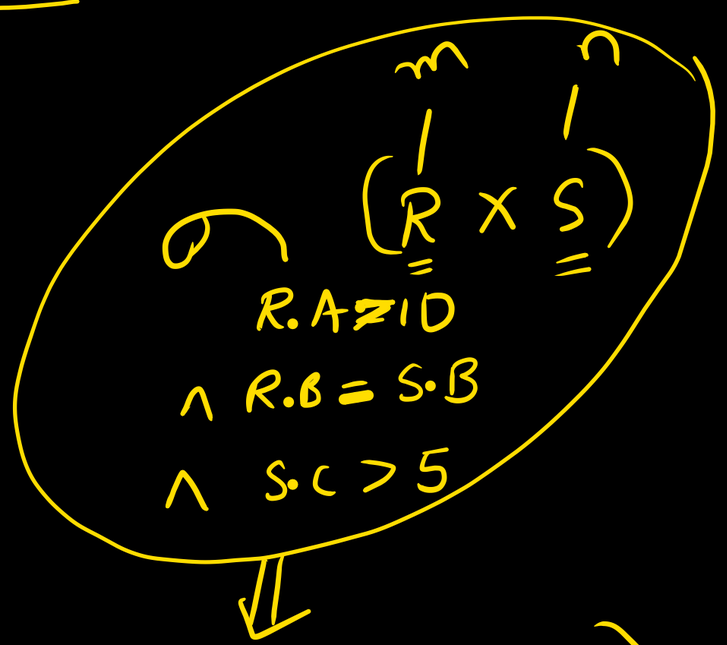
Query optimization:



If X will be applied then 3 Condition.

$R \times S = |(m \times n)|$ rows

3 Comparisons

$3 \times m \times n$ Comparisons

$\sigma_{R.B=S.B}\left(\sigma_{A>10} R \times \sigma_{C>5} S\right)$

$\times$ of $\bowtie$ of $\bowtie_C$.

Try to apply Comparisons before $\times$ of $\bowtie_C$ of $\bowtie$ Conditions.

$\sigma_{R.A>10}(R \bowtie S) \iff (\sigma_{A>10}(R)) \bowtie S$.

$\sigma_{R.A>10 \wedge S.C>5}(R \bowtie S) \iff (\sigma_{A>10} R \bowtie \sigma_{C>5} S)$

more no of rows to join

less no of rows to join

→ Enroll (Sid , Cid)

10000 rows

Course (Cid, Instr)

100 rows

σ (Enroll ⋈ Course)

Instr = Kalvh

⇒ ≡

(Enroll × Course)

E.Cid = C.Cid

σ

Instr = Kalvh

100000 × 100 = 10 lakhs

= 20 lakh comp.

2 Comparison of Condition

10000

Enroll × σ (Course)

E.Cid = C.Cid

Enroll Course

ex: 5 row 100

Instr = Kalvh

5 0000 × rows.

50,000 + 100

Emp(eid, gen, sal)

π_{eid}(Emp ⋈ ζ(Emp))

gen = fem

IGS

∧ G = male

∧ sal > 5

σ Emp
gen = female

Sal > 5.

σ ζ(Emp)
G=male I GS

Queries related to atleast / only / atmost :

Ex: Sid of students who took
" "
" "
" "

atleast 2 courses
only 2 courses
atmost 2 courses
atleast 100 courses
only 100
atmost 100

Enroll (Sid, Cid) Retrieve Sids who enrolled for at least two courses.

Enroll

Sid	Cid
S_1	C_1
S_1	C_2
S_1	C_3
S_2	C_1
S_2	C_2
S_3	C_1

Enroll

Sid	Cid
S_1	C_1
S_1	C_2
S_1	C_3
S_2	C_1
S_2	C_2
S_3	C_1

$\geq$ 2 courses.

$=$ 2 courses.

(2) 3, 4, 5.

Which means S_1 has taken 2 courses.

Enroll $\bowtie$ Enroll.

