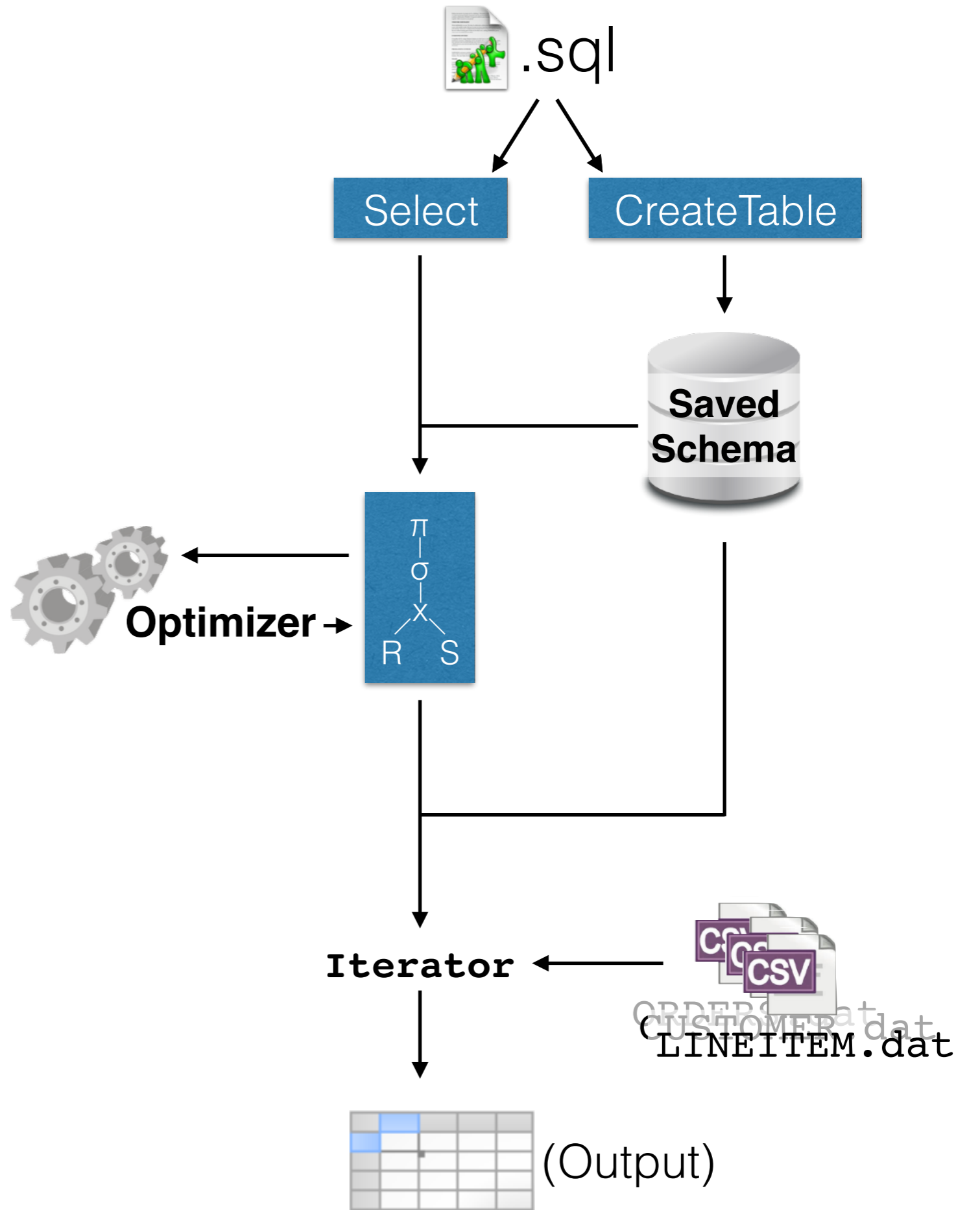
