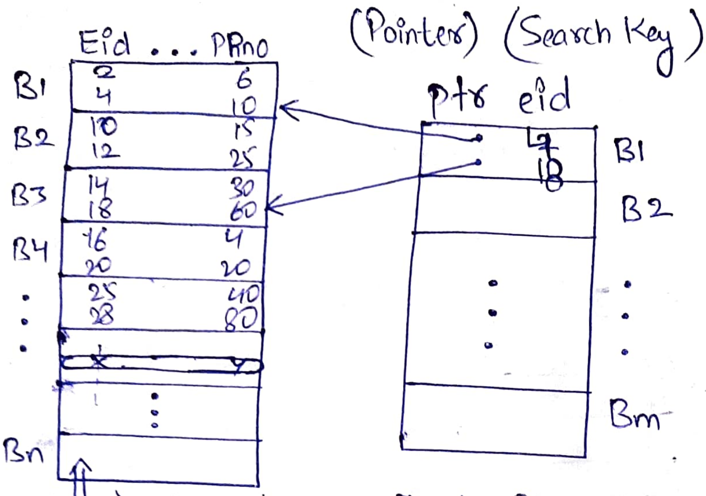
I/O Cost (without Index)

- Based on Ordered Field: $\lceil \log_2 n \rceil$ blocks — transfer a part of File
- Based on Unordered Fields: n blocks (worst case) — transfer entire File

* Indexing is used to reduce the I/O Cost

(203)

Indexing



Ordered Field

Unordered Field

Index ~~Block~~ File
(Index file is always ordered)

I^{st} level index

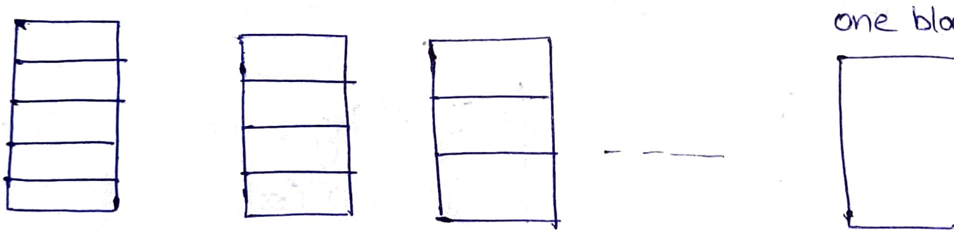
I/O Cost

$\lceil \log_2 m \rceil$ gives key pair of required search key, we need to access one more data block

$\Rightarrow \lceil \log_2 m \rceil + 1$ block access cost

* To reduce I/O cost further we create multi-level indices

one block



I'st level Index — 2nd level Index — 3rd level Index

→ # of blocks reduces

last level is one block
k^{th} level Index

DB file	---	1B acc.	1B access	one block access
1 Block access	---	K-2	K-1	k^{th} level

$1 + \dots + 1 + 1 + 1 = (1+k)$ Block access in Multilevel Indexing

* I/O Cost in Multilevel Indexing = (K+1)

Q A File consists of 100000 records with each record of size 100B and block size of 1024 B, size of search key is 10B, and pointer size is 5B. (204)

(i) I/O Cost to access without index

a) by using ordered field of the file?

b) by using unordered field of the file?

(ii) If dense index is built over search key

a) How many index blocks are required in Ist level Index?

b) What is the I/O cost to access record using Ist level Index?

(iii) If sparse index is built over search key

a) How many index blocks are required in Ist level Index?

b) What is the I/O cost to access records using Ist level Index?

Given Data

\# of records in file $= 10^5$

Size of each record $= 100B$

total file size $= 100B \times 10^5 = 10^7 B$

size of one block $= 1024B = 2^{10}B$

size of search key $= 10B$

size of pointer $= 5B$

\# of blocks needed for file $n = \frac{\text{File size}}{\text{Block size}} = \frac{10^7 B}{2^{10} B} = 9766$ Blocks

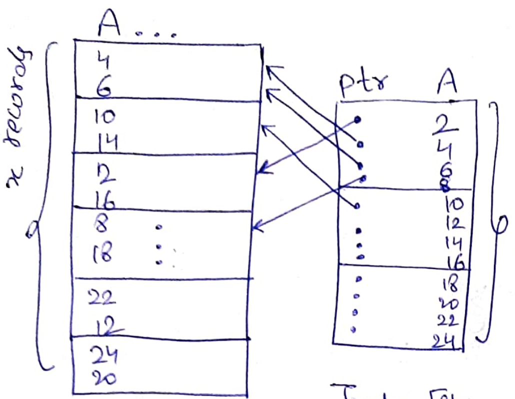
~~(i)(a) $\lceil \log_2 n \rceil = \lceil \log_2 9766 \rceil = 14$~~

(i)(a) $\lceil \log_2 n \rceil =$

(i)(b) 9766 blocks

Concepts

★ **Dense Index** (1) For each record of database file, there exists entry in index file



Index File

DB File

BF = 2 records / block

each block of index file can accomodate 4 entries => Block Factor = 4

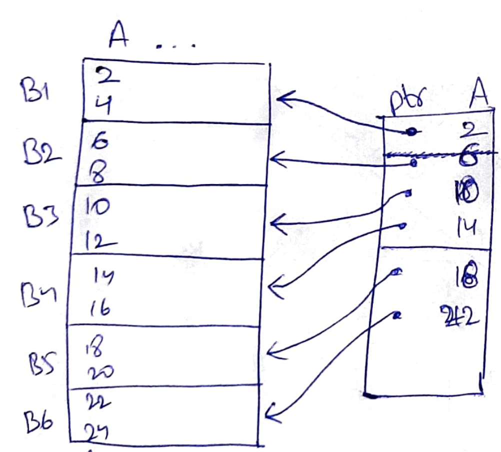
(2) Number of index file entries = Number of records of DB File

(3) Dense Index can be build for both ordered and unordered fields.

