CS W186 - Spring 2024
Exam Prep Section 7
Query Optimization

Query Optimization 1

(Modified from Fall 2017)

For the following question, assume the following:

- Column values are uniformly distributed and independent from one another
- Use System R defaults (1/10) when selectivity estimation is not possible
- Primary key IDs are sequential, starting from 1
- Our optimizer does not consider interesting orders

We have the following schema:

Table Schema	Records	Pages	Indices
CREATE TABLE Student (sid INTEGER PRIMARY KEY, name VARCHAR(32), major VARCHAR(64), semesters_completed INTEGER);	25,000	500	+ Index 1: Clustered (major). There are 130 unique majors + Index 2: Unclustered (semesters completed). There are 11 unique values in the range [0, 10].
CREATE TABLE Application (sid INTEGER REFERENCES Student, cid INTEGER REFERENCES Company, status TEXT, (sid, cid) PRIMARY KEY);	100,000	10,000	+ Index 3: Clustered(cid, sid). + Given: status has 10 unique values
CREATE TABLE Company (cid INTEGER PRIMARY KEY, open_roles INTEGER);	500	100	+ Index 4: Unclustered(cid) + Index 5: Clustered(open roles). There are 500 unique values in the range [1, 500]

Consider the following query:

```
SELECT Student.name, Company.open_roles, Application.referral
FROM Student, Application, Company
WHERE Student.sid = Application.sid                        -- (Selectivity 1)
AND Application.cid = Company.cid                          -- (Selectivity 2)
AND Student.semesters_completed > 6                        -- (Selectivity 3)
AND (Student.major='EECS' OR Company.open_roles <= 50)     -- (Selectivity 4)
AND NOT Application.status = 'limbo'                       -- (Selectivity 5)
ORDER BY Company.open_roles;
```

1. For the following questions, calculate the selectivity of each of the labeled Selectivities above.

   (a) Selectivity 1

   (b) Selectivity 2

   (c) Selectivity 3

(d) Selectivity 4

(e) Selectivity 5

2. For each predicate, which is the first pass of Selinger's algorithm that uses its selectivity to estimate output size? (Pass 1, 2 or 3?)

(a) Selectivity 1

(b) Selectivity 2

(c) Selectivity 3

(d) Selectivity 4

(e) Selectivity 5

3. Mark the choices for all access plans that would be considered in pass 2 of the Selinger algorithm.

(a) Student ⋈/ Application (800 IOs)

(b) Application ⋈ Student (750 IOs)

(c) Student ⋈ Company (470 IOs)

(d) Company ⋈ Student (525 IOs)

(e) Application ⋈ Company (600 IOs)

(f) Company ⋈ Application (575 IOs)

4. Which choices from the previous question for all access plans would be chosen at the end of pass 2 of the Selinger algorithm?

5. Which plans would be considered in pass 3?

(a) Company ⋈ (Application ⋈ Student) (175,000 IOs)

(b) Company ⋈ (Student ⋈ Application) (150,000 IOs)

(c) Application ⋈/ (Company ⋈ Student) (155,000 IOs)

(d) Application ⋈ (Company ⋈ Student) (160,000 IOs)

(e) Student ⋈ (Company ⋈ Application) (215,000 IOs)

(f) (Company ⋈ Application) ⋈ Student (180,000 IOs)

(g) (Application ⋈ Company) ⋈ Student (200,000 IOs)

(h) (Application ⋈ Student) ⋈ Company (194,000 IOs)

(i) (Student ⋈ Application) ⋈ Company (195,000 IOs)

(j) (Student ⋈ Company) ⋈ Application (165,000 IOs)

6. Which choice from the previous question for all plans would be chosen at the end of pass 3?

Query Optimization 2

(Modified from Spring 2016)

1. True or False

   • When evaluating potential query plans, the set of left deep join plans are always guaranteed to contain the best plan.

   • As a heuristic, the System R optimizer avoids cross-products if possible.

   • A plan can result in an interesting order if it involves a sort-merge join.

   • The System R algorithm is greedy because for each pass, it only keeps the lowest cost plan for each combination of tables.

2. For the following parts assume the following:

   • The System R assumptions about uniformity and independence from lecture hold • Primary key IDs are sequential, starting from 1

   We have the following schema:

CREATE TABLE Flight (fid INTEGER PRIMARY KEY, from_id INTEGER REFERENCES City, to_id INTEGER REFERENCES City, aid INTEGER REFERENCES Airline)	NTuples: 100K, NPages: 50 Index: (I) unclustered B+-tree on aid. 20 leaf pages. (II) clustered B+-tree on (from_cid, fid). 10 leaf pages.
CREATE TABLE City (cid INTEGER PRIMARY KEY, name VARCHAR(16), state VARCHAR(16), population INTEGER)	NTuples: 50K, NPages: 20 Index: (III) clustered B+-tree on population. 10 leaf pages. (IV) unclustered index on cid. 5 leaf pages. Statistics: state in [1, 50], population in [10^6, $8*10^6$]
CREATE TABLE Airline (aid INTEGER PRIMARY KEY, hq_cid INTEGER REFERENCES City, name VARCHAR(16))	NTuples: 5K, NPages: 2

   Consider the following query:

```
SELECT *
FROM Flight F, City C, Airline A
WHERE F.to_id = C.cid
AND F.aid = A.aid
AND F.aid >= 2500
AND C.population > 5e6
AND C.state = 'California';
```

   Considering each predicate in the WHERE clause separately, what is the selectivity for each?

   (a) C.state='California'

(b) F.to_id = C.cid

(c) F.aid >= 2500

(d) C.population > 5 * 10^6

3. For each blank in the System R DP table for Pass 1. Assume this is before the optimizer discards any rows it isn't interested in keeping and note that some blanks may be N/A. Additionally, assume B+ trees are height 2.

Table(s)	Plans	Interesting Orders from Plan (N/A if none)	Cost (I/Os)
Flight	Index (I)		
City	Filescan		
City	Index (III)		

4. After Pass 2, which of the following plans could be in the DP table?

(a) City [Index(III)] JOIN Airline [File scan]

(b) City [Index (III)] JOIN Flight [Index (I)]

(c) Flight [Index (II)] JOIN City [Index (III)]

5. Suppose we want to optimize for queries similar to the query above in part 2, which of the following suggestions could reduce I/O cost?

(a) Change Index (III) to be unclustered

(b) Store City as a sorted file on population