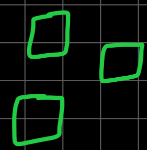
Lec 1

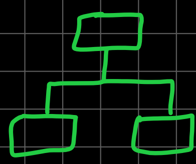
- Data: Anything that can be recorded
- SQL is used to fetch data
- Conceptual layer, Physical layer
- ERD (Entity-Relationship diagram)
- Normalization (ERD → Conceptual layer)
- RA (Relational Algebra) Pseudocode of DB
- Transactions (ensuring same state)
- History
- Relational DBMS
-



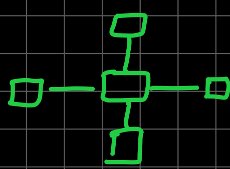
DB [Data + Database management sys

=> Database Systems

- .NET



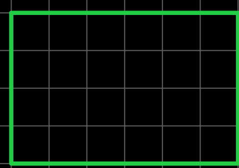
User
conceptual
Physical
→ Linear
storage



- HTML, CSS
- SQL practice
- w3schools

Lec 2

- File System
→ Redundancy
→ Search
→ Updation
→ User view
→ Big Data



Sol: RDBMS

- key = Super key

- Attributes: Column
- Record: Row
- Unique key to identify similar data : key
- key: 1 / more attributes
- Super key
- max num of super keys: 2^n-1 ; n = N.o of attributes ; attributes are unique
→ can be less than $< 2^n - 1$, if less, then data not unique

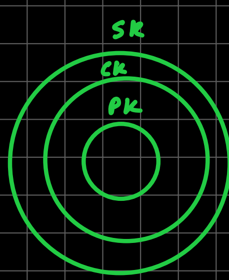
Super key

* A (B C D)

check combinations whether it is a super key

- candidate key: SK whose proper subset is not a sk.
- candidate key who is not null can be a primary key
- Only 1 Primary key whether primary or composite
- F.K: P.K of another table's P.K / self-table P.K
  F.K: can have a null
  → F.K can have repitition

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



A, AB, AC, CB, ABC
B, C

- Referential Integrity:
- Insert
- Delete
- Search
- Update
- Delete cascade / set Null → if composite primary key, then errors
- Update cascade

* minimal & minimum S.K

- R.I errors → issue with F.k
  F.k function is to maintain R.I

Lec 3

- Schema: Structure of a table

→ Drop keyword for table structure

Always append ; at end of query
- no spaces/ specific signs
  Use
- Create Database myDB;
- Drop Database myDB;
- Indentation
- Create Table (
    id int NOT Primarykey;
    name varchar (100);
    Primarykey (id);
)

col. constraint

table - level constraint
↳ F.k preferred

SQL
- DQL: select, fetch & show
- DDL: Create, Drop, Alter
- DML: Insert, Delete, update
- DCL: Remove, Grant

Data type:
numeric: int
big int
small //
tiny //
Range

decimal ($\overset{5}{size}$, $\overset{2}{decimal}$)
num of digits — digits after decimal

- Schema constraints
- Application constraints // Unique
- NOT NULL

*char ( ) mention size
*varchar (255)
*text ( ) max len of 2gb

- Drop Table myDB;
- *Truncate Table myDB; → schema preserved, then delete, and then create ; attributes preserved
- Delete

- Add column column
  Alter Table syntax from w3schools
- Rename → diff in MS SQL + Oracle

describe → oracle
sp- col .... → MS SQL

- sp _ keyword lookup

Lec 4

Constraints

- P.k, F.k → constraints
- NOT NULL → col. level constraint

- CREATE TABLE myTab(
    Id INT NOT NULL DEFAULT

    Id INT CONSTRAINT pk PRIMARY KEY;

    Constraint constraint_name PRIMARY KEY (),

    // // CHECK (Col. names)

