COL362/632 Introduction to Database Management Systems

Query Processing - Processing Models

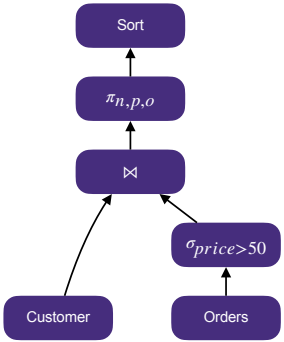
Kaustubh Beedkar

Department of Computer Science and Engineering
Indian Institute of Technology Delhi

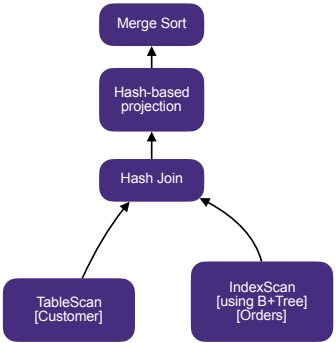
Query Processing (Overview)

- SQL : What?
- Physical plan: How?
- Query processing and optimization "compiles" a logical plan in to a physical plan

```
SELECT c.name, o.price, o.order_date
FROM customer c, orders o
WHERE c.customer_id = o.customer_id
AND o.price > 50
ORDER BY o.order_date DESC;
```



Logical Plan



Physical Plan

Outline

Processing Models

1. Materialization model
   - Evaluate one operation at a time, and materialize intermediate output
2. Vocano model

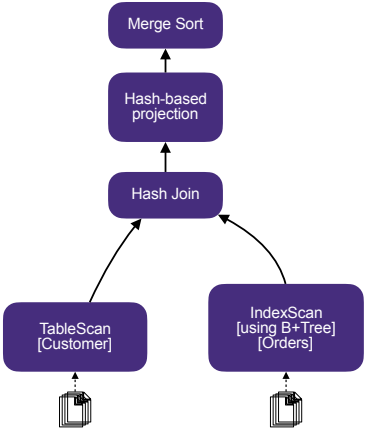
   – also known as pipelining model or iterator model
   - Evaluate multiple operations in parallel in a pipline

Outline

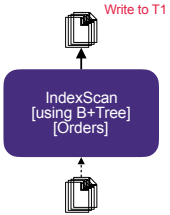
Materialization Model

- Tuples generated by an operator are written to disk in intermediate table
- Has no direct benefit!
- But, necessary
  - For certain operator implementations
  - When there is not enough memory

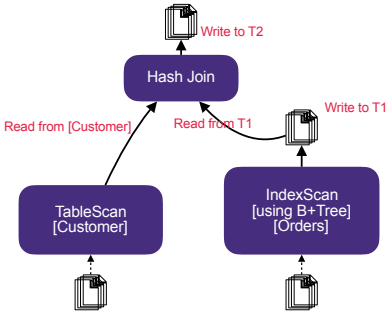
Materialization Model (Example)



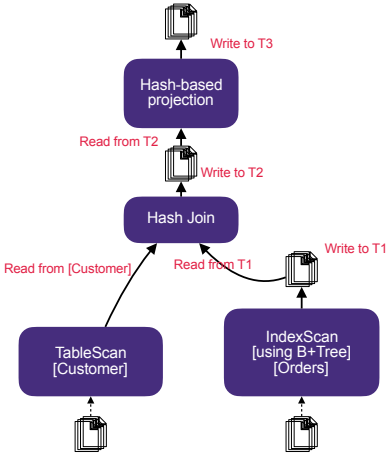
Materialization Model (Example)



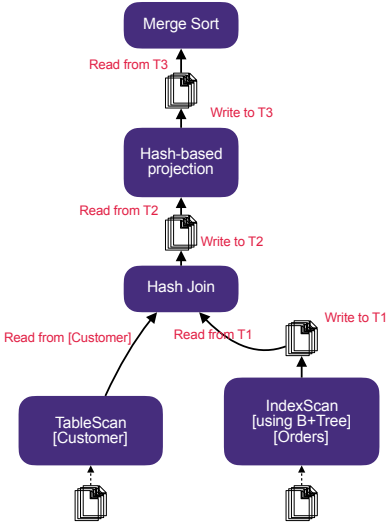
Materialization Model (Example)



Materialization Model (Example)



Materialization Model (Example)



