TECH VAULT

JOIN ALGORITHMS

- NESTED LOOP
- INDEX NESTED LOOP
- MERGE
- HASH

AMR ELHELW

Customers

cid	cname	email	Region
c1	Alice	alice@example.com	West
c2	Bob	bob@example.com	East
c3	Charlie	charlie@example.com	North
c4	Danny	danny@example.com	East

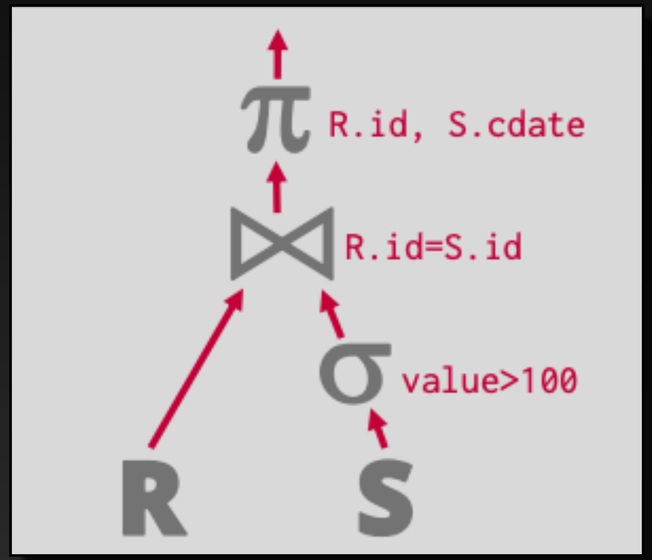
Orders

oid	odate	cid	amount
101	2022-04-10	c2	$150
102	2023-05-07	c1	$20
103	2021-10-04	c4	$37
104	2021-10-04	c1	$126
105	2023-06-20	c1	$30
106	2022-07-25	c4	$74

“Return all customer names and order amounts from the East region”

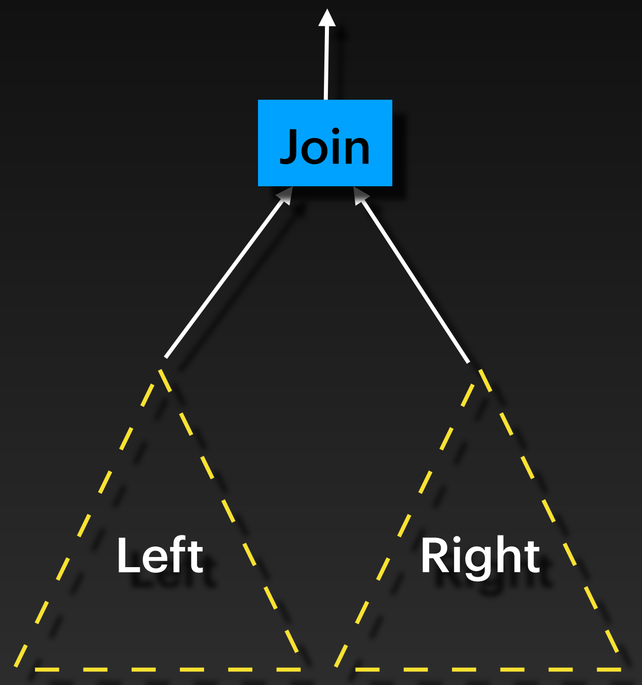
Join Query

```
SELECT R.id, S.cdate
FROM R JOIN S
ON R.id = S.id
WHERE S.value > 100
```



Join Plan

- Each input can be a subplan
  - Left (outer)
  - Right (inner)
- Each join algorithm has a different cost
- Cost can be estimated in terms of number of rows fetched from each side



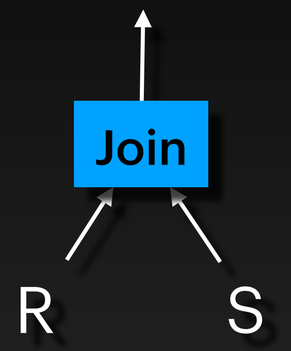
Nested Loop Join

```
SELECT *
FROM R JOIN S
ON R.id = S.id
```

```
foreach tuple r in R:
  foreach tuple s in S:
    if (condition): output row
```