    // fk FOREIGN KEY

    id INT PRIMARY KEY
  );

primary key to be made

only col
- NOT NULL

col/Table
- UNIQUE
- CHECK
- DEFAULT
- PRIMARY KEY
- FOREIGN KEY

F.k: A col. that is defined unique

table level constraints for a composite primary key

- Date function in diff SQLs
- composite fks
- Same constraint names are not allowed in a database.
- DROP Constraint
- ALTER Constraint

Lec 5

- select] used to fetch data
  from table name
  where

from ①
then select,
where

- select c.name from customer as c
- use ' ' when space is to be entered
- alias
- from customer where name between 'a' and 'c'
- order by → sorting
- distinct as a whole on 2 fields
- select distinct name * 2 Ahmed , 2 cgpa
- avg (field name)
- sum (field name)
- count (*)
→ * counts null entries as well
- min()] for char
- max()
- select top and % , always ceiling value

- between
- and
- or
- not

_ = one char
_h%   % = zero to many

in
like
dsc
asc

Lec 6

- SELECT clause

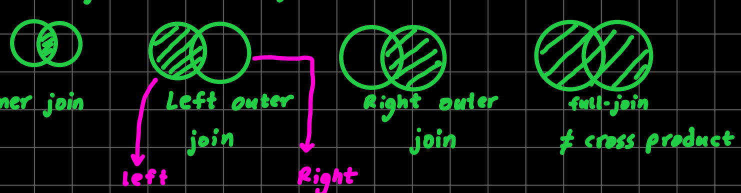
- UPDATE
  WHERE
- DELETE
  FROM
- wildcards
- SELECT __ FROM __ WHERE __ GROUP BY __ HAVING __ ORDER BY __
→ can take aggregate funcs, and columns on which grouping is done.
→ sum/avg of all catered by the above clause
→ having works on aggregate funcs.

- cross product
→ link not required b/w 2 tables
- SELECT __ FROM __ , __ ;
  e.g: select * from customer, Billing ;
  3 rows   4 rows
  3x4= 12 rows in result

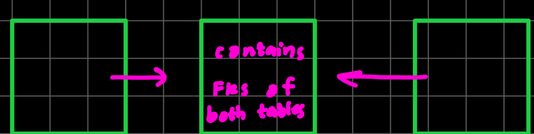
- join
- SELECT __ FROM __ , __ WHERE __ ;
→ Foreign key required

- SELECT __ FROM __ inner join on __ ;   no where
  → not necessary for column names to be same , repetition
- inner join & natural join → common column names must be same. , equal compulsory

equijoin , non-equijoin



bridge table



- bridge table to break many-to-many relation
- Foreign key not required but will depend on condition.
→ self join

- Attributes / Fields : columns → No repetition
- Records / Tuples : row

- key / super key: Single / set of attributes that uniquely defines every row

Name	Marks	Dept	course
→ a	78	CS	c_1
b	60	EE	c_1
*→ a	78	CS	c_2
b	60	EE	c_3
c	80	IT	c_2

- C.K: SK whose proper subset is not a sk.
  → Minimal SK ; $s_2 \subseteq s_1$, $s_1 \not\subseteq s_2$] conditions for proper subset

- Minimal S.K = $AB\cancel{C}$, $A\cancel{B}$, A] all are S.K
→ As minimum as possible

R (A, B, C)

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

R(A, B, C, D)

A	B	C	D
1	1	5	1
2	1	7	1
3	1	7	1
4	2	7	1
5	2	5	1
6	2	5	2

- P.K → C.K having no null values

S.K = A, AB, AC, BC, ABC
C.K = A, BC

S.K = A, AB, AC, AD, ABC, ABD, ABCD

		S.K	C.K
ϕ	A	✓	✓
{A}, {B}	AB	✓	✗
A, C	AC	✓	✗
B, C	BC	✓	✓
A, B, C, AB, AC, BC	ABC	✓	✗

Q- R(A,B,C) A is a CK
no. of SK?

