

## (Volcano) Iterator Model

- everything implements ONC interface
  - open
  - next
  - close
- query execution from root node
- hierarchical and scalable
- everything one by one
- blocking operations may prevent this
  - sorting/grouping/aggregation/hash joins
  - require knowledge of all tuples not just one
- e.g.

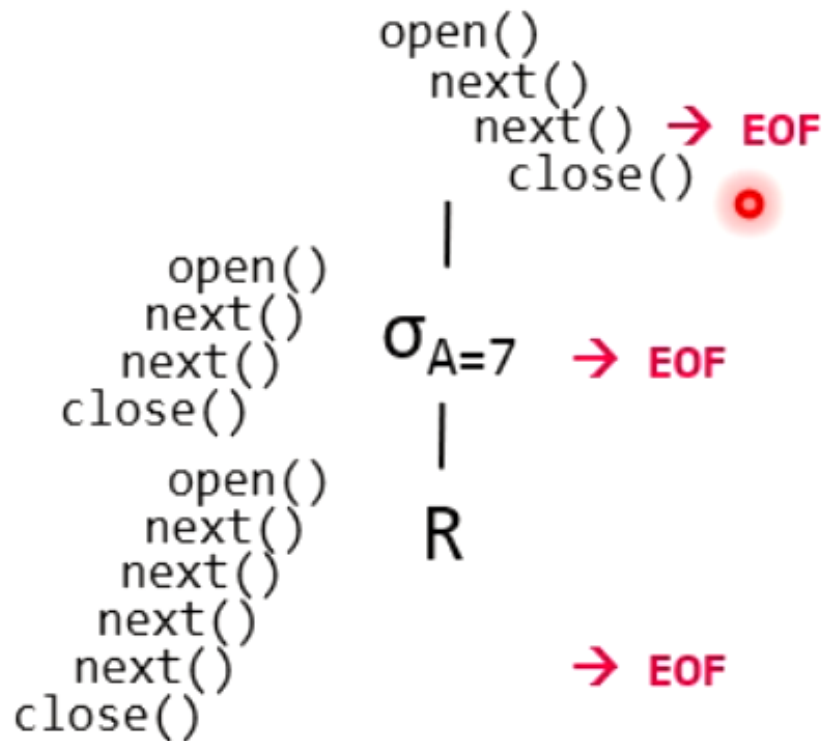
### Example $\sigma_{A=7}(R)$

```
void open() { R.open(); }
```

```
void close() { R.close(); }
```

```
Record next() {  
    while( (r = R.next()) != EOF )  
        if( p(r) ) //A==7  
            return r;  
    return EOF;  
}
```

–



–

### Physical Table Access Operator

- seq scan
  - sequential read of table
  - reading tuples one by one
- index scan
  - first reads all indexes
  - throwing all away which do not satisfy certain criteria
  - reads remaining attributes afterwards
- index only scan
  - only reads indexes

### Physical Join Operators

- nested loop join
  - like two nested for loops
  - for every tuple in table A iterate over every tuple in table B
  - slow
- hash/hash join
  - only for equi joins
  - smaller table A is read first
  - create hashmap out of A

- if value in B equal to A ==> hash equal
- access via hashmap
- fast
- sort/merge join
  - no clue... VO#04 1:28
  - efficient if one table is already sorted

## Physical Grouping Operators

- hash aggregate
  - groups into hash tables
  - useful for additive/incremental aggregations
- group aggregate
  - sorting
  - group by easy if sorted

## Analyzing/Explaining Queries

- EXPLAIN command before SQL-query
- returns query tree
  - physical operators instead of SQL operators
- EXPLAIN does not update regularly
  - ANALYZE beforehand necessary

▪ **Step 1: EXPLAIN SELECT \* FROM Participant AS R, Locale AS S**  
**WHERE R.LID=S.LID;**

Hash Join (.. rows=70 width=1592)

Hash Cond:(s.lid = r.lid)

-> Seq Scan on locale s (.. rows=140 width=520)

-> Hash (.. rows=70 width=1072)

-> Seq Scan on participant r (.. rows=70 width=1072) } build side

▪ **Step 2: ANALYZE Participant, Locale;**

▪ **Step 3: EXPLAIN SELECT \* FROM Participant AS R, Locale AS S**  
**WHERE R.LID=S.LID;**

Hash Join (.. rows=17 width=47)

Hash Cond:(r.lid = s.lid)

-> Seq Scan on participant r (.. rows=17 width=30)

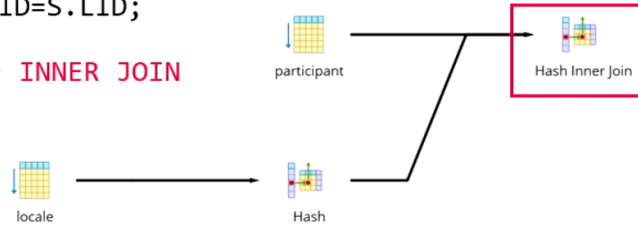
-> Hash (.. rows=11 width=17)

WHY?

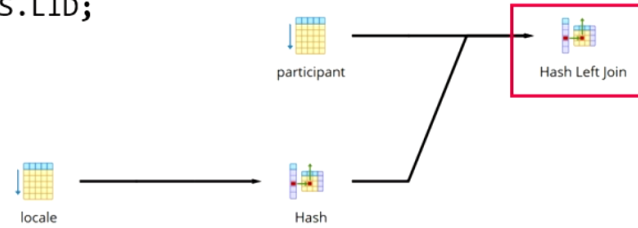
- -> Seq Scan on locale s (.. rows=11 width=17)
- Visual EXPLAIN

▪ **SELECT \* FROM Participant AS R, Locale AS S**  
**WHERE R.LID=S.LID;**

$\sigma_F(R \times S) \rightarrow$  **INNER JOIN**



▪ **SELECT \* FROM Participant AS R LEFT JOIN Locale AS S**  
**ON R.LID=S.LID;**



[[Relational Algebra]]