Outline

Iterator Interface

- Analogous to constructors and destructors
- Each operator implements the three functions
- Iterator maintains the state of its execution

Iterator Interface

- Analogous to constructors and destructors
- Each operator implements the three functions
- Iterator maintains the state of its execution

open()

- Initialize the operator state
- sets parameters (such as selection condition)

next()

- Performs processing and produces an output tuple
  - each invocation, operator returns either a single tuple of EOF
- Operator invokes the next function recursively on its inputs

close()

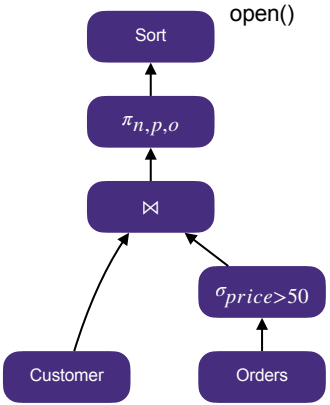
- Clean up

Selection Operator (Java Example)

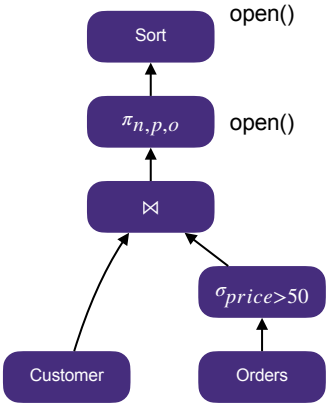
```java
interface Operator<T>{
    void open();
    T next();
    void close();
}
```

```java
class Selection implements Operator<Row>{
    private Operator<Row> input;
    private Predicate<Row> predicate;
    public Selection(Operator<Row> input, Predicate<Row> predicate)
        {
        this.input = input;
        this.predicate = predicate;
    }

    public void open() {
        input.open();
    }

    public Row next() {
        for(Row row = input.next; row!=null; row=row.next()) {
            if(predicate.check(row)) {
                return row;
            }
        }
        return null;
    }

    public void close() {
        input.close();
    }
}
```

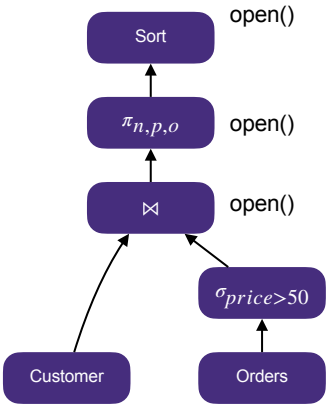
Pipeline Execution (Illustration)



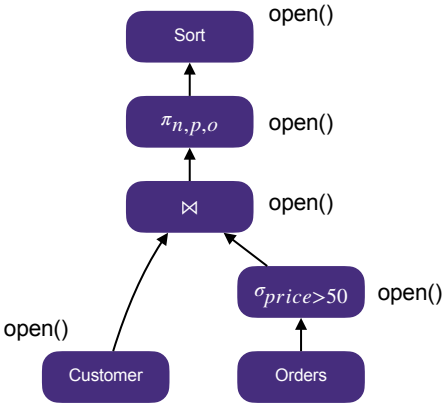
Pipeline Execution (Illustration)



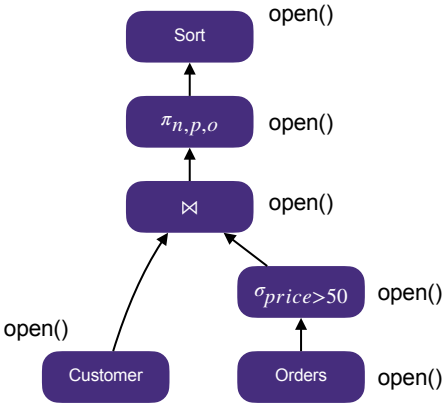
Pipeline Execution (Illustration)



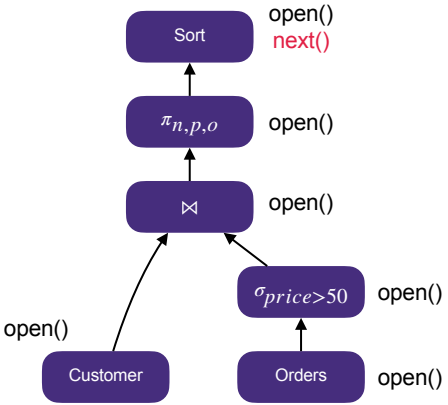
Pipeline Execution (Illustration)