→ AB, AC, ABC, A → 4

AC is a C.K, no. of SK

→ AC, ABC => 2

Q R(A,B,C,D) A, D = C.K

→ A, AB, AC, AD, ABC, ABD, ACD, ABCD
D, DB, DC, DCB

= 12

F.D → $x \to y$
if $t_1.x = t_2.x$
then $t_1.y = t_2.y$

Type of FD
- Trivial : $x \to y$; if $y \subseteq x$ => $x \to x$ Always valid

  e.g: R.No → R.No
  $\underbrace{(R.No, Name)}_{x}$ → Name

② Non-trivial nothing common

- Armstrong Axioms
- Reflexivity: $x \to x$
  $x \to y$; $y \subseteq x$
- Transitivity: if $x \to y$ and $y \to z$, then $x \to z$
- Augmentation: if $x \to y$, then $xA \to yA$
- Union: if $x \to y$ and $x \to z$, $x \to yz$
- Decomposition: if $x \to yz$, then $x \to y$, $x \to z$; can only split RHS
- Pseudo-transitivity: if ($x \to y$ and $yz \to A$) then $xz \to A$
- composition: if ($x \to y$ and $A \to B$), then $xA \to yB$

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

FD: { A→B, B→C, C→D, D→E }

A→A	B→D		
A→C	B→E	C→C	E→E
A→D	B→B	C→D	
A→E		C→E	
A→ ABCDE	B→BCDE	C→CDE	

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

- A^+ = { A,B,C,D,E } → S.K   S.K set of attributes whose closure contains all attributes of a relation
- AD^+ = { A,D,B,C,E } → S.K
- B^+ = { B,C,D,E } ×
- CD^+ = { C,D,E } ×

N.O of SK = $2^n = 2^4 = 16$

S.K = A, AB, AC, AD ....

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

FD = { A→B, D→E }

$A\cancel{B}CDE^+$ = { A,B,C,D,E } ✓

$ACD\cancel{E}^+$ = { A,C,D,E,B } ✓   A→B , so B discarded

ACD^+ = { A,C,D,E,B } ✓

Proper Subsets

A^+	AC^+
C^+	AD^+
D^+	CD^+

if none are S.K, then ACD is a C.K

- Prime attributes: A,C,D , attributes making a C.K
- check for Prime attribute on rhs

Q- R(A,B,C,D)

FD = { A→B, B→C, C→A }

A→C

① $ABCD^+$ = A,B,C,D

AD^+ = A,D,B,C ⇒ C.K

A^+ = A,B,C × S.K

D^+ = D × S.K

② Prime Attributes = A,D,C,B

→ check if present on RHS   C.K = AD, CD, BD

→ Replace

→ if C.K, update prime attributes

CD → SK ⇒ C.K

C^+ = C,A ×

D^+ = D ×

BD

B^+ = B,C ×

D^+ = D ×

Q- R(A,B,C,D)

FD = { AB→CD, D→B, C→A }

AB ⊄ Ø

→ AB

A^+ = A ×

B^+ = B ×

→ CB ✓

C^+ = C,A ×

B^+ = B ×

P.A = A,B

AB → AD

↓

CB

↓

CD

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

F = { AB→C, C→DE, E→F, D→A, C→B }

~~ABCDEF~~

P.A = A, B, C, D

C

→ AB

A^+ ×

B^+ ×

DB

D^+ = D, A ×

B^+ ×

CB ×

C^+ →

→ AC ×

A^+ = ×

C^+ = C, D, E, A, B, F

AB → AC ×

AB → DB

DB → CB ×

C.K = AB, BD, C

Q. No 1: Consider a relation R (A, B, C, D, E, F), with the set of FDs F= {AB→C, CD→E, EF→A, BC→ D, DE→F}. Find all possible keys (i.e. candidate keys) of this relation? Prove it. [5]

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

F = { AB→C, CD→E, EF→A, BC→D, DE→F }

→ Axioms

~~A~~B~~C~~DE~~F~~

P.A = BDE C A B

BDE = BDEFAC

C.K

BDE → BCD ×

BDE → BCE × (BC)

(BC) → AB

AB → EFB

EFB →

C.K = BDE, BEF, BC, AB

B^+ = B ×

D^+ = D ×

E^+ = E ×

BD^+ = BD ×

BE^+ = ×

DE^+ = DEF ×

BCE ×

B^+ ×

C^+ ×

E^+ ×

BC^+ = B, C, D, E, F, A

BE^+ = B, E ×

CE^+ = C, E, ×

→ BCD ×

BC^+ = B, C ...