★ **Sparse Index** (1) Index entry for some set of records (not for all records of DB File, only a subset of records from DB File)

(2)

* Sparse index is built only with ordered fields



Ordered field

DB File

Index File

* one pointer for all records of one block

(3) * Sparse Index File size is less than Dense Index File size

(4) * if ordered field is CK of DB file then # of entries in sparse index file = # of blocks in DB file

* Block Factor of DB File : maximum possible records per block is the block factor.

$$B_f \text{ of DB File} = \left\lfloor \frac{B-H}{R} \right\rfloor \text{ Records per block}$$

→ B: Block size in bytes
→ H: Block Header size in Bytes (if not given take H=0)
→ R: Size of Record

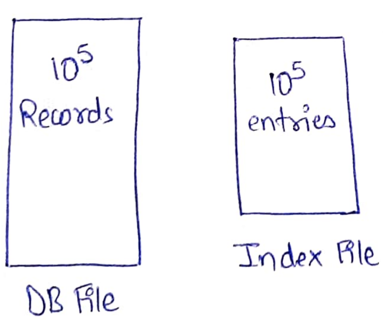
* Block Factor of index file : maximum possible index entries (Key-pointer pair) per block is the block factor.

$$B_f \text{ of Index File} = \left\lfloor \frac{B-H}{K+P} \right\rfloor \text{ entries / block}$$

→ B: Block size in bytes
→ H: Block Header Size
→ K: Size of Search Key
→ p: Size of Pointer

Solution

(ii)(a) Dense Index : For each record of DB File there exists entry in index file



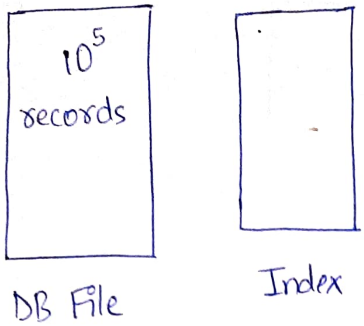
$$B_f \text{ of index} = \left\lfloor \frac{B-H}{K+P} \right\rfloor$$

$$= \left\lfloor \frac{2^{10}-0}{15} \right\rfloor = 68 \text{ entries/block}$$

$$\text{Dense Index Blocks} = \left\lceil \frac{\text{total entries}}{\text{entries per block}} \right\rceil = \left\lceil \frac{10^5}{68} \right\rceil = 1471 \text{ blocks}$$

(207)

(iii) (a) Sparse Index



# of entries = # of blocks of DB File

$$B_f \text{ of DB} = \left\lfloor \frac{B-H}{R} \right\rfloor = \left\lfloor \frac{2^{10}-0}{100} \right\rfloor = 10 \ \frac{\text{Records}}{\text{block}}$$

$$\text{DB File Blocks} = \frac{10^5 \text{ records}}{10 \text{ records}} \times \text{Blocks} = 10^4 \text{ Blocks of DB File}$$

B_f of Index File = 68 $\frac{\text{ent}}{\text{block}}$ (previously calculated)

$$\text{Blocks in Index File} = \left\lceil \frac{10^4 \text{ entries}}{68 \text{ entries}} \right\rceil \times \text{Blocks} = 148 \text{ Blocks of Sparse Index}$$

(i) (a) $\lceil \log_2 n \rceil$ $n = 10000$ (# of blocks in DB File, $B_f = 10$ of DB File)

$= 14$

(b) n blocks $= 10^4$ blocks

(ii) (b) Step 1 : Find search key in index file (Binary search)

