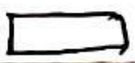
classmate
Date
Page 1

DBMS - Short NOte

- ER-MODEL:

→ Entity → 

→ Weak entity → 

→ relationship → 

→ Identifying relation → 

→ attribute → 

→ Key attribute → 

→ Multivalued Attribute → 

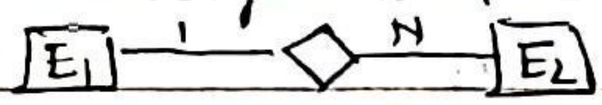
→ Composite Attribute → 

→ Derived Attribute → 

→ Total participation of E2 in R

→ 

→ Cardinality ratio $E_1 : E_2 = 1 : N$

→ 

- Relational Database Model:

→ $R(A_1, A_2, \cdots A_n)$ = relation schema.

- Relational constraints —
  - → Domain (entire schema of attribute should atomic)
  - → Key (NO two tuples can't be same value)
  - → Entity Integrity (primary key never allowed NULL value)
  - → Referential Integrity.

- operation on Database —

(I) Insert (II) Delete (III) Update.

↓ ↓

all constraints violate · only Referential Integrity Violated.

- SuperKey → Superset of a Key is a superkey.

→ $R(A_1, A_2, A_2, \cdots A_n)$, ~~$CK = A_1A_2$~~

$CK = A_1A_2$ then $SK = 2^{n-2}$

$CK = A_1$ then $SK = 2^{n-1}$

$CK = A_1A_2A_3$ then $SK = 2^{n-3}$

$CK = (A_1, A_2)$ etc, $SK = SK(A_1) + SK(A_2) - SK(A_1A_2)$

$= 2^{n-1} + 2^{n-1} - 2^{n-2}$

$CK = (A_1, A_2, A_3)$, $SK = SK(A_1) + SK(A_2) + SK(A_3) - SK(A_1A_2) - S(A_2A_3) - S(A_3A_1) + SK(A_1A_2A_3)$

- Conversion of ER model to relational model:

ER Model	Relational model
1:1 or 1:N	1FK
M:N	relation + 2FK.
n-ary	relation + n FKs.
Multivalued attribute	Relation + FK.
Key attribute	Primary key.

*** Normalization: (Functional Dependency & Canditate key)

→ Divide table into small (small table. (to avoid redundency)

Candidate Key → $R(A_1, A_2, A_3 \cdots A_n)$

** Candidate key may possible = $\boxed{(2^n - 1)}$

→ $R[A_1, A_2, A_3 \cdots A_n)$

$X \rightarrow Y$

$2^n \quad 2^n = \boxed{2^{2n}}$ (total FDs possible with n-attributes)

** → Equivalence of FDs -

$F: \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$

$G: \{A \rightarrow CD, E \rightarrow AH\}$

$\boxed{F \geq G}$ ✓
$\boxed{G \geq F}$ ✓ $\longrightarrow \boxed{F \simeq G}$

** → FD preserving

$R(A_1 A_2 A_3 \cdots A_n)$

$F \rightarrow F^+$

R_1 — $F_1 \subseteq F^+$; R_2 — $F_2 \subseteq F^+$

$(F_1 \cup F_2)^+ = F^+ \rightarrow$ FD preserving.

** → for lossless decomposition common attribute of both the table should be a key (for atleast one table).

*** → Normalization:

$\boxed{1NF \rightarrow 2NF \rightarrow 3FF \rightarrow BCNF}$

↳ with BCNF 0 % redundency.

- 1NF → (No multivalued and no composite attribute Allowed)
- 2NF → No partial dependency allowed. [$AB \rightarrow C$, ~~$B \rightarrow C$~~ (Pd)]

$R(ABCDE)$, $CK = AC$, $FD \rightarrow \{\underset{Pd}{A \rightarrow B}, B \rightarrow E, \underset{Pd}{C \rightarrow D}\}$

- 3NF → NO transitive dependencies. [$A \rightarrow B, \underset{Td}{B \rightarrow C}, \Rightarrow A \rightarrow C$]

Candidate key

$X \supset Z$
$Z \rightarrow Y$ → PD

↓
Subset of non-prime attribute

$X \rightarrow Y$ → TD.

↓ Non-prime attribute ; ↓ Non prime (Key) attribute

• BCNF → Left side determinants of all FDs must be Superkey.

R(ABC), FD ((A)^{SK} → B, (B)^{SK} → C, (C)^{SK} → A)

CK = (A, B, C) $A^+ \to ABC$, $B^+ \to ABC$, $C^+ \to CAB$

↳ I, II, III, BCNF.

** • Relational Algebra:

π → projection (clm), σ → selection (row), ρ → rename.