BD^+ = B, D, ×

CD^+ = C, D, E, F, A

EFB

EF^+ = EFA ×

EB^+ = ×

FB^+ = F, B, ×

AB

→ A^+ ×

→ B^+ ×

Normalization

· Problems

→ Redundancy

→ Insertion Anomaly

→ Update Anomaly (can lead to inconsistency)

→ Delete Anomaly

1 NF Conditions: Each cell contains only 1 value

· Values must be atomic (can't be further decomposed)

→ 2 Phone Num, seperate records

*

· Store from same domain

· Each col. should have unique name

· No ordering to rows & col.

· No duplicate rows

· FD's already in 1NF

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

F.D = {A→B, B→C, C→D, D→E}

A→C
A→D
A→E

$A\cancel{B}\cancel{C}\cancel{D}\cancel{E}F$

AF^+ = { A,F,B,C,D,E }

P.A= A,F
N.PA= B,C,D,E

A^+= A,B,C,D,E ×
F^+= ×

A→BCDE , so PD exists
F No 2nf

2 NF conditions

- Be in 1NF
- No Partial Dependency in relation
  Proper subset of
  ck → non prime attribute

Q- R(A,B,C,D)
FD= {A→B, B→C, C→D}

$A\cancel{B}\cancel{C}\cancel{D}$

A
A^+= ABCD

P.A=A
N.PA= BCD

Proper subset is Null , so in 2nf.

Q- R(A,B,C,D)
FD= { AB→CD, C→A, D→B}

AB→C , AB→D
AB→A , AB→B

CK= AB, DB

AB
A→ ×
B→ ×

P.A= ABD
NPA= C

2NF , no NPA determined

Q- R(ABCD)
FD = { A→B, B→D}

$A\cancel{B}C\cancel{D}$

AC^+= ACBD
A^+= A,B,D ×
C^+= C ×

p.A= A,C
N.PA= BD

C.K= AC

A→B ✓ , Not it in 2nf
C→ ×

- 3NF conditions
- It is in 2NF
- Transitive Dependency must not be present for NPA
  NPA → NPA
  Both must be NPA, otherwise not in 2NF and, thus not in 3NF

For Non-trivial FDs , at least one hold
① LHS is SK
② RHS is PA

Q- R(A,B,C,D) $\not{}$ fails using ②
FD= {A→B, B→C, C→D}
SK ① NPA NPA , not in 3NF
1st NPA

C.K= A ②
P.A= A
NPA= BCD

Q- R(ABCDEF)
FD = { AB→CDEF, BD→F}
NPA NPA

$AB\cancel{C}\cancel{D}E\cancel{F}$

PA = A,B
NPA= C,D,EF

AB
A^+= A ×
B^+= B ×

Not in 3NF

Q- R(A,B,C,D,E)
FD={ A→B, B→C, C→D, D→A}

$A\cancel{B}\cancel{C}\cancel{D}E$

AE = ABCDE

P.A= A,E,D,C,B

A^+ = ABCD ×
E^+ = E ×

✓DE
D^+= DABCD ×
E^+= E ×

✓CE
C^+= CDABC ×
E^+= ×

✓AE
↓
✓DE
↓
✓CE
↓
✓BE

All are prime attributes so in 3NF , already in 2NF

- BCNF Conditions:

1. Must be in 3NF
2. For each non-trivial FD $X \to Y$, X must be a superkey

Q- R(A,B,C)