$\lceil \log_2 m \rceil$

Step 2 : Access record of DB File

Total : $\lceil \log_2 m \rceil + 1 = \lceil \log_2 1471 \rceil + 1 = 12$

m = # of blocks in dense index = 1471

(iii) (b) Step 1 : Find Search key in Sparse index file

$$\lceil \log_2 m \rceil$$

Step 2 : Access DB File Record

$$\lceil \log_2 m \rceil + 1 = \lceil \log_2 148 \rceil + 1 = 9$$

m = # of blocks in sparse index file = 148

Lec-16 04/01/2021 **Types of Indexes**

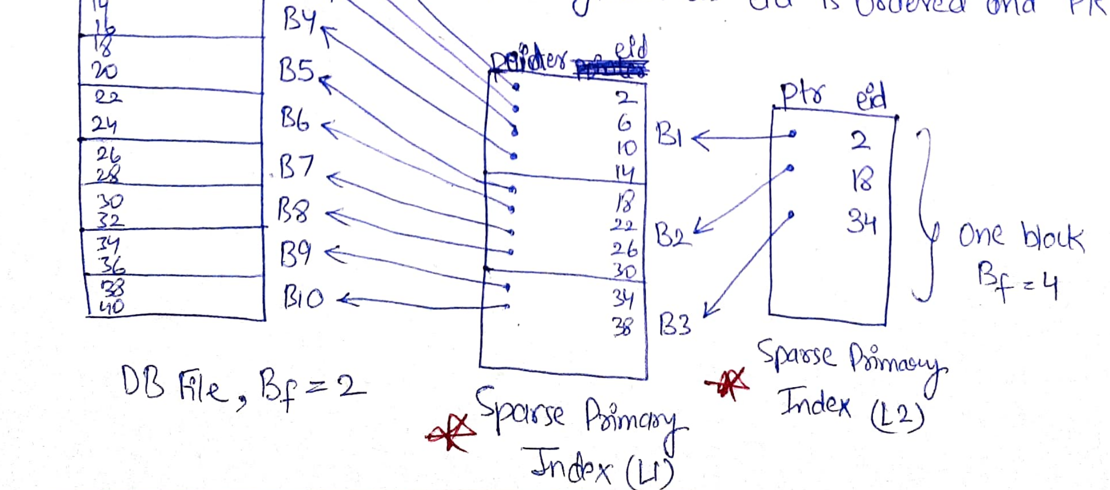
Types of Indexes — can be built over:

- Ordered Field
  - Key (PK/AK) ⇓ ① Primary Index
  - Non-Key ⇓ ② Clustering Index
- Unordered Field
  - Key (PK/AK)
  - Non-Key

  ⇓ Secondary Index: ③ Over key ④ Over non-key

① Primary Index (Index over ordered field which is a key)

* if we build an index file for this DB file with ordered field eid as search key then it will be called as primary index as eid is ordered and PK



(209)

* At most one primary index is possible for any relation
* Primary Index can be sparse or dense (sparse is preferred)

example

SELECT *
FROM EMP
WHERE eid = 24;
} executed ⟹

Step 1 : L2 Primary Index Access
2 ⟶ 18 (1)

Step 2 : L1 Primary Index Access
18 ⟶ 22 (1)

Step 3 : DB File Block Access
22 ⟶ 23 ⟶ 24 (1)

Step 4 : Final Record Access

Total Access Cost
= (K+1) Blocks
= (2+1) blocks
= 3 blocks

k → levels of indexing

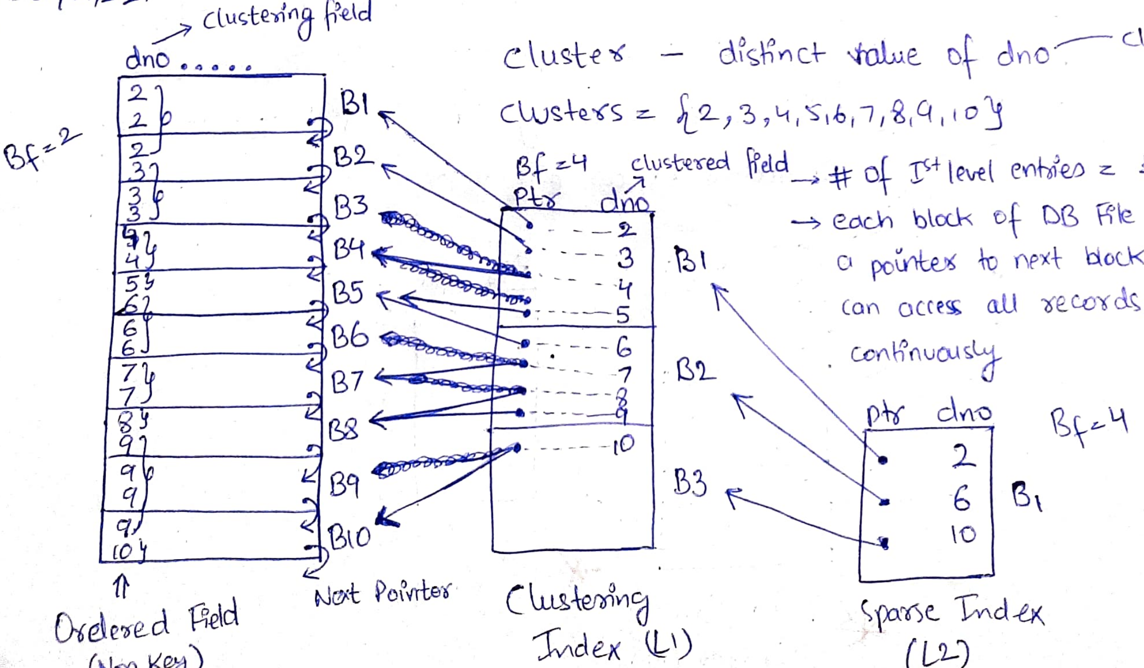
* I/O Cost to access record using primary index with multi-level indexing : (k+1) blocks , k → # of levels of multi-level indexing

* In Sparse Primary Index *anchor records exists*
⇓
first record of every block

*

(2) Clustering Index (Index over ordered field which is non-key)

07/01/2021



cluster – distinct value of dno – clustering field or attribute

clusters = {2, 3, 4, 5, 6, 7, 8, 9, 10}

→ # of 1st level entries = # of clusters

→ each block of DB File should have a pointer to next block so that we can access all records of a cluster continuously

* Clustering Index is mostly sparse index
* At most one Clustering Index is possible for any DB File
* Either Primary index or clustering index is possible for any DB File but not both

Example

SELECT *
FROM EMP
WHERE dno=2; $\Rightarrow$

Step 1 : L2 sparse index access
$2 \rightarrow 2$ (1)