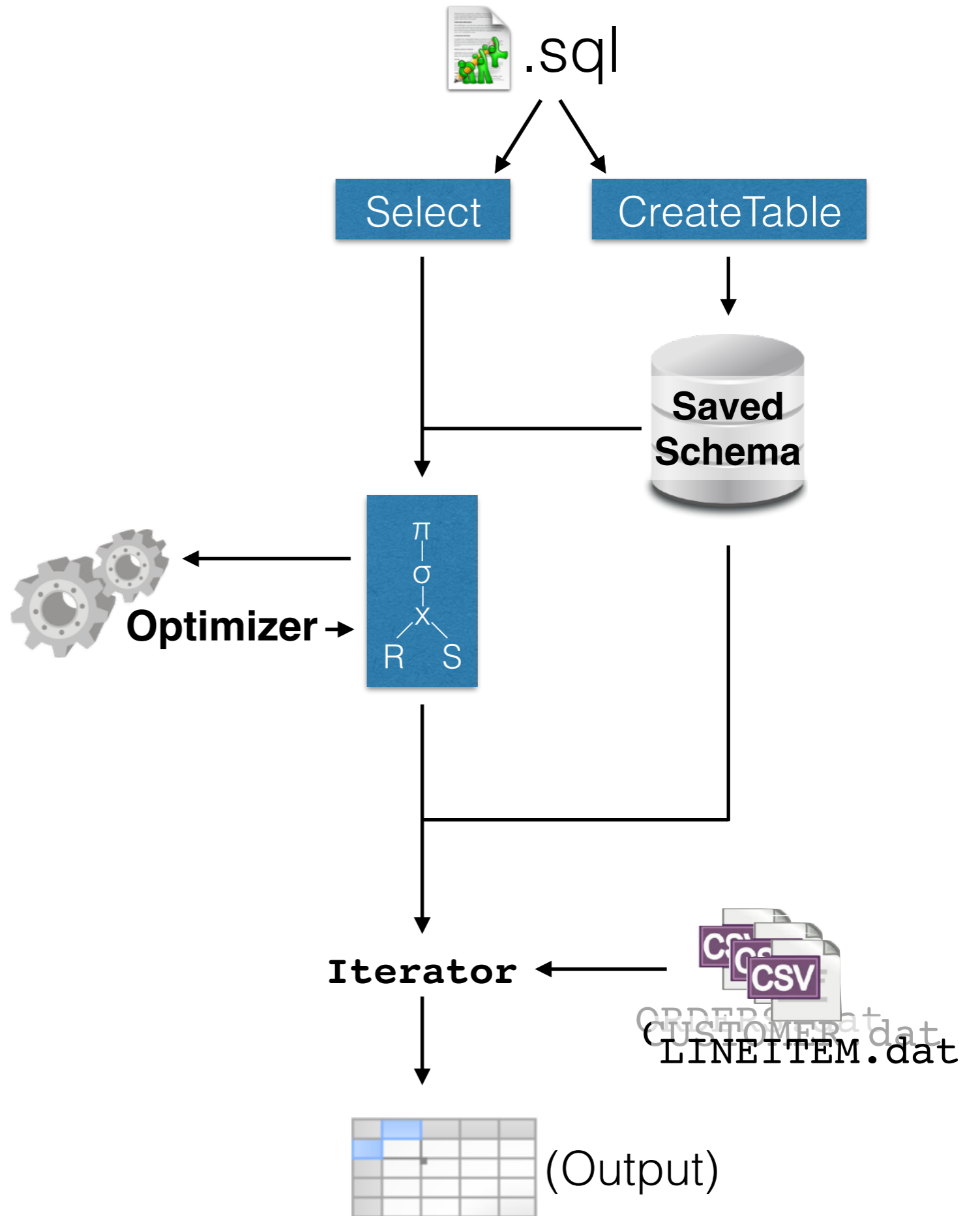