FD= { A→B, B→C, C→A }

C.K= A, B, C → BCNF

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

FD= { A→BCDE, BC→ACE, D→E }

BCNF	✓	✓	✗
3NF	✓	✓	✗
2NF	✓	✓	✓

A ~~BCDE~~

A^+= ABCDE ✓

P.A= A, B, C
NPA= D, E

BC
B^+=B ✗
C^+=C ✗

AC

A
↓
BC → AB
↓
AC

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

FD= { AB→CDE, D→A }

	AB→CDE	D→A
BCNF	✓	✗
3NF	✓	✓
2NF	✓	✓

AB~~CDE~~

AB
A^+= A
B^+= B

DB

D^+= AD ✗
B^+= B ✗

P.A= A, B, D
N.PA= C, E

AB
↓
DB

Dependency Preservation (Equivalency) see PP (For those use equivalencies on both FDs)

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

F= { A→B, B→C, C→D, D→A }

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

// Do not check combos of Attribute that can determine the whole Sub-relation

// Remove trivial and not included attributes in the relation

R_1 (A,B,C)

A→B, A→C

A^+ = ~~A~~BC~~D~~	A→BC
B^+ = ~~B~~C~~D~~A	B→CA
C^+ = ~~CD~~AB	C→AB
AB^+ = ~~AB~~C~~D~~	~~AB→C~~
BC^+ = ~~BCD~~A	~~BC→A~~

] Duplicate FD as A→C

F_1 = A→BC, B→CA, C→AB

R_2 (C, D, E)

C^+ = ~~C~~D~~AB~~	C→D
D^+ = ~~DA~~BC	D→C
E^+ = ~~E~~	
CD^+ = ~~CDAB~~	
DE^+ = ~~DE~~C~~DAB~~	~~DE→C~~
CE^+ = ~~CE~~D~~AB~~	~~C→D~~

F_2= C→D, D→C

$\underbrace{F_1 \cup F_2}_{G} \equiv F$

$D \to C \to A$

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

Take closure of FD not present for checking

$G \equiv F$ DP preserved
$G \subset F$ ✗ DP
$F \subset G$ Not Possible

Q- R(A,B,C,D)
$F = \{A \to BCDE, BC \to AED, D \to E\}$
$R_1(A,B)$, $R_2(B,C)$ $R_3(C,D,E)$

$R_1(A,B)$
$A^+ = \cancel{A}B\cancel{C}\cancel{D}\cancel{E}$ | $A \to B$
$B^+ = \cancel{B}$

$R_2(B,C)$
$B^+ = \cancel{B}$
$C^+ = \cancel{C}$

$R_3(C,D,E)$
$C^+ = \cancel{C}$
$D^+ = \cancel{D}E$ | $D \to E$
$E^+ = \cancel{E}$

$\underbrace{F_1 \cup F_2 \cup F_3}_{G} = \{A \to B, D \to E\}$

G covers F, hence $G \equiv F$
F does not cover G, hence $F \not\equiv G$, so DP not preserved

Q- R(A,B,C,D)
$F = \{A \to B, B \to C, C \to D, D \to A\}$
$B^+ = BCDA$

$R_1(A,B)$
$A^+ = \cancel{A}B\cancel{C}\cancel{D}$ | $A \to B$
$B^+ = \cancel{B}\cancel{C}DA$ | $B \to A$

$R_2(B,C)$
$B^+ = \cancel{B}C\cancel{D}\cancel{A}$ | $B \to C$
$C^+ = \cancel{C}\cancel{D}AB$ | $C \to B$

$R_3(C,D)$
$C^+ = \cancel{C}D\cancel{A}\cancel{B}$ | $C \to D$
$D^+ = \cancel{D}\cancel{A}\cancel{B}C$ | $D \to C$