Step 2 : L1 Clustering Index
$2 \rightarrow 2$ (1)

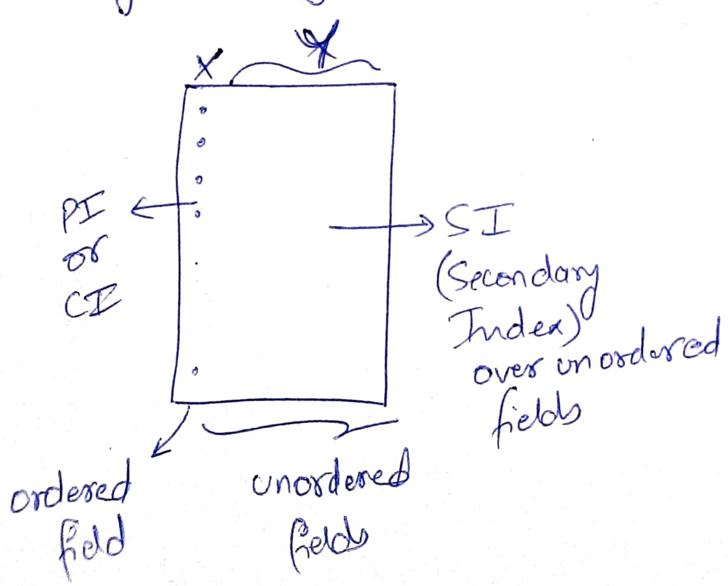
Step 3 : DB File record access until all records of 2 are read
=> this could be one or more records being accessed

* <u>I/O cost</u> to access one cluster using clustering index with multi-level indexing => * K + one or more blocks of DB access until beginning of next cluster

30:02 09/01/2021

(3) Secondary Index (index over unordered field which may be either key or a non-key field)

* it is a secondary way to access data even if primary or clustering index already exists
* many secondary indexes are possible for one DB File [DB Table]

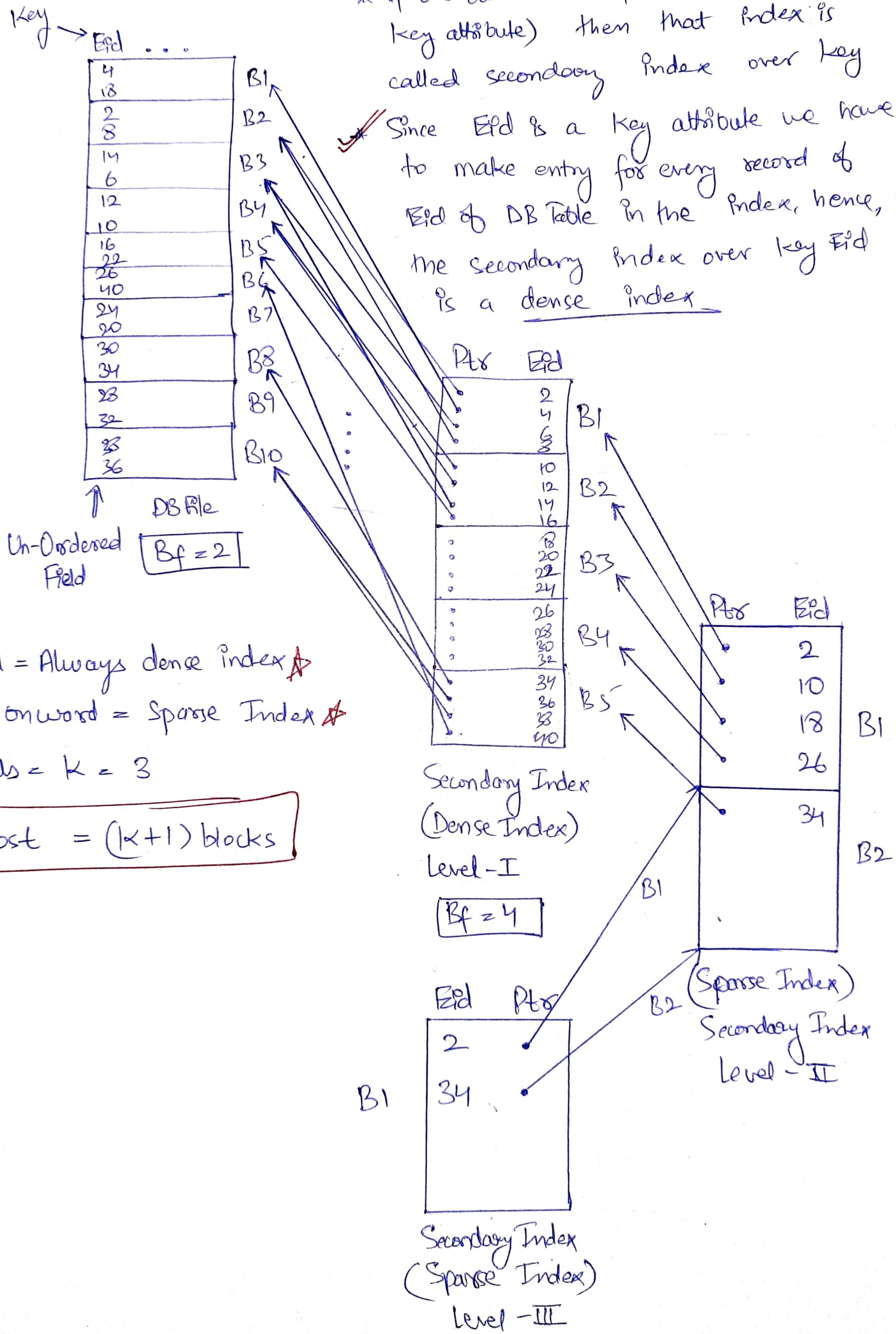


(211) 09/01/2021

Secondary Index (over key)