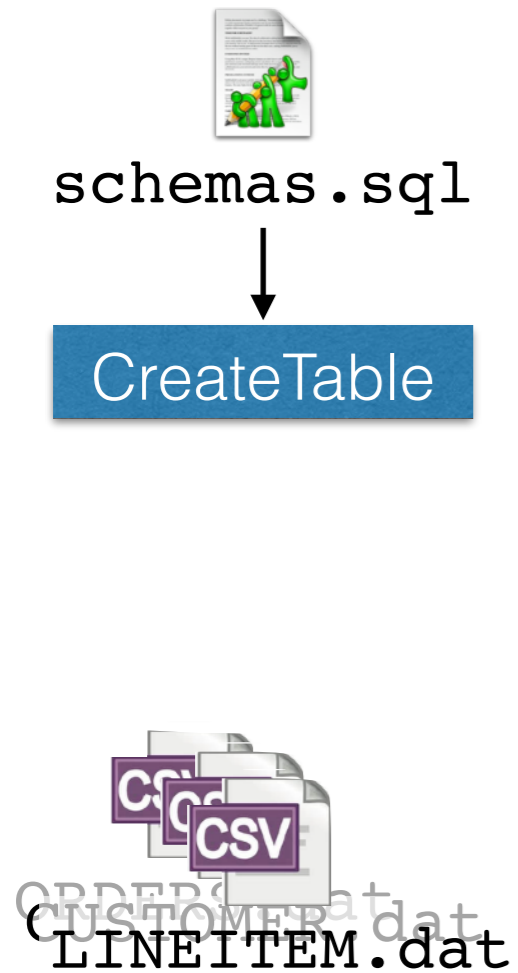


