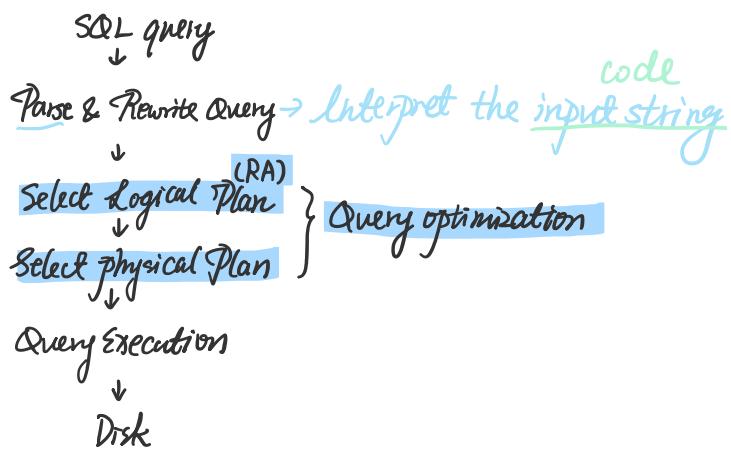


RDBMS internals

Monday, May 6, 2019 1:30 PM

- From Logical RA plans to physical plan.



- Main Memory Algorithms

- Three algorithms: Supplier $\propto_{s,d=sid}$ Supply

I. Nested Loops $O(n^2)$ ($|S|=|R|=n$)

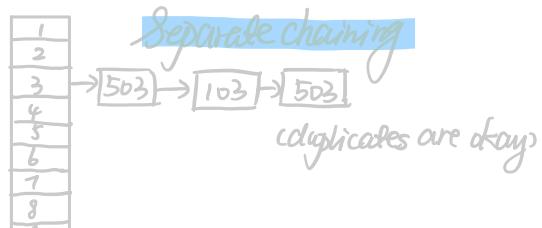
II. Hash join $O(n)$ if having good hash function

- Hash table (array of fixed length)

$$h(x) = x \bmod 10$$

insert(k, v)

find(k) return the list
of values of key k.



primary key

<pre> for x in Supplier do insert(x.sid,x); } </pre>	$\Rightarrow O(n)$ $ S = R = n$
<pre> for y in Supply do x = find(y.sid); output(x,y); } </pre>	$\Rightarrow \text{hash join } O(n)$ $O(n)$ if hash function is bad, $O(n^2)$ might result.

<pre> for y in Supply do insert(y.sid,y); for x in Supplier do for y in find(x.sid) do output(x,y); } </pre>	\Rightarrow $O(n)$ $ R = S = n$ but can be $O(n^2)$ if use non key e.g. if all supply come from one supplier if data in supplier is huge.
--	---

III. Merge join $O(n\log n)$

`Sort(Supplier); Sort(Supply);` Need to have key.

`x = Supplier.first();`

`y = Supply.first();`

`while y != NULL do`

`x.sid < y.sid : x = x.next();`

`x.sid = y.sid : output(x,y); y = y.next();`

`x.sid > y.sid : y = y.next();` not possible because foreign key.
 but should still write.

• Execution

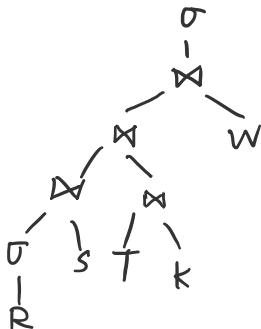
- Index-based selection
 - on the fly (just iterate through and check)
- Group by Y
 - Hash-based search, save the previous
 - Merge-based same group will be ordered together

• Combine: The Iterator Interface

interface Operator {

```
void open(); // initialize operator state  
           // sets parameters  
// process an input tuple  
// produce output tuple(s), return null when done  
Tuple next();
```

```
// cleans up  
void close();
```



}