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

- G covers F
- F covers G

$G \equiv F$, hence Dependency Preserving decomposition preserved

✗ ———————————— ✗

Lossless join Decomposition

- Decomposition
- ~~Conical~~
- ERD
- ~~Transactions~~
- ~~Sab~~

- Conditions for lossless
- Attributes of R1 ∪ Attributes of R2 = Attributes of R
- Attributes of R1 ∩ Attributes of R2 ≠ ∅
- common Attribute b/w tables must be a super key in at least one decomposed table

Q- R(A,B,C,D,E)
$FD = \{AC \to B, D \to E\}$

$R_1(A,B,D)$ $R_2(ABC)$ $R_3(D,E)$

R1 (A,C,D)
{ }

$R_2(A,B,C)$
$AC \to B$

$R_3(D,E)$
$D \to E$

- $R_1 \cup R_2 \cup R_3 = ABCDE$ — condition 1

$R_1 R_2$ → AC common — condition2
R_2 (AC SK)
R_1 (AC not SK) — condition 3

Q- a) R1 (C,E) R2(A,B) R3(A,C,D) FD= { A→B, BC→D, E→C, D→A }
E→C A→B D→A

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

$R_1 \cap R_2$ x

$R_1 \cap R_3$ = C

$R_2 \cap R_3$ = A

R_2 = A SK

R_{23} (A BCD) R_1 (E,C)
A→B, D→A

R_1 R_2 x

R_1 R_3 X

$R_1 \cap R_2$ x

$R_1 \cap R_3$ = E

R_{13} (A,C,D,E) , R_2 (A,B)

A

Q- F= { AC→B, D→E }

R_1 (ACD) , R_2 (ABC) R_3 (DE)
AC→B D→E

$R_1 \cap R_2$ = AC

R_{12} (ABCD) R_3 (DE)
AC→B D→E

$R_{12} \cap R_3$ = D , lossless

b) R_1 (ABD) , R_2 (ABC) , R_3 (DE)
AC→B D→E

$R_1 \cap R_2$ = AB x

$R_1 \cap R_3$ = D

R_{13} (ABDE) R_2 (ABC)
D→E AC→B

$R_{13} \cap R_2$ = AB x lossy

Minimal cover (Minimal Equivalence set)

- steps

1. Split RHS (Decomposition)
2. Remove Extraneous Attributes from LHS + remove trivial FDs
3. Remove redundant Dependencies

Q- FD= { AB→C , AB→BD, A→BC, B→D, BD→A }

Extra Trivial

= { A~~B~~→C, AB→B, A~~B~~→D, A→B, A→C, B→D, B~~D~~→A } - 1 , 2

= { A→C , A→D, A→B, A~~→~~C, B→D , B→A }

= checking for A→C

A^+ = ABDC

- AB→C , A→C
- AB→C , A→B

Q- R(ABCDE)

{ A→BC, BCD→E , BC→D, A→D }

F_c = { A→BC, BC→DE }

= { A→B , A→C, BC~~D~~→E, BC→D, A→D }

{ A→B, A→C, BC→E , BC→D, A~~→~~D }

A^+= ABCD

Q- $F = \{ \cancel{A}BC \to D,\ \cancel{A}D \to B,\ E \to C,\ A \to B,\ D \to A,\ BC \to A \}$

$= BC \to D,\ D \to B,\ E \to C,\ A \to B,\ D \to A,\ \cancel{BC \to A}$

$BC^+ = BCDA$

Q- $F = \{ A \to B,\ C \to D,\ E \to F,\ A\cancel{B} \to D,\ A\cancel{B} \to C \}$

$A \to B,\ C \to D,\ E \to F,\ \cancel{A \to D},\ A \to C$

$A \to BC,\ C \to D,\ E \to F$

Relational Algebra

① select * from employee — (employee)

② select name from employee — π_{name} (employee)

③ select name as ename from employee — $\rho_{(ename)}(\pi_{name}\text{(employee)})$

④ select name from employee where id=2 — $\pi_{name}(\sigma_{id=2}\text{(employee)})$

⑤ select c.name, o.order id from customers as c, order as o where c.customerid = o.customerid