S_1 C_1 S_1 C_2

Course in diff

$S_1 \equiv S_1$

Sid are equal $\equiv$ same student

$$\pi_{T_1.sid}\left(\rho(T_1, enroll) \bowtie \rho(T_2, enroll)\right)$$

$T_1.sid = T_2.sid \rightarrow$ Same student

$T_1.cid \neq T_2.cid \rightarrow$ Two courses

(b1)

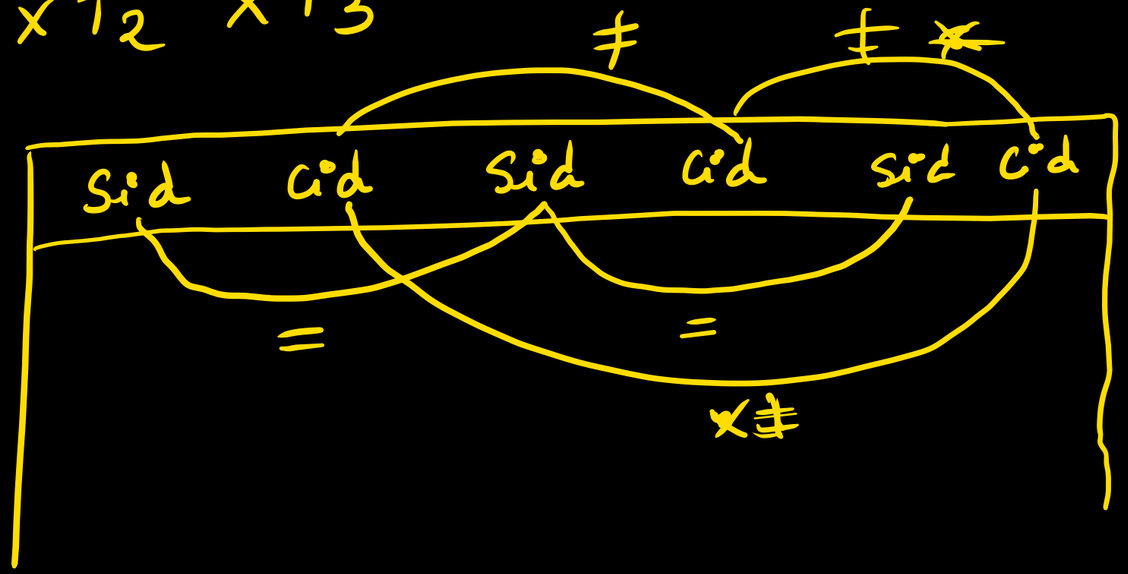
$$\pi_{sid}\left(enroll \bowtie_{sid=S \wedge cid \neq C} \rho_{S,C}(enroll)\right)$$

Enroll (Sid, Cid)

Sids of enrolled students who enrolled atleast 3 Courses.

$T_1, T_2, T_3 \rightarrow$ Enroll.

$T_1 \times T_2 \times T_3$



S_1 C_1 C_2 C_3 ✓

S_1 having (3) Courses.

≥ 3.

$=$

Sids enrolled only one course. of exactly 1 course.

{ Sids enroll at least one course } − { Sids enrolled at least two courses }

(1, 2, 3, 4, 5.....) − (2, 3, 4, 5, 6.....)

1 ✓

only → =

π_{sid} (enroll) − π (enroll ⋈ ρ enroll, sid=S S, C ∧ Cid ≠ C)

↓ atleast one course.

↓ atleast two

→ Sids enrolled in atleast 10 courses.

we need to put cross product on 10 tables.

But (Ugh. There is no other solution

But in SQL we can write it in simple way because of extra features like groupby, having etc.

