Tutorial 6 — Query Optimization, Planning, Evaluation

Richard Wong
rk2wong@edu.uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

March 5, 2018

Give instances of relations R and S that show that the following pairs of relational algebra expressions are not equivalent:

1. $\pi_A(R - S)$ and $\pi_A(R) - \pi_A(S)$
2. $\sigma_\theta(R \leftouterjoin S)$ and $R \leftouterjoin \sigma_\theta(S)$, where θ uses only attributes of S

Exercise 6-1 Solution (1/2)

We are showing that $\pi_A(R - S)$ and $\pi_A(R) - \pi_A(S)$ are not equivalent.
Let our schemas be $R(A, B)$ and $S(A, B)$.
Let $R = \{(1, 2)\}, S = \{(1, 3)\}$.

$$\begin{aligned}
LHS &= \pi_A(R - S) \\
&= \pi_A(\{(1, 2)\} - \{(1, 3)\}) \\
&= \pi_A(\{(1, 2)\}) \\
&= \{(1)\} \\
RHS &= \pi_A(R) - \pi_A(S) \\
&= \pi_A(\{(1, 2)\}) - \pi_A(\{(1, 3)\}) \\
&= \{(1)\} - \{(1)\} \\
&= \emptyset \\
LHS &\neq RHS
\end{aligned}$$

Exercise 6-1 Solution (2/2)

We are showing that $\sigma_\theta(R ⟕ S)$ and $R ⟕ \sigma_\theta(S)$ are not equivalent when θ uses only attributes of S.
Let our schemas be $R(A, B)$ and $S(A, C)$.
Let $R = \{(1, 2)\}, S = \{(42, 1337)\}$.
Let θ be a predicate like $C = 1$, or anything that satisifies no elements in C.

$$
\begin{aligned}
LHS &= \sigma_\theta(R ⟕ S) \\
&= \sigma_{C=1}(\{(1,2)\} ⟕ \{(42, 1337)\}) \\
&= \sigma_{C=1}(\emptyset) \\
&= \emptyset \\
RHS &= R ⟕ \sigma_\theta(S) \\
&= \{(1,2)\} ⟕ \sigma_{C=1}(\{(42, 1337)\}) \\
&= \{(1,2)\} ⟕ \emptyset \\
&= \{(1, 2, null)\} \\
LHS &\neq RHS
\end{aligned}
$$

Exercise 6-2

Consider relations $R(A, B, C), S(C, D, E), T(E, F)$, where A, C, and E are their respective primary keys.
Suppose $n_R = 1000, n_S = 1500, n_T = 500$.

1. What is the tightest upper bound we can place on $n_{R \bowtie S \bowtie T}$?
2. How could we compute the join efficiently?

Exercise 6-2 Solution

$n_R = 1000, n_S = 1500, n_T = 500.$
Note that the size of the fully-joined relation ($n_{R \bowtie S \bowtie T}$) will be the same no matter the order we execute the joins in, since natural joins are associative and commutative.
Suppose we consider the execution order $((R \bowtie S) \bowtie T)$.

$$
\begin{aligned}
n_{R \bowtie S} &\leq n_R && \text{since C is a key of S} \\
n_{(R \bowtie S) \bowtie T} &\leq n_{R \bowtie S} && \text{since E is a key of T} \\
n_{R \bowtie S \bowtie T} &= n_{(R \bowtie S) \bowtie T} && \text{by associativity} \\
&\leq n_{R \bowtie S} \\
&\leq n_R \\
&= 1000
\end{aligned}
$$

So $n_{R \bowtie S \bowtie T} \leq 1000$.
To efficiently compute the join, it helps to have indices on the primary keys of S and T, and to use those indices to prevent linear scans of those relations during the join.

Exercise 6-3

Using the relational algebra equivalence rules, show how to derive the RHS expression from the LHS expression.

1. $\sigma_{\theta_1 \wedge \theta_2 \wedge \theta_3}(R) = \sigma_{\theta_1}(\sigma_{\theta_2}(\sigma_{\theta_3}(R)))$
2. $\sigma_{\theta_1 \wedge \theta_2}(R \bowtie_{\theta_3} S) = \sigma_{\theta_1}(R \bowtie_{\theta_3} \sigma_{\theta_2}(S))$, where θ_2 uses only attributes of S

Exercise 6-3 Solution (1/2)

$$\begin{aligned}
LHS &= \sigma_{\theta_1 \wedge \theta_2 \wedge \theta_3}(R) & \\
&= \sigma_{\theta_1 \wedge \theta_2}(\sigma_{\theta_3}(R)) & \text{by } \sigma\text{-cascade (rule 1)} \\
&= \sigma_{\theta_1}(\sigma_{\theta_2}(\sigma_{\theta_3}(R))) & \text{by } \sigma\text{-cascade (rule 1)} \\
&= RHS &
\end{aligned}$$

$$
\begin{aligned}
LHS &= \sigma_{\theta_1 \wedge \theta_2}(R \bowtie_{\theta_3} S) & \\
&= \sigma_{\theta_1}(\sigma_{\theta_2}(R \bowtie_{\theta_3} S) & \text{by } \sigma\text{-cascase (rule 1)} \\
&= \sigma_{\theta_1}(R \bowtie_{\theta_3} \sigma_{\theta_2}(S)) & \text{since } \theta_2 \text{ only uses attributes of } S, \\
&= RHS & \text{we can distribute } \sigma_{\theta_2} \text{ over } \bowtie_{\theta_3}
\end{aligned}
$$

Let R be our relation with n_r records.
Suppose s_i records in R match a predicate θ_i: that is, $\sigma_{\theta_i}(R) = s_i$.
The *selectivity* of θ_i, $sel_{\theta_i}(R)$ is defined to be $\frac{s_i}{R}$. This represents the probability that a record in R satisifies θ_i.

Derive the selectivity formulas for the following complex selections:

1. conjunction: $\sigma_{\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_m}(R)$
2. negation: $\sigma_{\neg\theta}(R)$
3. disjunction: $\sigma_{\theta_1 \vee \theta_2 \vee \ldots \vee \theta_m}(R)$

Exercise 6-4 Solution

We make the simplifying assumption that predicates are independent of one another, allowing us to use standard probability rules for dealing with independent events.

1. $sel_{\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_m}(R) = \prod_{i=1}^{m} sel_{\theta_i}(R)$
2. $sel_{\neg\theta}(R) = 1 - sel_{\theta_i}(R)$
3. $sel_{\theta_1 \vee \theta_2 \vee \ldots \vee \theta_m}(R) = 1 - sel_{\neg\theta_1 \vee \theta_2 \vee \ldots \vee \theta_m}(R)$
   $= 1 - \prod_{i=1}^{m} sel_{\neg\theta_i}(R)$
   $= 1 - \prod_{i=1}^{m} 1 - sel_{\theta_i}(R)$

Exercise 6-5

What are some strategies that a query optimizer could use to reduce the cost of query plan selection, or the cost of the query itself?

Exercise 6-5 Solution

- limit the size of intermediate results early on in the plan (select and project ASAP)
- cache subplans
- materialize commonly used views which result from expensive queries
- remove unnecessary joins
- reinterpret subqueries as joins
- pipeline where possible
- and more...