

(3) SELECT A_1, A_2, \dots, A_N
(1) FROM R
(2) WHERE CONDITION

bag π - unless DISTINCT
is used in (3)

π_{A_1, \dots, A_N} (R))
CONDITION



Give all IDs and dates
for applications to 'CSC'
at 'NCSU'

Σ
DATE
ID
NCSU

SELECT ID, date
FROM Apply
WHERE MAJOR = 'CSC'
AND LOCATION
= 'NCSU'

Π ID, date
|
 \sqcap MAJOR = 'CSC'
AND
LOCATION = 'NCSU.'

ORDER BY DATE DESC,
ID ASC;

Apply



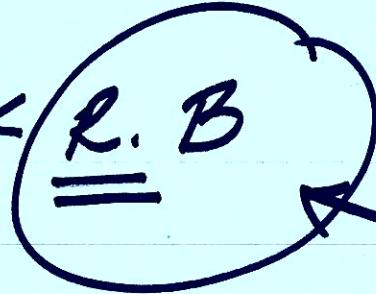
SELECT A₁, A₂, ..., A_M
FROM R₁, R₂, ..., R_N
WHERE C

\Downarrow bag π (unless DISTINCT)
 $\pi_{A_1, \dots, A_M} (\sigma_C (R_1 \times R_2 \times \dots \times R_N))$



\sqcap SELECT *
FROM \sqcup
WHERE $B < \underline{\underline{R.B}}$

$\pi_{B,C}(\sqcap_{\underline{\underline{B < R.B}}}^{(\sqcup)})$



syntax
problem

R:	A	B
	1	2
	4	0

S:	B	C
	1	3
	2	5

How to write
this query
syntactically
correctly



$R \bowtie S$ (with bag π in
the definition):

~~SELECT *~~ $\Leftarrow R.B, \cancel{S.B}, A, C$
~~FROM R, S~~
~~WHERE R.B = S.B;~~

$R(A, B)$
 $S(B, C)$



Names and application dates
for all prospective students
applying to CSC at NCSU,
with SAT > 1400

Select name, date AS applied
from Student, Apply A
where major = 'CSC' AND
location = 'NCSU' AND
SAT > 1400 AND
Student. ID = ~~Apply. ID~~
A. ID;



Student(ID, name, addr, GPA, SAT)

Return pairs of student names, for students who live at the

- each pair just once

SELECT $\downarrow 1.$ name, $\downarrow 2.$ name

FROM Student $\downarrow 1$, Student $\downarrow 2$

WHERE $\downarrow 1.$ addr = $\downarrow 2.$ addr

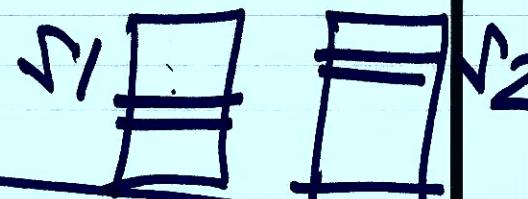
AND $\downarrow 1.$ ID < ~~$\downarrow 2.$ ID~~

DISTINCT



('n1', 'n2')
('n2', 'n1')

} same pair



SQL meaning

< standard meanings
>

\geq means \geq

\leq means \leq

\neq means \neq



$R(A, B)$

Select A
from R
where $B \leq 5$;

 $\pi_A(\sigma_{B \leq 5}(R))$

$R:$	A	B
	5	7
	6	3
	8	2
	6	1

does not go to answer

$\Rightarrow (6)$
 $\Rightarrow (8)$
 $\Rightarrow (6)$

SELECT
DISTINCT
 would remove duplicates



Patterns:
LIKE

(WHERE major LIKE

'C%')

length
↑
zero or more

UnderScore:
length 1

(WHERE phn LIKE '919-----')



THREE-VALUED LOGIC (for NULLs)

$R(a, b)$

SELECT *

FROM R

WHERE A=10 OR B=20;

R:	A	B	
	10	20	$\Rightarrow (10, 20)$
10	NULL		$\Rightarrow (10, \text{NULL})$
NULL	10		does not go to answer
20	10		does not go to answer



When we compare a constant
to in the query to a
NULL in the relation
⇒ the value of the answer
is UNKNOWN

FALSE < UNKNOWN < TRUE ;
a tuple goes into answer
only if the condition in
the WHERE clause
evaluates to TRUE



THREE-VALUED LOGIC :

$$\begin{array}{l}
 \checkmark y \text{ AND } z \rightarrow \min(y, z) \\
 \checkmark y \text{ OR } z \rightarrow \max(y, z) \\
 \text{NOT } z \rightarrow (1 - z)
 \end{array}$$

TRUE \rightarrow 1

FALSE \rightarrow 0

UNKNOWN \rightarrow 1/z (0.5)



Names of students whose
SAT score is not known,
but the GPA is known



Select name
from Student
where SAT IS NULL
AND GPA IS NOT NULL



$R(A, B, C)$

R :	A	B	C
	10	10	NULL

Select *

from R

where

$A = 10 \text{ AND } (B = 11 \text{ OR NOT } (C > 7))$

1 0 0.5
 0.5 0.5 0.5
 0.5

(E)