$\Rightarrow$ $R1 \leftarrow \rho_c\text{(customer)} \times \rho_o\text{(order)}$

$R2 \leftarrow \sigma_{c.customer\ id = o.customer\ id}(R1)$

$R3 \leftarrow \pi_{c.name,\ o.order\ id}(R2)$

⑥ select c.name, o.orderid from customer as c inner join order as o on c.customer id = o.customerid

$\rightarrow \pi_{c.name,\ o.order\ id}\ \rho_c\text{(customer)} \bowtie_{(c.customer\ id = o.customerid)} \rho_o\text{(order)}$

⑦ select min(price) from products

$\mathcal{F}_{min(price)}$ (products)

⑧ select count(customer id), country from customers group by country;

$R1 \leftarrow {}_{country}G\text{ (customers)}$

$R_2 \leftarrow \pi_{\mathcal{F}_{count(customerid)},\ country}(R1)$

⑨ select count(customer id), country from customers group by country having count(customer id) > 5

$R1 \leftarrow {}_{country}G_{count(customer\ id) > 5}\text{(customer)}$ $\Rightarrow$ $R_2 \leftarrow \pi_{\mathcal{F}(count(customer\ id)),\ customer)}(R1)$

ρ_T = Table rename

$\rho_{T(c_1, c_2, c_3)}$ = column rename

π = select

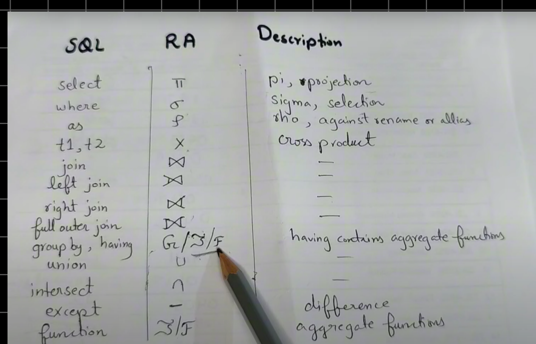
ρ = rename

σ = where

$\times$ = cross

$\bowtie$ = inner join condition in subscript

group by G having



SQL	RA	Description
select	π	pi, projection
where	σ	Sigma, selection
as	ρ	rho, against rename or allias
t1, t2	$\times$	Cross product
join	$\bowtie$	—
left join	⟕	—
right join	⟖	—
full outer join	⟗	—
group by, having	$G / \mathfrak{I} / \mathcal{F}$	having contains aggregate functions
union	$\cup$	—
intersect	$\cap$	—
except	$-$	difference
function	$\mathfrak{I} / \mathcal{F}$	aggregate functions

Select e.lastname, count (o.orderid) from orders as o inner join employee as e on o.employeeid = e.employeeid where lastname = 'Ahmad' group by lastname having count (o.orderid) >25

$R1 \leftarrow \rho_o(\text{order}) \bowtie_{(o.employeeid = e.employeeid)} \rho_e(\text{employee})$

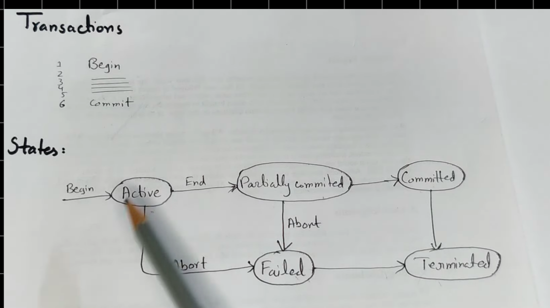
$R2 \leftarrow \sigma_{lastname = \text{'Ahmad'}}(R1)$