Outer

Inner



R

id	name
3	Alice
2	Bob
1	Charlie
4	Danny

S

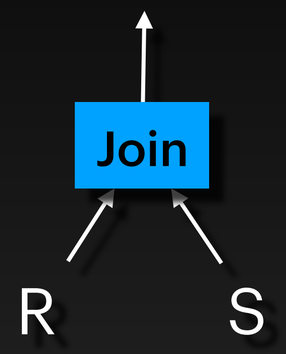
id	Value
2	1000
3	2000
4	1300
2	5000
4	2500

Output

R.id	R.name	S.id	S.value
3	Alice	3	2000
2	Bob	2	1000
2	Bob	2	5000
4	Danny	4	1300
4	Danny	4	2500

Nested Loop Join

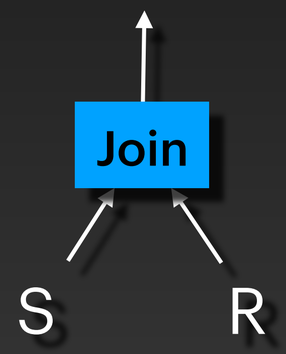
```
SELECT *
FROM R JOIN S
ON R.id = S.id
```



Cost = $N_R + (N_R * N_S)$

N_R: #rows in R

N_S: #rows in S



Cost = $N_S + (N_S * N_R)$

Better to use the smaller input on the left side!

Nested Loop Join

```
SELECT *
FROM R JOIN S
ON R.id = S.id
```

```
foreach tuple r in R:
  foreach tuple s in S:
    if (condition): output row
```

Search for tuples in S with id = *r.id*

What if there was already an index on S.id?

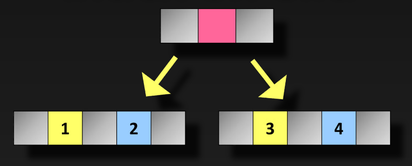
```
foreach tuple r in R:
  SS = Search Index for r.id
  foreach tuple s in SS:
    output row
```

Index Nested Loop Join

Index Nested Loop Join

```
SELECT *
FROM R JOIN S
ON R.id = S.id
```

Index on S.id



R

id	name
3	Alice
2	Bob
1	Charlie
4	Danny

S

id	Value
2	1000
3	2000
4	1300
2	5000
4	2500

Output

R.id	R.name	S.id	S.value
3	Alice	3	2000
2	Bob	2	1000
2	Bob	2	5000
4	Danny	4	1300
4	Danny	4	2500

Merge Join

```
SELECT *
FROM R JOIN S
ON R.id = S.id
```

R

id	name
3	Alice
2	Bob
3	Alice
4	Danny

S

id	Value
2	1000
3	2000
4	1300
4	2500

Output

R.id	R.name	S.id	S.value
2	Bob	2	1000
2	Bob	2	5000
3	Alice	3	2000
4	Danny	4	1300
4	Danny	4	2500

```sql
SELECT *
FROM R JOIN S
ON R.id = S.id
```

If there was already an index on S.id

```
foreach tuple r in R:
  SS = Search Index for r.id
  foreach tuple s in SS:
    output row
```

Index Nested Loop Join

If there was no index S.id

Build an “index” on the fly?

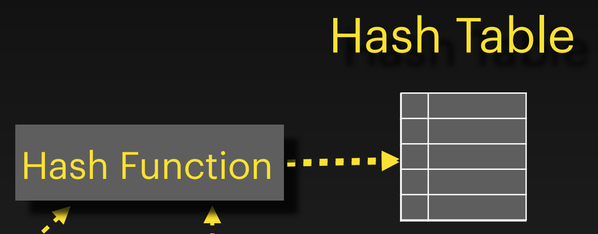
Focus on equi-joins —> Hash index

```
Build Hash table on S.id
foreach tuple r in R:
  SS = Search Hash table for r.id
  foreach tuple s in SS:
    output row
```

Hash Join

Hash Join

```
SELECT *
FROM R JOIN S
ON R.id = S.id
```



R

id	name
3	Alice
2	Bob
1	Charlie
4	Danny

S

id	Value
2	1000
3	2000
4	1300
2	5000
4	2500

Output

R.id	R.name	S.id	S.value
3	Alice	3	2000
2	Bob	2	1000
2	Bob	2	5000
4	Danny	4	1300
4	Danny	4	2500