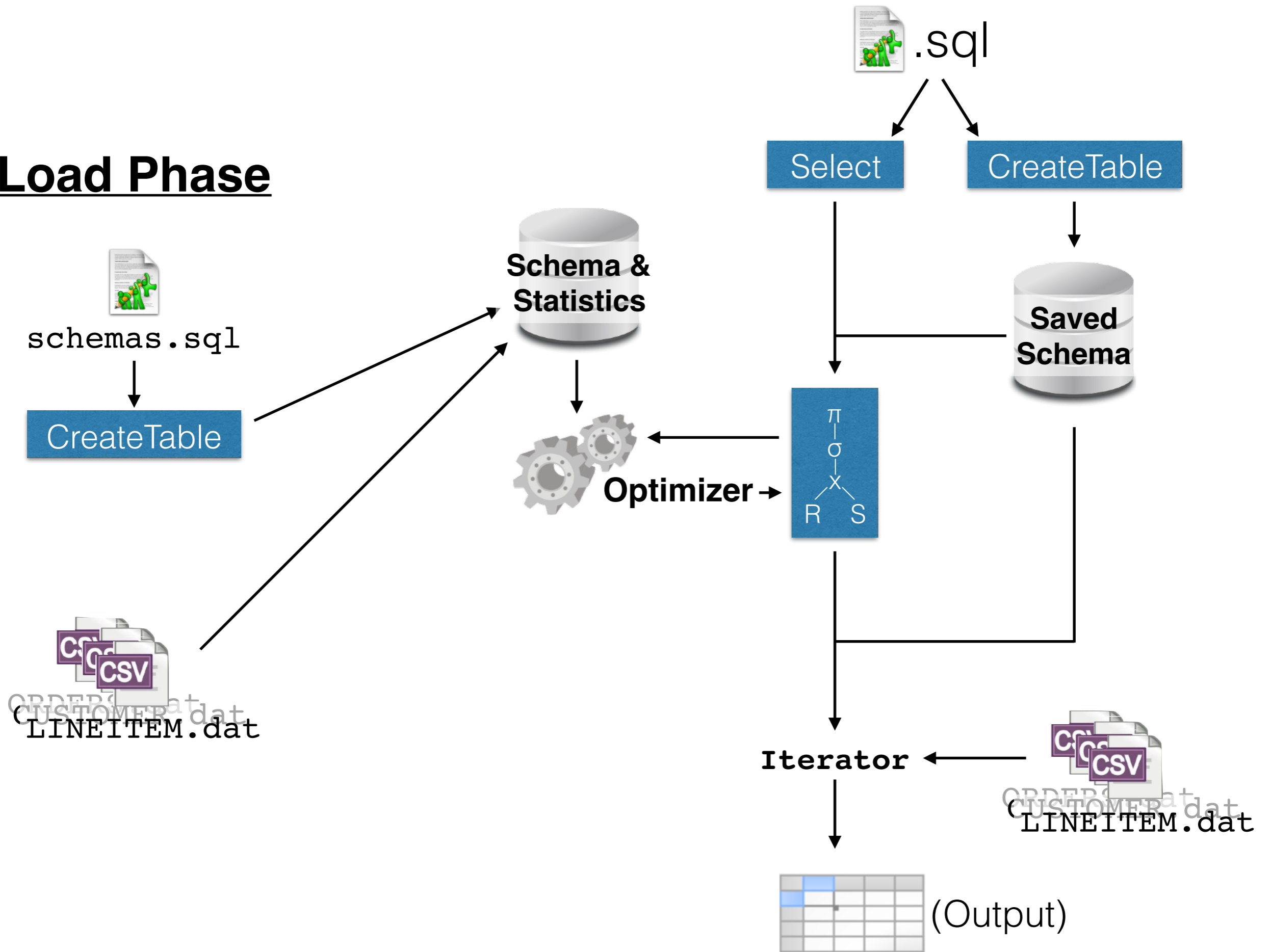
Project 4 Review



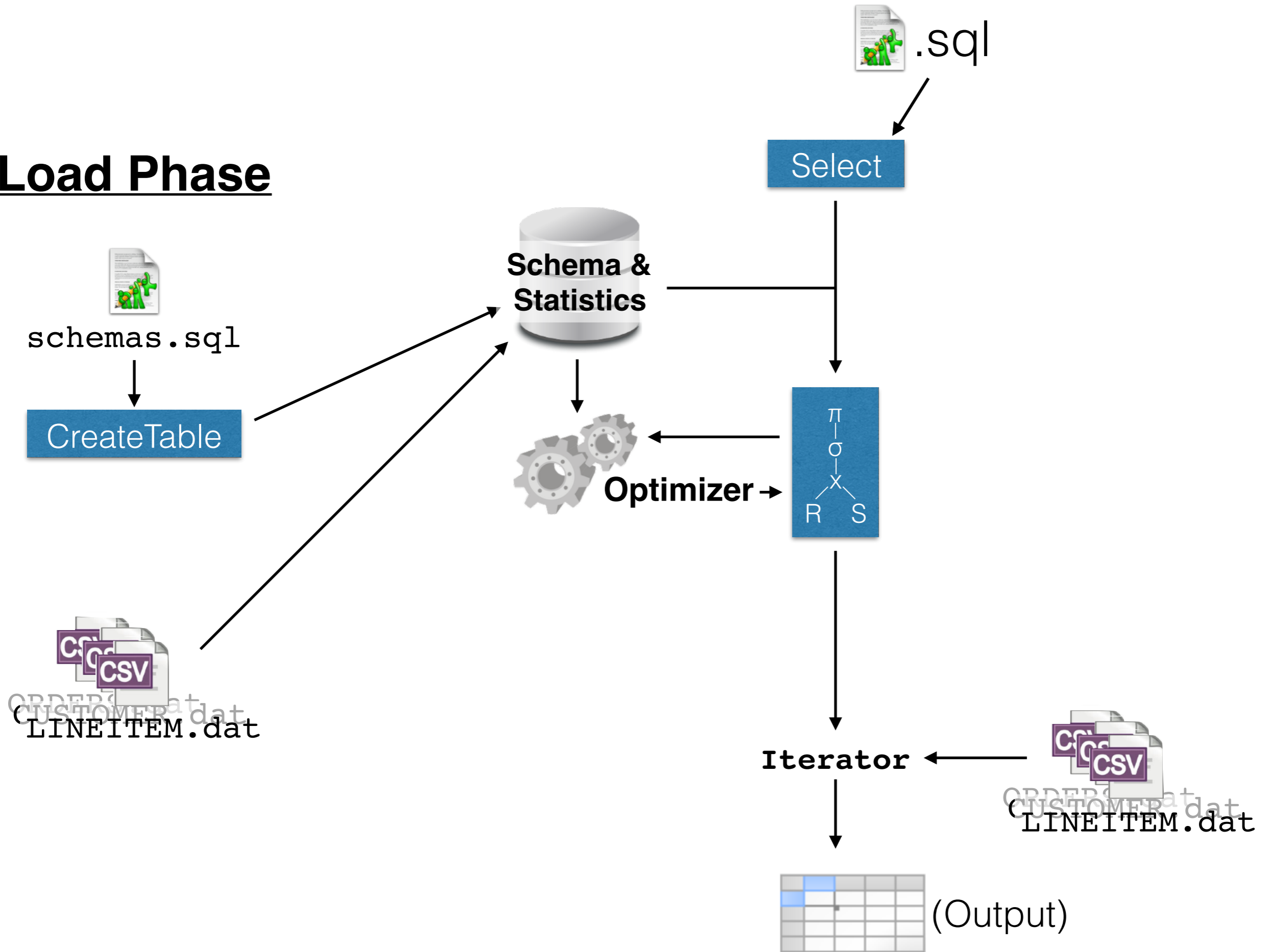
