Joins!!

9/20

Equi-joins:

given relations r & s

$r \bowtie_{r.A=s.B} s$ ⇝

select * (proj (not important for this discussion))
from r, s
where r.A = s.B

⇝

select *
from r inner join s
on r.A = s.B

Algos:

* - nested loop join
* - block nested loop join
* - indexed (block) nested loop
- merge join
- hash join

Example $r \bowtie s$

- $n_r = 5{,}000$
- $b_r = 100$
- $n_s = 10{,}000$
- $b_s = 400$

Nested loop join

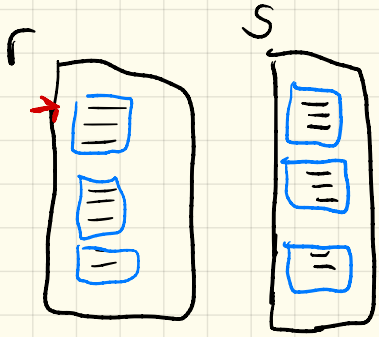
$r \bowtie_\theta s$

r outer relation
s inner relation

Cartesian product
$r = \{r_1, r_2, r_3, r_4\}$
$s = \{s_1, s_2\}$
$r \times s =$
$\{ r_1 s_1 \quad r_2 s_1 \quad r_3 s_1 \quad r_4 s_1 \quad r_1 s_2 \quad r_2 s_2 \quad r_3 s_2 \quad r_4 s_2 \}$

```
for each tuple t_r in r do begin
    for each tuple t_s in s do begin
        test pair (t_r, t_s) to see if they satisfy the join condition θ
        if they do, add t_r · t_s to the result;
    end
end
```

+ easy to implement
- slow
+ useful regardless of conditions
+ no indicies



Analysis:
- # of tuples $n_r * n_s$ (# of tuples in r, # of tuples in s)
- assume buffer size is 1 block per relation
  - block transfers: $n_r * b_s + b_r$
    (n_r: # of block w/ tuples from s; b_r: # of blocks w/ tuples from r)
- # of seeks

Which should be r and which should be s

- $b_r + b_s$
- $b_r + n_r$