* If we build index for Eid (unordered key attribute) then that index is called secondary index over key

✓ Since Eid is a key attribute we have to make entry for every record of Eid of DB Table in the index, hence, the secondary index over key Eid is a dense index



Ist level = Always dense index ☆

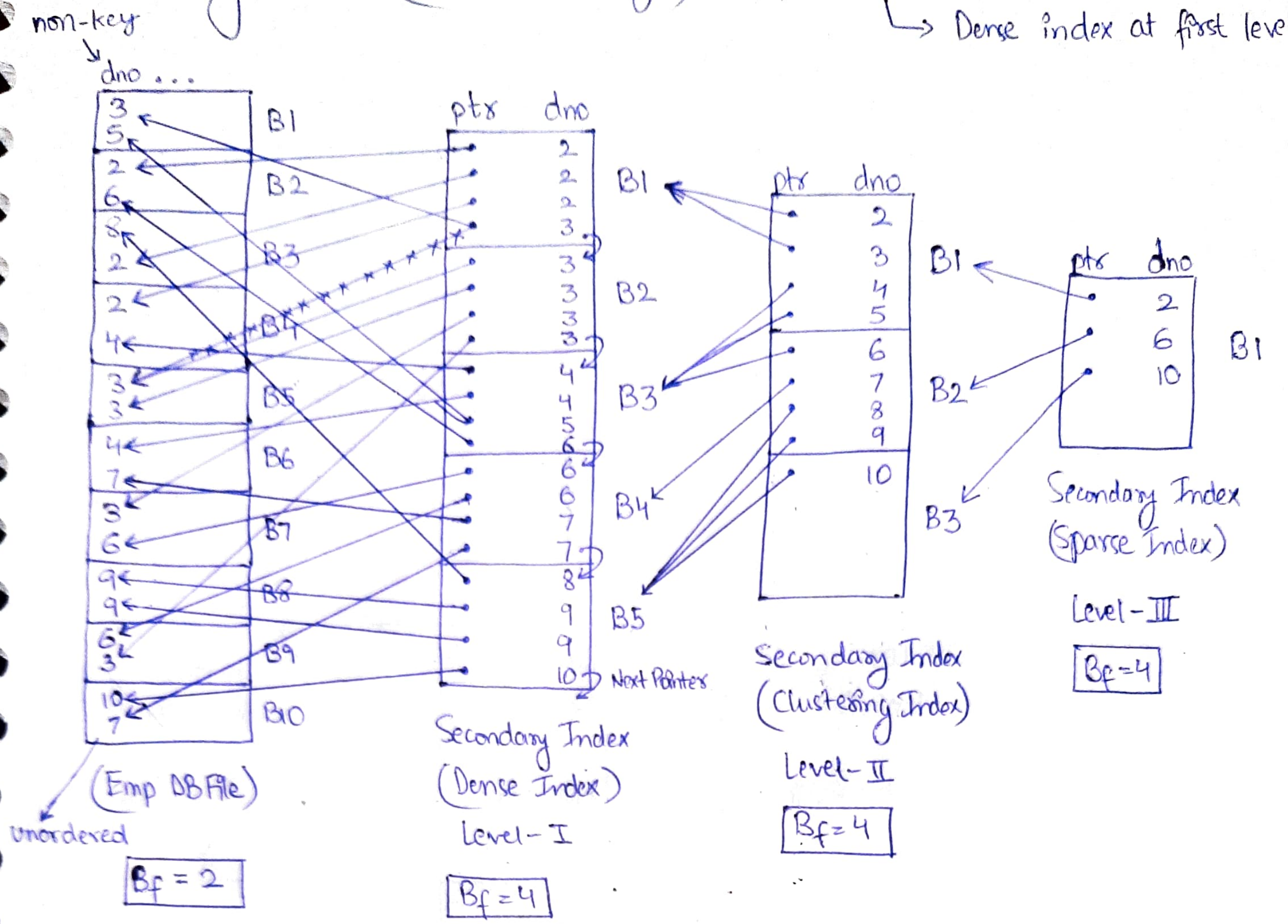
IInd level onword = Sparse Index ☆

# of levels = k = 3

$\boxed{\text{I/O Access Cost} = (k+1) \text{ blocks}}$

Secondary Index (Over non-key) : Option - 1

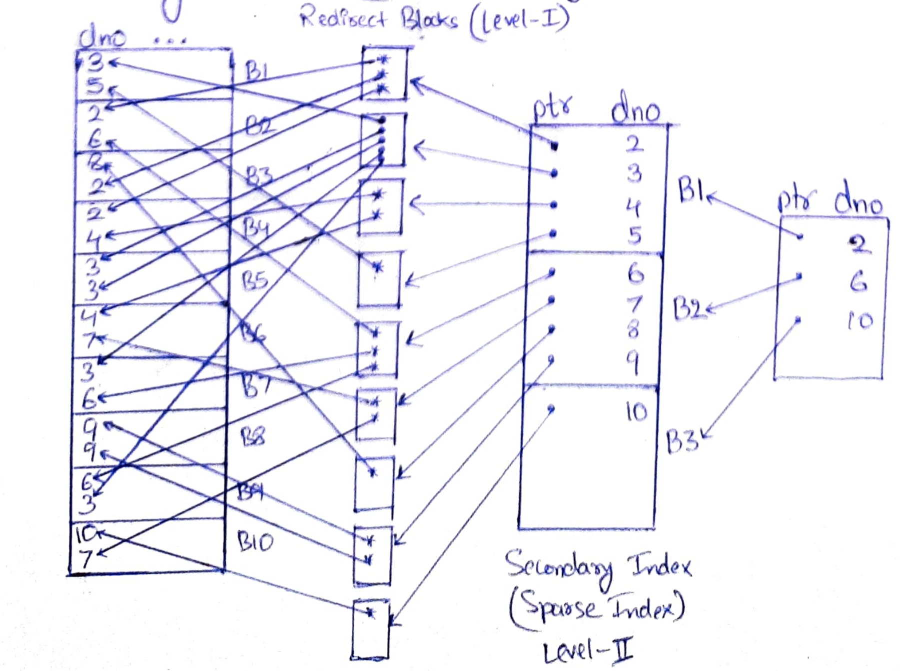
↳ Dense index at first level



Secondary Index over non-key (option 1)

* 1^{st} level index is dense index
* 2^{nd} level index is clustering index to 1^{st} level index
* I/O Cost : (k+1) blocks

Secondary Index (over non-key) : option - 2



* I/O Cost :

K + # of DB block access

⇓

# of records of same non-key

Conclusion

	Ordered Field	Unordered Field
Key Field	Primary Index	Secondary Index (over key)
Non-Key Field	Clustering Index	Secondary Index (over non-key)

Type of Index	# of I^{st} Level Index Entries	Dense or Non-Dense	Block Anchoring on the Data File
Primary Index	Numbers of blocks in the DB File	Non-Dense (Sparse)	Yes
Clustering Index	Number of distinct index field values	Non-Dense	Yes/No [*Note:A]
Secondary Index (over key)	Number of records in data file	Dense	No