$R3 \leftarrow {}_{lastname}G_{count(o.orderid > 25)}(R2)$

$R_4 \leftarrow \pi_{(e.lastname,\ \mathcal{F}(count(o.orderid)))}(R3)$

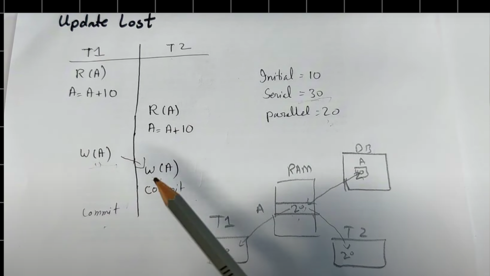
x ——— x

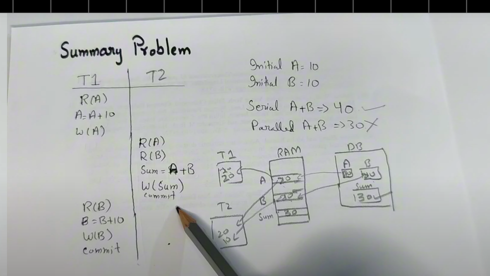
Transactions

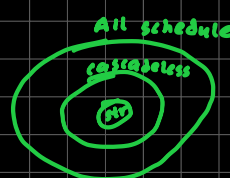


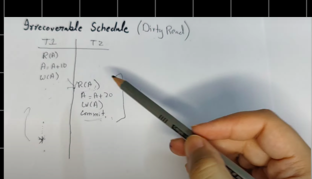
A Atomicity (all or none)
C consistency (state before and after remains valid)
I Isolation (implemented as single) (serial / parallel)
D Durability (No undo)

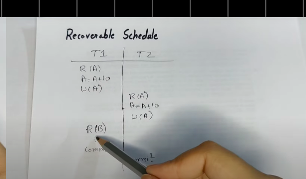
- Operations
- R(A) Read from RAM
- W(A) write on DB
- commit: Update DB

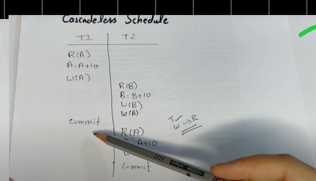




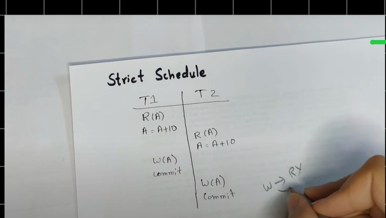








→ Recoverable
- Read after write not allowed until commit applied



→ Read/write not permitted after write until committed

- schedule → serializable

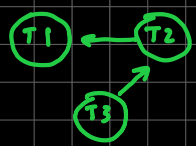
conflict $\xrightarrow{yes}$ serializable

no ↓

view $\xrightarrow{yes}$ serializable

view $\xrightarrow{no}$ non-serializable

- conflict pairs

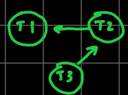


- R(A) → w(A)
- w(A) → R(A)
  → w(A)

- No loop → conflict serializable → serializable → consistent

- Finding out serial schedules

→ check for indegree = 0



- T3

- T3 → T2

- T3 → T2 → T1

- conflict equivalent
→ swap adjacent pairs if allowed
- If a conflict pair exists, then cannot swap.

T_1	T_2
	w(B)
w(A)	

→ Loop aagyi ab view serializability

	A	B	C
Initial Read	T_1, T_2	T_2, T_3	T_2
Update Read	x	x	x
Final write	T_1	T_2	T_3

read after write

A: $T_2 \to T_1$

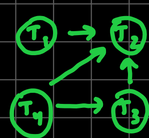
B: $T_3 \to T_2$

C: $T_2 \to T_3$

$T_3 \to T_2 \to T_1$

loop, not serializable

- For view equivalent draw tables, if tables are same, then view equivalent

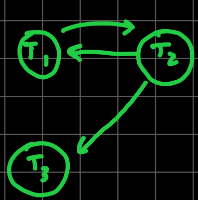


$T_4 \to T_3 \to T_2 \to T_1$

$T_4 \to T_1 \to T_3 \to T_2$

$T_1 \to T_4 \to T_3 \to T_2$

Q-



	A	B
Initial Read		
Update Read		
Final write		