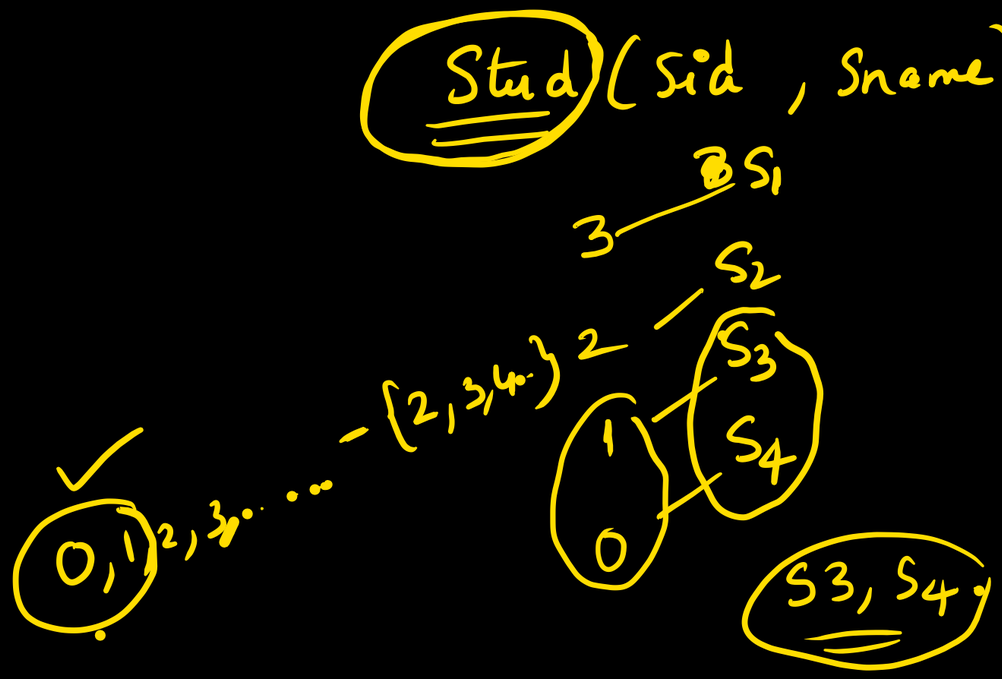
Sid enrolled "only two" courses

(Sids enrolled in atleast 2 courses) − (Sids enrolled in atleast 3 courses)

You do it as assignment.

Sid enrolled in at most one course

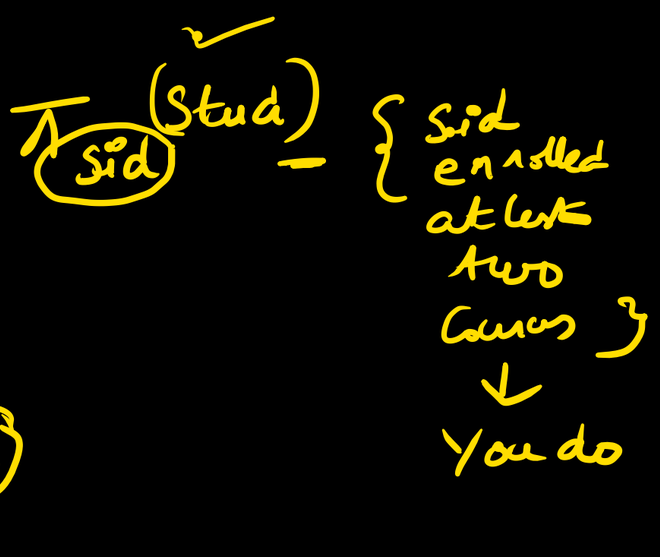
Stud(Sid, Sname)



Enroll(Sid, Cid)

s_1	c_1
s_1	c_2
s_1	c_3
s_2	c_1
s_2	c_2
s_3	c_1

→ s_4



{ all students } − { sid's enrolled at least two course }

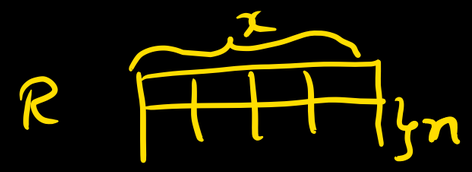
Sids enrolled at most two courses:

{ all students } − { Sids enrolled at least three courses }

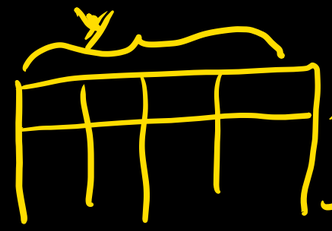
$\Pi_{sid}(stu)$ not $\Pi_{sid}(enroll)$

Some theory questions relation to division:

→ Rel R with x set of attributes & n distinct tuples.



→ Relation S with y set of attributes & m distinct tuples.