Secondary Index (over non-key)	Number of records (*Note:B) OR Number of distinct index field values (*Note:C)	Dense or Non-Dense	No

*Note:A Yes, if every distinct value of the ordered field starts a new block; No, otherwise.

*Note:B Option 1

*Note:C Option 2

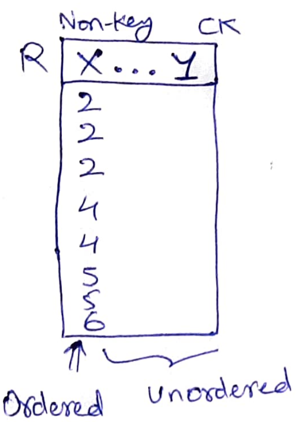
Lec-16 Continued 11/01/2021

Q If DB File is ordered over non-key attribute X and index is built over CK Y then index is ________

a) Primary Key b) Clustering Index
c) Secondary Index over key d) Secondary Index over Non-key

my ans: option C

Sir's soln



Index is built for attribut Y
(i) Y is CK (Key)
(ii) Y is unordered

Index over unordered key attribute Y => Sec Index over key

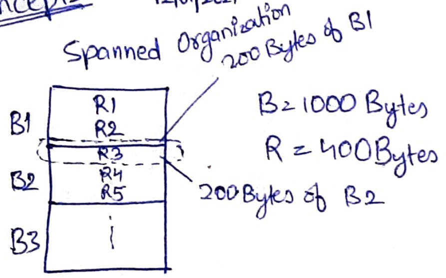
Q Consider a DB File which consists of 80000 records with record size of 100B and block size of 1024 B. Search key size is 15 B and it is the CK of the file. Unspanned organization used to store records in DB File. If sparse primary index build over DB File with pointer size 12B and multilevel indexing is used.

(i) How many levels of index are required? 3
(ii) How many index blocks are required in all levels?
1) 217 2) 6 3) 1 Total: 224

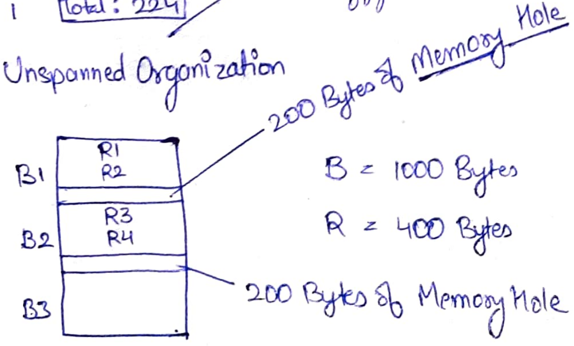
Most of the cases in the actual exam, it is unspanned organisation

