Uni Freiburg, Web Science Group
Prof. Peter Fischer
Systems Infrastructure for Data Science - Winter 2012/13

Exercise Sheet #5: Query Optimization

March 30, 2011

Exercise 5.1 : From Queries to Execution Plans

Consider the following partial schema of a database which keeps track of students and the courses they take:

$Student(\underline{matrikel\_no}, first\_name, last\_name)$
$Class(\underline{class\_id}, class\_name)$
$Takes(\underline{class\_id}, \underline{matrikel\_no})$

Primary keys are underlined. You may assume the obvious foreign-key relations and indexes on primary and foreign key columns. The table cardinalities are as follows:

$|Student| = 1,000$
$|Class| = 100$
$|Takes| = 5,000$

A. Express the following natural-language query in SQL: Give me the last names of all students who take the course called "Information Systems".

B. Convert the SQL statement to a straight-forward query-plan using only cartesian products, not joins. Estimate the cardinality of each intermediate result and the final result.

C. Push-down the selection operators below or into the cartesian products (turning them into joins). Estimate the cardinalities of each intermediate result using your knowledge of table cardinalities and foreign-key relations.

D. Choose access paths and physical implementations of the various operators (particularly joins), taking advantage of indexes or sort order. Try to find a near-optimal plan. Revise your join order if necessary.

Exercise 5.2 : Query Planning

Consider the database containing the following tables:

$Courses(\underline{ID}, Name, Room, Time)$
$Exercises(\underline{ID}, C\_ID, A\_ID, Room, Time)$
$Assistants(\underline{ID}, Firstname, Lastname)$

and the query:

```
SELECT C.Name, A.Firstname, A.Lastname, E.Room, E.Time
FROM Courses C, Assistants A, Exercises E
WHERE C.ID = E.C_ID AND A.ID = E.A_ID AND C.Room like '10%' AND E.Room not like 'CAB%';
```

A. Show the logical canonical form for this query expressed in relational algebra. The logical canonical form is the straight-forward translation of the SQL query into a relational algebra query plan containing only selection, projection and cross product operators.

B. Perform Query Rewrite on part [A]. Which rules would you apply to get to this intermediate representation? Show the logical query plan of the resulting query.

C. Assume that there is a non-clustered index on $C.Room$ and a clustered, direct index on $E.A\_ID$. Based on part [B], show possible physical query execution plans (i.e., access paths, join implementations) that take advantage of the indexes. Discuss the efficiency of your plans based on the following cost metrics: disk I/O, intermediate result size.

Exercise 5.3 : Optimization using Indexes

Consider the following scenario:

$Emp(\underline{eid}, sal, agel, did)$
$Dept(\underline{did}, \underline{projid}, budget, status)$
$Proj(\underline{projid}, code, report)$

Assume that each Emp record is 20 bytes long, each $Dept$ record is 40 bytes long, and each $Proj$ record is 2000 bytes long on average. There are 20,000 tuples in Emp, 5000 tuples in $Dept$ (note that did is not a key), and 1000 tuples in $Proj$. Each department, identified by did, has 10 projects on average. The file system supports 4000 byte pages, and 12 buffer pages are available. You can assume uniform distribution of values. (The cost metric to use is the number of page I/Os. Ignore the cost of writing out the final result.) Consider the following query:

```
SELECT D.did, COUNT(*)
FROM Dept D, Proj P
WHERE D.projid=P.projid
GROUP BY D.did
```

A. Suppose that no indexes are available. Show the plan with the lowest estimated cost.

B. If there is a hash index on $Proj.projid$ what is the plan with lowest estimated cost?

C. If there is a hash index on $Dept.projid$ what is the plan with lowest estimated cost?

D. Suppose that there is a clustered B^+ tree index on $Dept.did$ and a hash index on $Proj.projid$. Show the plan with the lowest estimated cost.

E. Suppose that there is a clustered B^+ tree index on $Dept.did$, a hash index on $Dept.projid$, and a hash index on $Proj.projid$. Show the plan with the lowest estimated cost.