↳ remove duplicate ↳ commutative

↳ not commutative

$\cup$ → eliminate duplicate → R ∪ S = S ∪ R, R − S ≠ S − R

• cartesian product → $(R \times S)$ (r, s)

→ Total attribute at new table (r + s) → no. of row at S

→ no. of tuples will present in new table → $n_r \times n_s$

• Join and Natural join:

(⋈) (*)

→ Join: ⋈ = (×, σ)

↳ $R \bowtie_{\text{<join condition>}} S$

↳ min tuple can present = 0 & max = $n_r \times n_s$

→ Natural join: *

↳ R * S → $R \bowtie_{<A=A \text{ AND } B=B>} S$, "min tuple = 0, max = mn"

• Aggregate function (SUM, AVG, MAX, MIN, COUNT).

R(A,C)

A	C
1	–
2	–

m = 2

S(B,C)

B	D
1	–
3	–

n = 2 (row)

(1) R × S =

A	C	B	D
1	–	1	–
1	–	3	–
2	–	1	–
2	–	3	–

mn = 4

(2) Join: $R \bowtie_{(A=B)} S$

A	C	B	D
1	–	1	–

(3) Left outer join: $R \bowtie_{A=B} S$

A	C	B	D
1	–	1	–
2	–	N	N

(4) Right outer join: $R \bowtie_{A=B} S$

A	C	B	D
1	–	1	–
N	N	3	–

(5) full outer join: $R \bowtie_{A=B} S$

A	C	B	D
1	–	1	–
2	–	N	N
N	N	3	–

**

→ min Max tuples —

$R \rightarrow n$ (no. of tuples in R)

$S \rightarrow m$

	min	max
X	$m+n$	$m+n$
⋈	0	$m \times n$
⟕	n	$m \times n$
⟖	m	$m \times n$
⟗	$\max(m,n)$	$m \times n$

→ Basic or complete set ($\sigma, \pi, \cup, \rho, -, \times$)

→ not Basic set ($\bowtie, \cap, \frac{\rho}{i}, \bowtie, ⟕, ⟖$)

→ min max tuples —

	Min	Max
$\cup$	$\max(m,n)$	$m+n$
$\cap$	0	$\min(m,n)$
$-$	0	m

→ F-Language

↓ RA

↓ Relational calculus (RC)

↓ TRC (tuple relational cal) ↓ DRC (Domain relational

→ RA ≅ TRC ≅ DRC (same power)

↓

{t.FN | student(t) AND t.Mark > 50}

↑ select

P Q R

FN	LN	Mark

**• SQL (TRC):

SELECT <attribute>
FORM <table list>
WHERE <condition>

→ DISTINT used to eliminate duplicates.

→ LIKE '%' → no. of char 0 or more, '_' → single character.

→ ORDER BY — DESC, ASC

↳ By default sorting is done in the ascending order.

→ NULL (a = NULL) X, (P.no is NULL) (P.no Is not NULL)

→ IN, ANY, ALL, SOME.

** → Exist and NOT EXIST.

↳ opposite of exists.

→ In Natural join two prime attribute name should same.

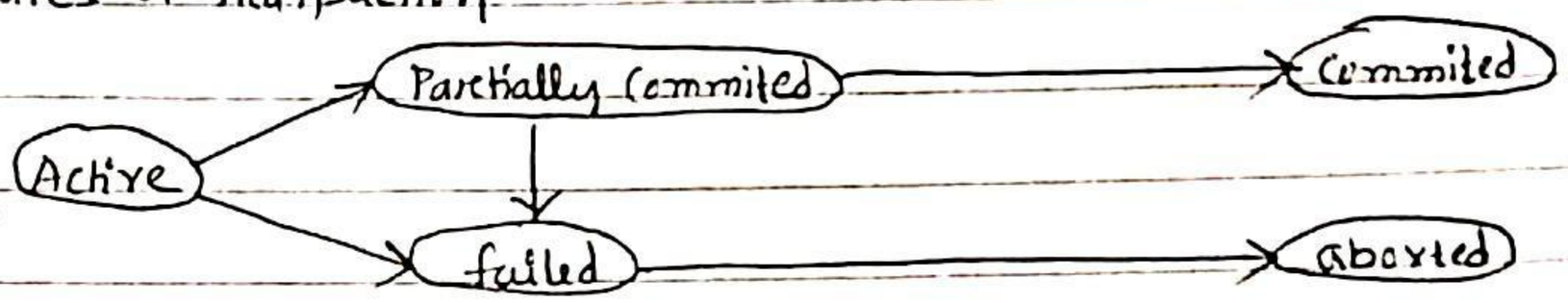
→ COUNT didn't count NULL.

→ GROUP BY. → GROUP BY A, GROUP BY AB

**

- Transaction Management and Concurrency Control:

→ States of Transaction -



→ problems due to Concurrent execution of transaction -

(I) W-W conflict (II) W-R conflict (III) R-W conflict.