Load Phase



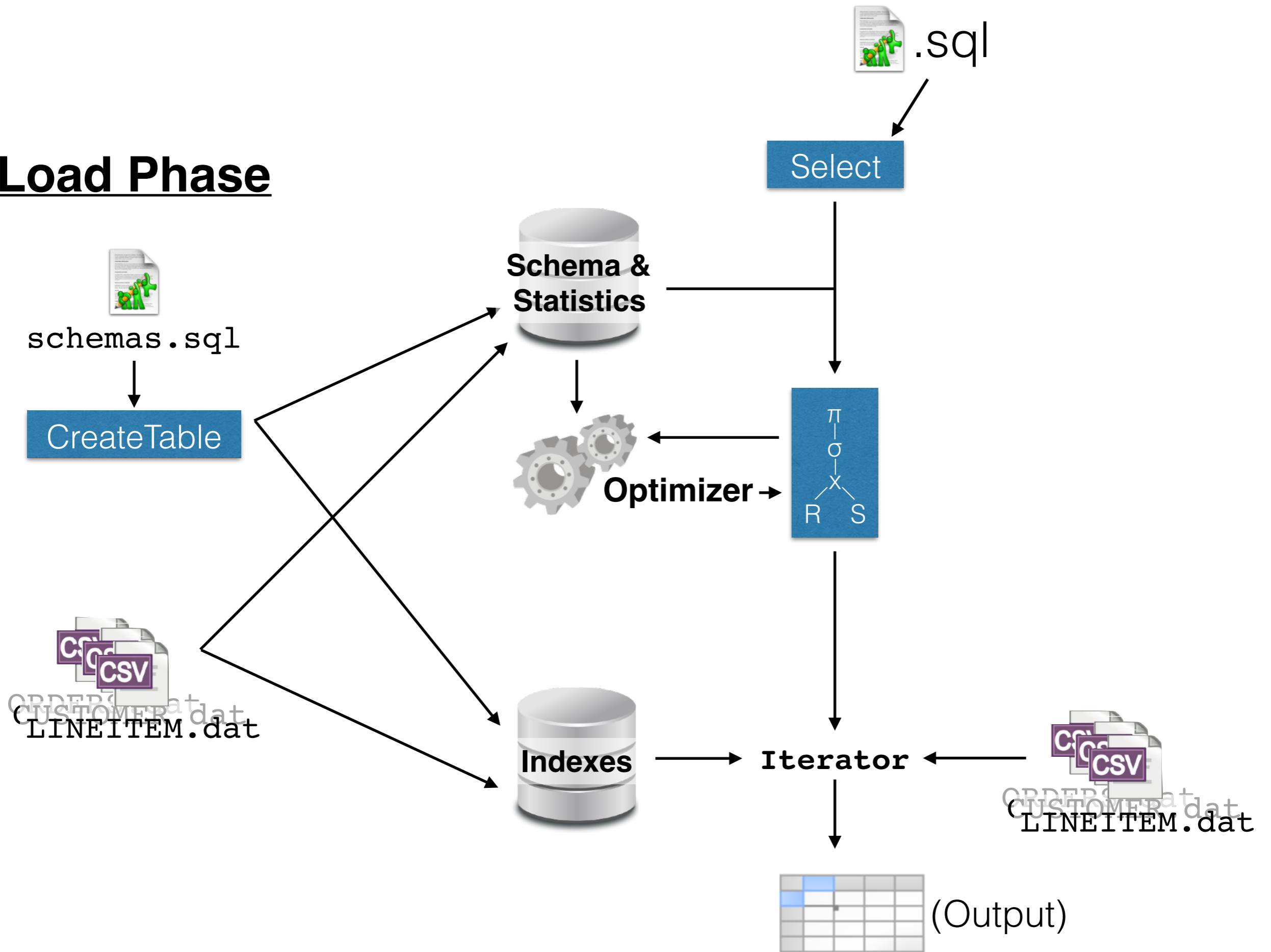
Load Phase



Load Phase



Load Phase



Load Phase

You will be given n number of CreateTable statements
(number will be announced)

