School of Computer Science, McGill University

COMP-421B Database Systems, Winter 2016

Written Assignment 3: Query Evaluation

March-31, 11:30pm

Assume a database maintaining patient information, samples and DNA Sequences (the information is taken from http://www.ncbi.nlm.nih.gov/books/NBK6828/). It contains information about patients. Each patient has samples taken. From each sample several DNA sequences are extracted.

`Patients(pid:CHAR(4), gender:CHAR(1), bday:DATE, city:VARCHAR(30), country:VARCHAR(30)`
e.g.: ('A1C1', 'F', '1980-01-21', 'Manaus', 'Brazil')

`Samples(sampleid:INT, organ:VARCHAR(20), collect:DATE, pid:CHAR(4), descr:VARCHAR(100)`
e.g.: (12346, 'liver','2015-05-21', 'A1C1' ,'taken from an area that was exposed....')

`Sequences(seqid:INT, region:CHAR(3), size:INT, seq:VARCHAR(1100), sampleid: INT)`
e.g., : (11111111, 'ENV', 432, 'AGCCAACATGA...CTCAGCTGAAGAAG', 12346)

INT has 4 Bytes, a char has 1 Byte, a DATE has 10 Bytes. On average, the values the VARCHAR attributes city, country, organ, desc are half the max. length of the attribute. The `size` attribute in `Sequences` indicates the length of the sequence (detailed in the `seq` attribute). It has values between 100 and 1100.

The relation `Patients` has around 1,000 tuples on 15 data pages. Patients came from 50 different countries.

The relation `Samples` has 10,000 tuples.

The relation `Sequences` has around 50,000 tuples that are stored on around 10,000 pages. There around 500 different genomic regions (`region` attribute).

You can assume all attribute values have uniform distribution and attributes are independent from each other.

The rids (record identifiers) have 10 Bytes. For any calculation, you have around 50 buffer pages.

Assume further there is an index on `region` of `Sequences`. There are 500 data entries (one for each possible value). On average each entry has 50,000 / 500 = 100 rids. Size of a data entry is thus (3 + 100 * 10) = 1003. With leaf pages on average full 75%, there are 3 entries per leaf page, and thus 500/3 = 167 leaf pages.

Exercise 1: Data Pages (5 Points)

How many data pages does the relation `Samples` require, assuming that data pages are filled up to around 75%.

Exercise 2: Query execution (50 Points)

1. (35 Points) A typical query tries to find all DNA sequences with a minimum length that are in a specific region.

```
SELECT seq
FROM  sequences
WHERE size > 700 AND region = 'POL'
```

   a.) Indicate the access costs (in number of pages retrieved leading to I/O) for this query if
      i. there is no index available
      ii. there is an unclustered index on `size` only and you use it for the query
      iii. there an unclustered index on `region` only and you use it for the query
      iv. there are unclustered indices on both `size` and `region` and you use both.

   b.) What if the condition is `size` > 200 or `size` > 1000. Keep your answer short.

   c.) Would a clustered index on either `size` or `region` increase performance? Give a short explanation.

2. (15 Points) Another typical query is

```
SELECT region, count(*)
FROM sequences
GROUP BY region
HAVING count(*) > 120
```

   a.) *Give an execution strategy that uses no index.*

   b.) *Give an execution strategy that uses the index on region*

   c.) *Assume now, that instead of* `count(*)`*, the projection is* `count(DISTINCT sampleid)`*. What kind of index would you create to speed up this type of query.*

Exercise 3: Joins (25 Points)

Now assume there is an index on `sampleid` on `Samples`. Choose inner and outer relations appropriately. You can always assume that the root node and any other inner index pages are in main memory.

1. Estimate the number of output tuples
2. Calculate the estimated I/O (a bit more than 50 buffer pages available)
   a.) index nested loop join between `Samples` and `Sequences`
   b.) block nested loop join between `Samples` and `Sequences`
   c.) sort merge join between `Samples` and `Sequences`
   d.) hash join between `Samples` and `Sequences`

Exercise 4: Optimization (20 Points)

Now look at the query

```
SELECT p.gender, s.organ, se.region
FROM patients p, samples s, sequences se
WHERE p.pid = s.pid
AND  s.sampleid = se.sampleid
AND se.Size > 800
AND p.country = 'Brazil'
```

A non-optimized relational expression for this query is

$\pi_{gender,organ,region}(\sigma_{size>800 \wedge country='Brazil'}((patients \times sequences) \bowtie samples))$

Perform an algebraic optimization of your expression according to the rules discussed in class and show the operator tree. Indicate the types of joins you would perform. Give an estimate of the number of tuples that flow from one operator to the next. These numbers, of course, depend on the selectivity of certain attributes. Make reasonable assumptions, and indicate your assumptions.