- Characterizing schedules -

① Serial schedule → if 'n'-transaction in a schedule the possible no. of serial schedules are $\boxed{n!}$ $T_1 \rightarrow T_2$ or $T_2 \rightarrow T_1$

② Complete schedule → if at least operation of each transaction is either abort or commit.

* ③ Recoverable schedule → T_j reads a data item that was previously written by T_i then commit operation of T_i should appare before commit operation of T_j.

* ④ Carecadeless schedule → T_j reads data that written by T_i the commit operation of T_i should appare before the read operation of T_j.

* ⑤ Strict schedule → If a value written by a transaction can't be read or over write by other transaction unhill the transaction is either aborts or commit.

⑥ Equivalent schedule →

→ result equivalent schedules → S_1 & S_2 said result eq if both produce same final result for some initial value.

→ Conflict equ schedule → R(A) ... W(A), W(A) ... W(A), W(A) ... R(A) } Conflict operations.

→ Order of Conflict should be same then, $S_1 \stackrel{c}{=} S_2$ → Conflict equivalent

** • Conflict Serializable -

→ $T_1 \rightarrow T_2$, T_3 ↗ $= C.S$

$T_1 \rightarrow T_2$, $T_2 \rightarrow T_3$, $T_3 \rightarrow T_1$ $\neq C.S.$ → cycle present

* • View Serializable → If it is view equivalent to serial schedule.

•

T_1	T_2
R(A)	(W(A)) → Blind write
W(A)	

↳ it is not C.S and not V.S.

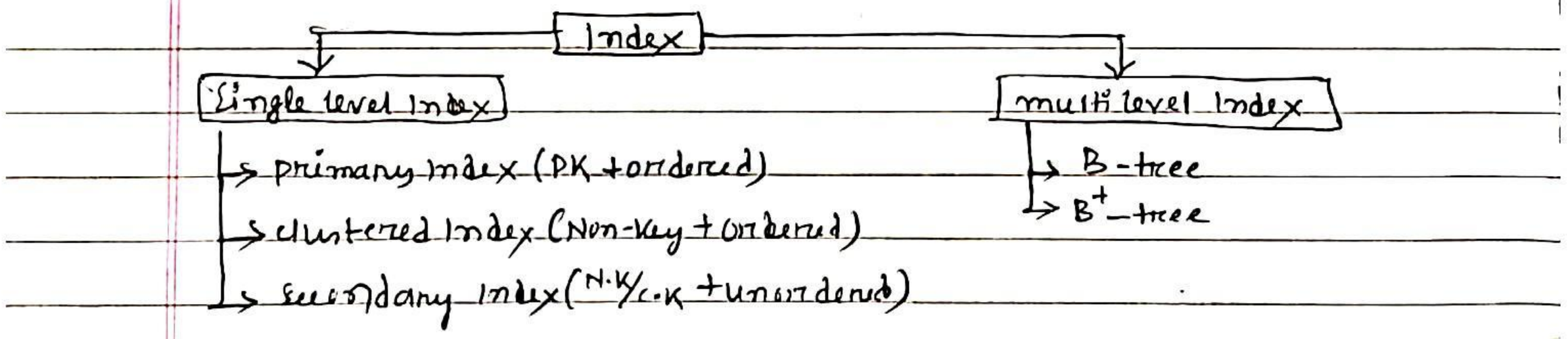
→ A schedule that is C.S also V.S

→ A schedule that is V.S but not be C.S.

- serializable schedule → either C.S or V.S or both.
- Concurrency Control:
  - Lock based protocol (Shared, Exclusive Lock)
    - 2 Phase Locking protocol.
      - ensure C.S and Serializability.
      - Cascading rollback may occure.
      - Deadlock may occure. → ~~to avoid~~
      - can avoid - (1) ~~Strict two phase locking~~.
        1. Strict two phase locking
        2. Rigorous 2-Phase locking → avoid (Cascading rollback, not deadlock)
        3. Conservative 2-PL → (Avoid cascading and deadlock both)

- FILE STRUCTURE:

- Index → used to improve searching efficiency.

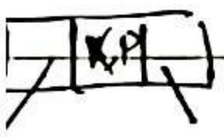


- B-tree →

$\Rightarrow n * P + (n-1)(K + P_r) \leq \text{BlockSize}$

→ Order of B-tree = 'P'

P → Block pointer.
P_r → Record pointer.
K → Search key.
n → Order of B-tree.



- Root:

min - 2 tree pointers, 1 key entry.

max - P tree pointers, P-1 key.

- Internal nodes:

min - $\lceil P/2 \rceil$, $\lceil P/2 \rceil - 1$

max - P , P-1

- Leaf:

min - $\lceil P/2 \rceil - 1$ key entries.

max - (P-1) key entries.