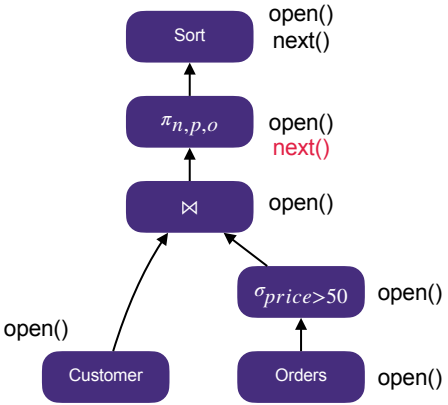
Pipeline Execution (Illustration)



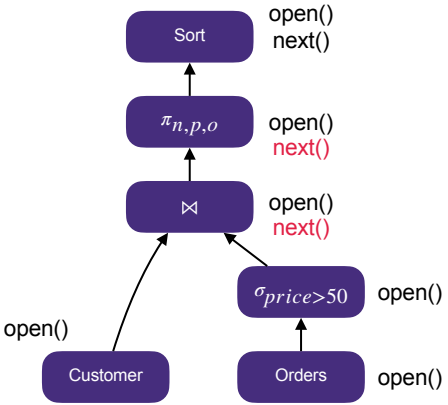
Pipeline Execution (Illustration)



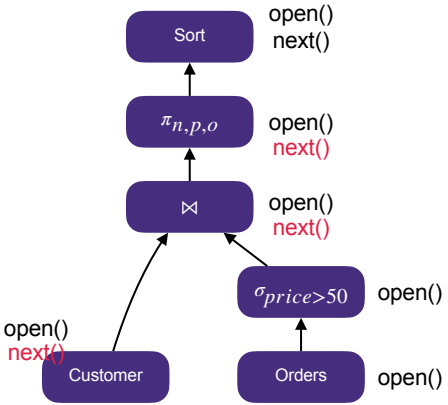
Pipeline Execution (Illustration)



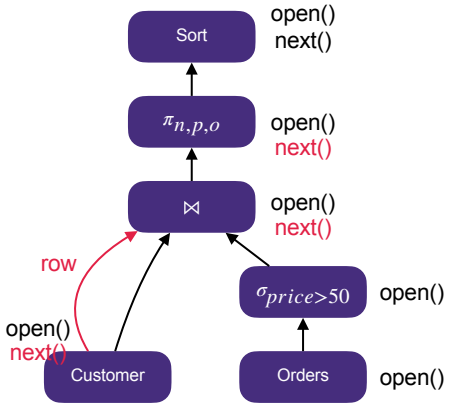
Pipeline Execution (Illustration)



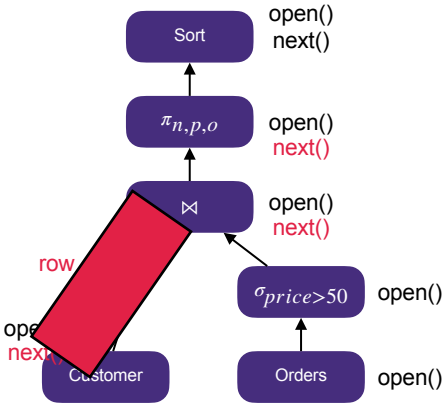
Pipeline Execution (Illustration)



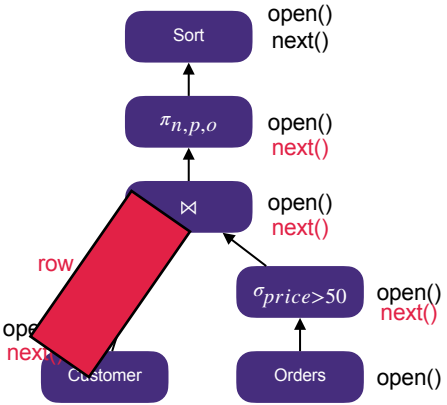
Pipeline Execution (Illustration)