Concepts 12/01/2021

Spanned Organization

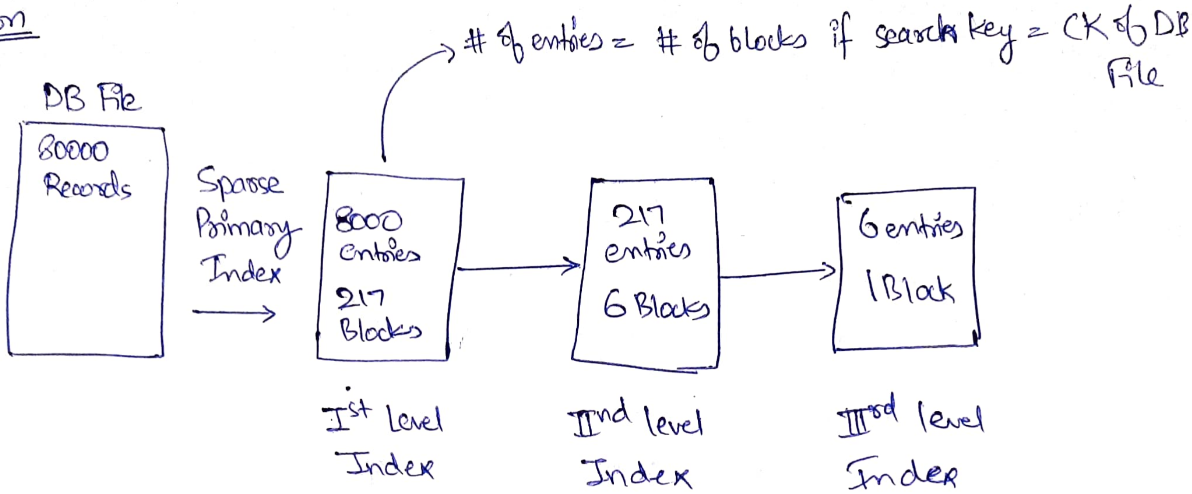


Unspanned Organization



(215)

Solution



# of Blocks of DB File → Calculate Bf of DB File

$$B_f(DB) = \left\lfloor \frac{B-H}{R} \right\rfloor = \left\lfloor \frac{(1024-0)B}{100B} \right\rfloor = 10$$

$\boxed{B_f(DB) = 10}$ records / block

# of Blocks of DB File $= \frac{80000}{10} = 8000$ Blocks in DB File

# of entries in sparse PI = 8000 entries

# of Blocks required for 8000 entries in Index → Calculate Bf of Sparse PI

$$B_f(\text{Index}) = \left\lfloor \frac{B-H}{P+K} \right\rfloor = \left\lfloor \frac{1024-0}{15+12} \right\rfloor = 37$$

$\boxed{B_f(\text{Index}) = 37}$ entries / block

# of Blocks in PI $= \left\lceil \frac{8000}{37} \right\rceil = 217$ Blocks of Ist level Index

Since 217 > 37 entries / block

we need 2nd level index (Sparse index)

\# of entries = # of blocks of 1st level index

\# of entries in 2nd level = 217 entries

\# of Blocks in 2nd level = $\left\lceil \frac{217}{37} \right\rceil$ = 6 Blocks

Since 6 < 37 entries / block ~~=> No further indexing require~~

=> we need 3rd level index (sparse index)

=> we need 1 Block in 3rd level

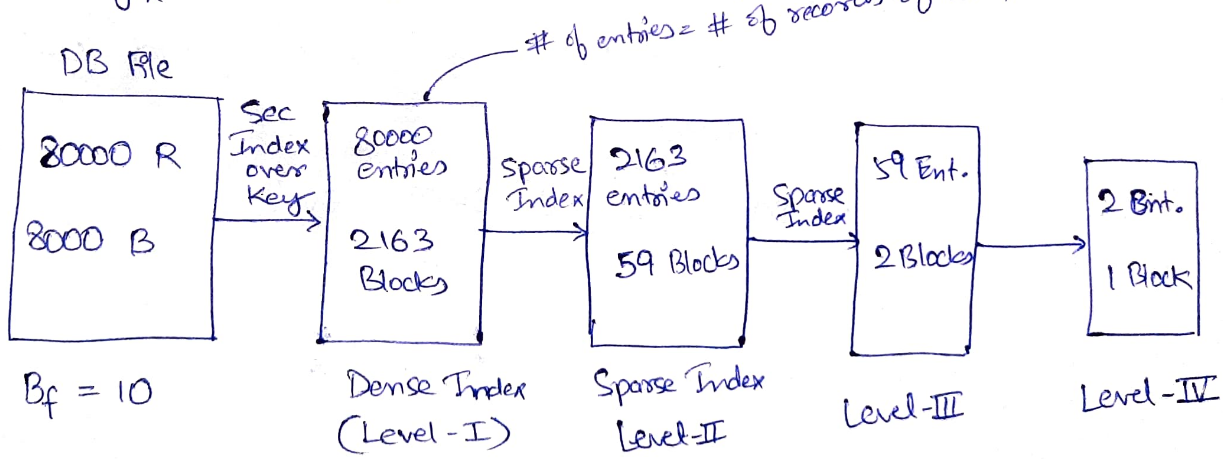
12/01/2021

Q Consider the DB File consists of 80000 records with record size of 100B and block size of 1024 B. Search key size is 15 B which is also the CK of the DB File. Unspanned organization is used to store records in DB File. If secondary index is built over DB file with ptr size of 12B and multilevel indexing is used then:

(i) How many levels of index is required? 4 Levels

(ii) How many index blocks are required in all levels? 2225 Blocks

Solution



B_f(Index) = 37 # of Blocks in level-I = $\left\lceil \frac{80000}{37} \right\rceil$ = 2163 Blocks