Do not print prompt until you process the data

You have 5 minutes with the data

Finally, print the next prompt

Query Phase

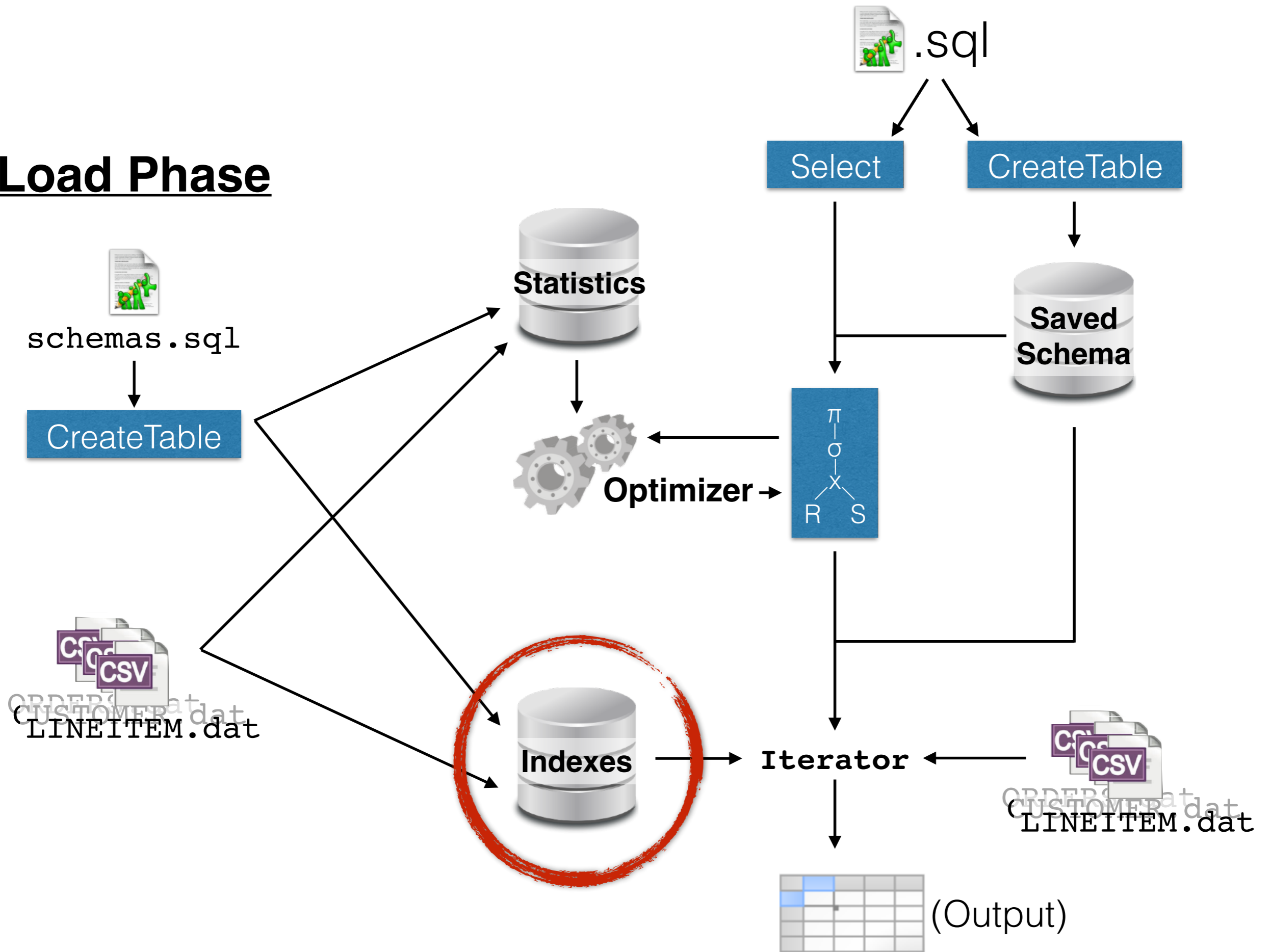
As before

Needs to be faster

Needs to run with a very limited memory available

Hint: External Sort & Indexing & Buckets

Load Phase



Serializing Records

Option 1: Object{In|Out}putStream

Faster! (Smaller data, Object serialization better than Strings)

```
public class Tuple implements Serializable { ... }
```

```
Tuple t = ...;  
ByteArrayOutputStream out  
    = new ByteArrayOutputStream();  
ObjectOutputStream objOut  
    = new ObjectOutputStream(out);  
objOut.writeObject(t);  
byte[] tupleData = out.toByteArray();  
  
... proceed as before ...
```

Serializing Records

Option 1: ObjectOutputStream

Faster! (Smaller data, Object serialization better than Strings)

```
... get tupleData byte array as before ...  
  
ByteArrayInputStream in  
    = new ByteArrayInputStream(tupleData);  
ObjectInputStream objIn  
    = new ObjectInputStream(in);  
Tuple t = objIn.readObject(t);
```