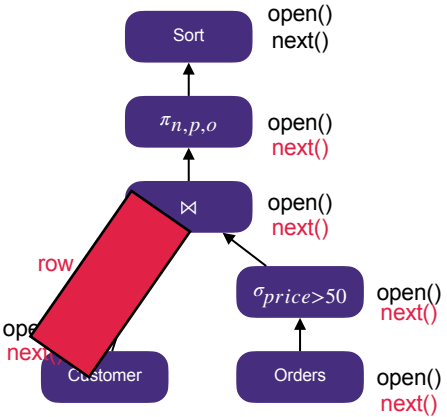
Pipeline Execution (Illustration)



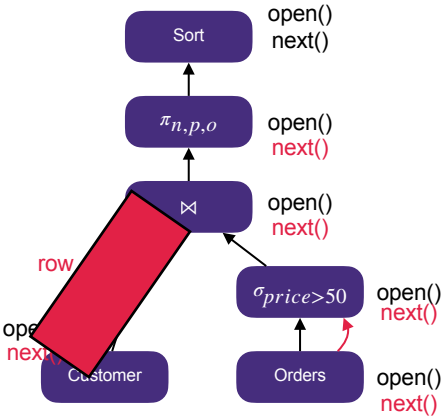
Pipeline Execution (Illustration)



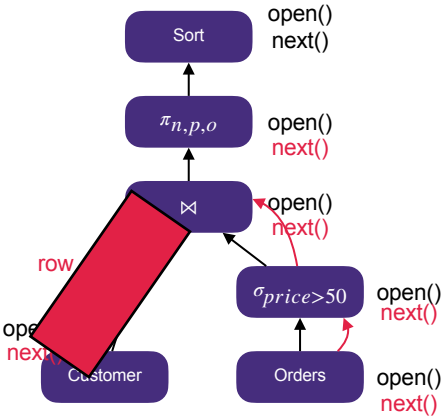
Pipeline Execution (Illustration)



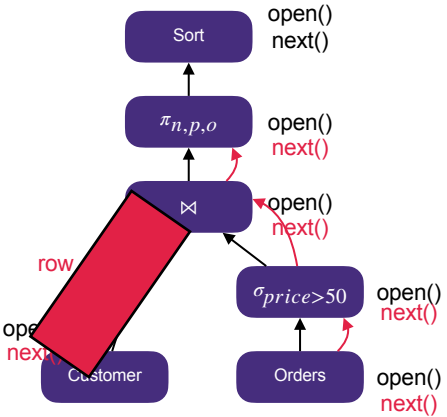
Pipeline Execution (Illustration)



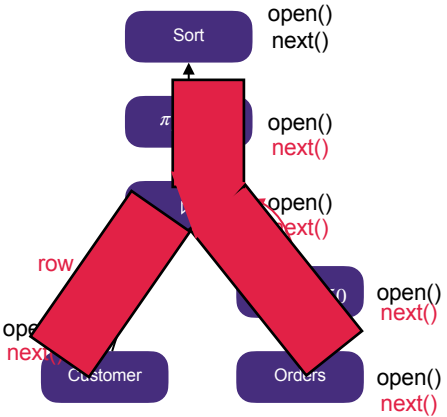
Pipeline Execution (Illustration)



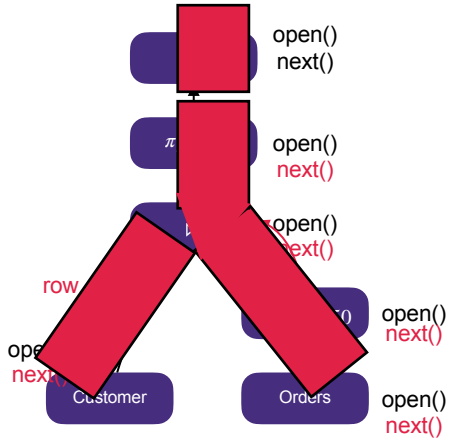
Pipeline Execution (Illustration)



Pipeline Execution (Illustration)



Pipeline Execution (Illustration)



Pipelined Execution

- Tuples generated by an operator are immediately sent to the parent
- Benefits
  - No operator synchronization issues
  - Saves cost of writing intermediate data to disk
  - Saves cost of reading intermediate data from disk
- This approach is used whenever possible

Memory Management

Each operator

- Pre-allocates heap space for tuples
  - Pointers to base data in buffer pool
    or new tuples in the heap
- Allocates memory for its internal states
  - Either on heap or buffer pool (depends on system)
- DBMS may limit how much memory each operator, or each query can use