CSE4/562 Database Systems

Practicum Component

02/21/2018
Recap

Parsed Query

Data

Results

.sql
Relational Algebra Trees

SELECT O.FirstName
FROM Officers O, Ships S
WHERE O.Ship = S.ID
AND S.Name = 'Enterprise'

\[ \pi_{\text{FirstName}}(O \bowtie \sigma \text{Name} = 'Enterprise' (Ships)) \]
Relational Algebra Trees

\[ \pi_{\text{FirstName}}(\text{Officers} \bowtie \text{Ship=ID}(\sigma_{\text{Name}='Enterprise'}\text{Ships})) \]
Statement statement = parser.Statement();

if (statement instanceof Select) {
    Algebra raTree = parseTree((Select)statement);
    evaluate(raTree);
}

else if (statement instanceof CreateTable) {
    loadTableSchema((CreateTable)statement);
}


SELECT [DISTINCT]
    target
FROM source
WHERE cond1
GROUP BY ...
HAVING cond2
ORDER BY order
LIMIT lim
UNION nextselect
SQL to RA

```
SELECT [DISTINCT] target
FROM source
WHERE cond1
GROUP BY ...
HAVING cond2
ORDER BY order
LIMIT lim
UNION nextselect
```

```
U

lim
nextselect

distinct

order by

target (π)

cond2 (σ)

agg

cond1 (σ)

source (X,嗥)
```
Project Outline

SQL Query ➔ Parser & Translator ➔ Relational Algebra (Query Plans)

Checkpoint 2
Real SQL

Optimizer ➔ Statistics

Execution Plan ➔ Evaluation Engine ➔ Query Result

Real SQL ➔ Statistics ➔ Optimizer ➔ Execution Plan
Checkpoint 2

- How do you join multiple tables, efficiently?
- How do you create a query plan?
- How do you deal with nested queries?
- Can you sort data? Just choose top-k rows?
The Evaluation Pipeline

Parsed Query

Results

Data

Checkpoint 2 focus

$\pi$ $\sigma$

Employee Department

.sql
What happens if I have a FROM-nested query?
FROM Clause

FROM R, (SELECT ...) S, T, ...

Selects are just relations!
FROM Clause

FROM R JOIN S ON cond
FROM Clause

FROM R JOIN S ON cond
FROM Clause

FROM R NATURAL JOIN S
FROM Clause

FROM R NATURAL JOIN S

cond = \text{schema}(R) \cap \text{schema}(S)

You need to be able to compute the schema of a RA operator
WHERE Clause - Join

WHERE R.a = S.b AND S.c = T.d

What happens if I have a nested query in where clause?
WHERE Clause - Join

WHERE R.a = S.b AND S.c = (SELECT ..)

\[ R \times S \\\xrightarrow{\text{SELECT }} .. \]

18
Creating Joins

FROM R, S, T WHERE R.a = S.b AND S.c = T.d
Creating Joins

FROM R, S, T WHERE R.a = S.b AND S.c = T.d

\[
\begin{align*}
&\sigma_{R.a = S.b} \\
&\times \\
&\sigma_{S.c = T.d} \\
&\times \\
&R \\
&S
\end{align*}
\]

\[
\begin{align*}
&T \\
&\times
\end{align*}
\]

\[
\begin{align*}
&R \\
&S
\end{align*}
\]

\[\Rightarrow\]

\[
\begin{align*}
&T \\
&\times
\end{align*}
\]

\[
\begin{align*}
&R \\
&S
\end{align*}
\]
SELECT Name, GamesPlayed
FROM Players
ORDER BY GamesPlayed

SELECT Name, GamesPlayed
FROM Players
ORDER BY GamesPlayed DESC
GetNext()

**Order By**
Read Each Tuple From Child
\[\rightarrow\]
Collections.sort()
(for now)
\[\rightarrow\]
**Return Tuple**
one by one
LIMIT

Ascending or Descending

SELECT Name, GamesPlayed
FROM Players
ORDER BY GamesPlayed
LIMIT 5
Implementing: Joins

**Solution 1 (Nested-Loop)**

For Each (a in A) { For Each (b in B) { emit (a, b); }}
Implementing: Joins

Solution 2 (Block-Nested-Loop)
Implementing: Joins

Solution 2 (Block-Nested-Loop)

1) Partition into Blocks
2) NLJ on each pair of blocks