Serializing Records

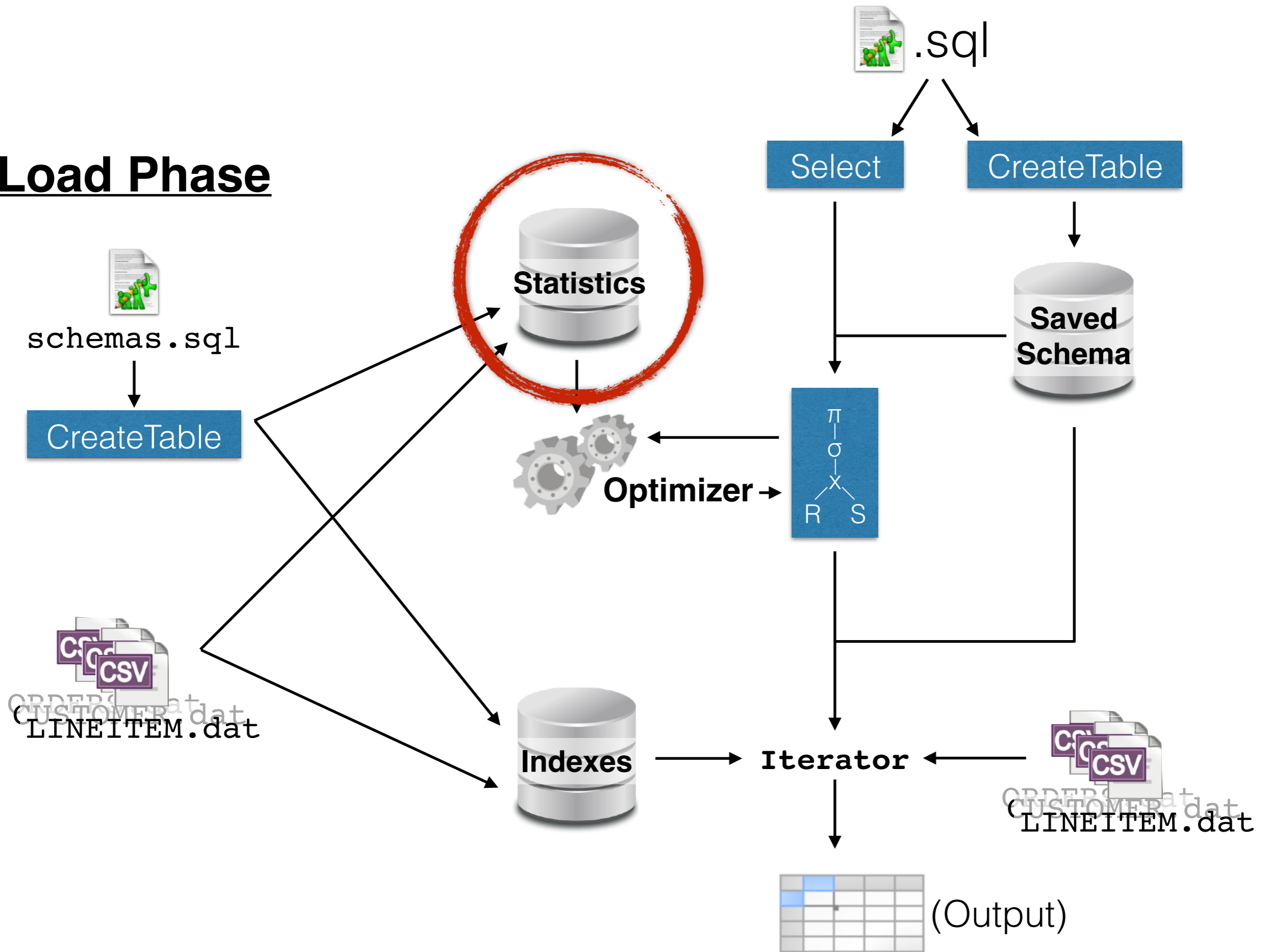
Option 2: Data{In|Out}putStream

Fastest! (Tiny data, No Reflection overheads)

```
Tuple t = ...;
ByteArrayOutputStream out
    = new ByteArrayOutputStream();
DataOutputStream dataOut
    = new DataOutputStream(out);
// dataOut.writeDouble(d);
// dataOut.writeLong(l);
// dataOut.writeUTF(s);

... get bytes as before ...
```

Load Phase



Cost-Based Estimation

Opportunity 1: Which index do I use?
(What's the most selective predicate)

Opportunity 2: Which join order do I use?
If you get this right... Oracle/MS/Google has a job for you.
(Which order creates the fewest intermediate tuples)

Cost-Based Estimation

Opportunity 1: Which index do I use?
(What's the most selective predicate)

of distinct values

Upper/Lower Bounds

Histograms