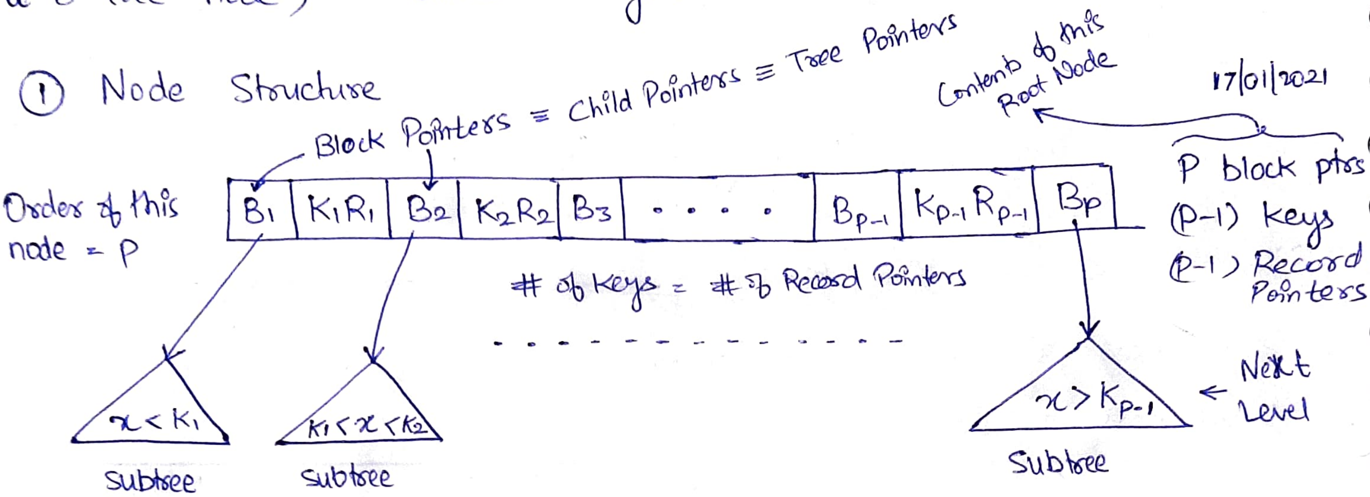
217 Lec-17 12/01/2021

Dynamic Multi-level Indexing
- B Tree
- B+ Tree

B-Tree Definition

(Order P : max possible block pointers which can be stored in a B Tree node) Order ≡ Degree ≡ Fan-out

① Node Structure

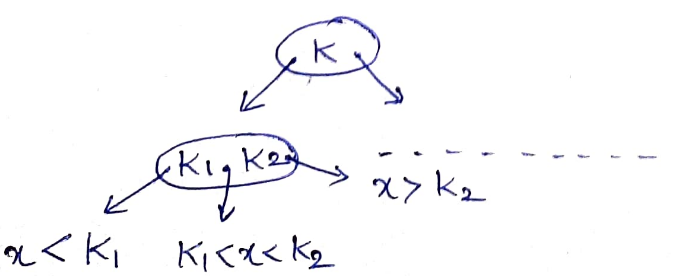


② Every internal node except the root node must have at least $\lceil P/2 \rceil$ block pointers and $\lceil P/2 \rceil - 1$ keys, at most p block pointers and (p-1) keys

③ Root can be at least 2 block pointers and one key
at most p block pointers and (p-1) keys

④ Every leaf node must be at same level

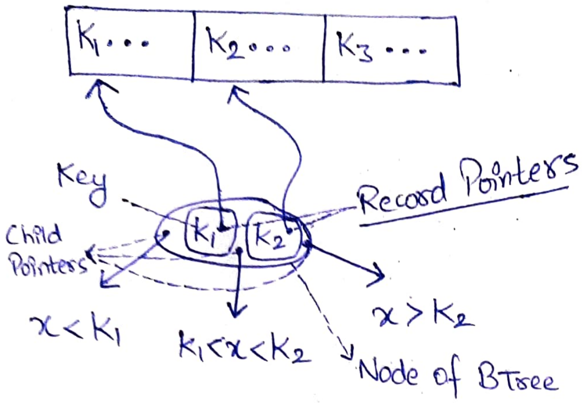
In General, if we have one key in one node ⇒ we need two pointers, one for left subtree and one for right subtree



two keys in one node ⇒ 3 pointers

p keys in one node ⇒ (p+1) pointers

* B Tree is balanced tree
* B Tree and B+ Tree are used as index to some DB File

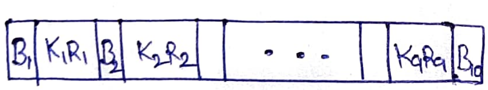


* every key of a node of B Tree has a Record Pointer which points to the DB File

* Child Pointer : pointer which is pointed to next level tree node
* Record Pointer : pointer which points to data in DB File
* Node consists of Child Pointers, Record Pointers, Keys

if # of keys $= (P) \Rightarrow$ # of Record Pointers $= (P)$
$\Rightarrow$ # of Child pointers $= (P+1)$

example   order $= 10$ (max 10 block pointers) $\Rightarrow$ 9 Keys



$K_{10}R_{10}$
$\Uparrow$
if we want to include tenth key then we have to do Node Split

~~more~~ less than   middle key

$$\underbrace{K_1 < K_2 < K_3 \ldots}_{\text{Left Node}} \overset{\Downarrow}{K_5} \underset{\Downarrow}{} < \underbrace{K_6 \ldots < K_{10}}_{\text{Right Node}}$$

$K_5 \Downarrow$ parent

Node Split