1) Condition required for R/S is $X \supseteq Y$

2) Resulting attribute set of R/S is $X - Y$

3) Cardinality of R/S min = 0

max = $\left\lceil \frac{n}{m} \right\rceil$

$\pi_{AB}(R) / \pi_{B}(S)$

$a_1\ b_1$
$a_1\ b_2$
$a_1\ b_3$

b_1
b_2
b_3

output (a_1, a_2)

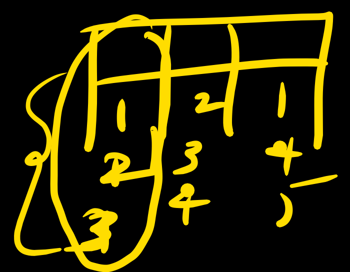
$a_2\ b_1$
$a_2\ b_2$
$a_2\ b_3$

$\left\lfloor \frac{n}{m} \right\rfloor$ $\frac{6}{3} = 2$

$\left\lfloor \frac{7}{3} \right\rfloor = 2$
3

Rel R with n distinct tuples, Rel S with m distinct tuples

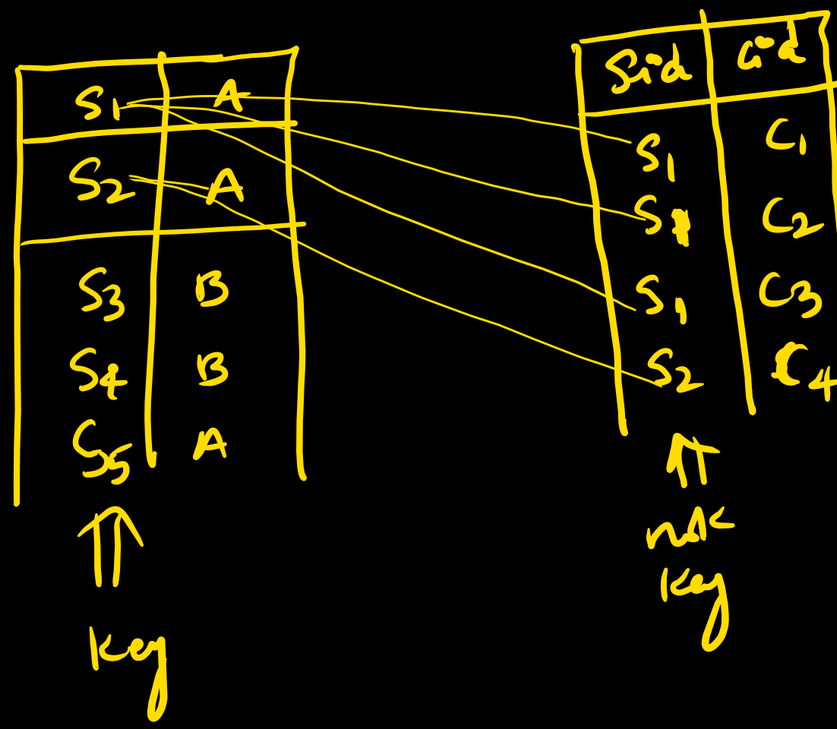
RA expression	Cardinality (min, max)	Commutative
1) $\pi_A(R)$	Atmost n = max; 1 = min	NO
2) $\sigma_c(R)$	min 0, max - n	yes
3) R × S	n × m	yes
4) R ⋈ S, R $⋈_c$ S	min = 0, max n * m	yes
5) R ⟕ S	min = n, max = n * m	NO
6) R ⟖ S	min = m max: n × m	NO



1) n

7	R ∩ S	$\max(n,m)$ to $m \times n$	Yes
8)	R ∪ S	$\max(n,m)$ to $m+n$ (min, max)	Yes
9	R ∩ S	0 to $\min(n,m)$	Yes
10	R − S	0 to n	NO
11	R / S	0 to $\lfloor n/m \rfloor$	NO.

Stud (Sid, Sname), Sid is primary key with 100 tuples, Enroll (Sid, Cid)
Sid Cid primary key with 50 tuples. How many (max, min) tuples
in result of Stud ⋈ ~~cross~~ Enroll.



min = no row matches = '0'

If Sid in FK, all the Sid's in Enroll are present in Stud.
∴ min = 50

RA ✓

SQL, ER, Trans, Indexes