School of Computer Science

COMP9120 Relational Database Systems

Tutorial Week 10: Storage and Indexing

Exercise 1. Calculating space and time

Suppose we have a table Rel1(A, B, C). Each A field occupies 4 bytes, each B field occupies 12 bytes, each C field occupies 8 bytes. Rel1 contains 100,000 records. There are 100 different values for A represented in the database, 1000 different values for B, and 50,000 different values for C. Rel1 is stored with a primary B+ tree index on the composite key consisting of the pair of attributes (A,B); assume this index has 2 levels, including the root page and excluding the leaf (record) pages. Assume that in this database, each page is 4K bytes, of which 250 bytes are taken for header information. Record locations within the index use a 4-byte *rowid*. Assume that reading a page into memory takes 150 msec, and that the time needed for any query can be approximated by the time spent doing disk I/O.

1a) Calculate the space needed for the records/data of Rel1.
Hint: How much space is used by a single record? How many records per page? How many pages for the table? Then convert this back to space. [Note that each record is not split across pages, and each page has a header.]

1b) Calculate the time taken to perform a table scan (i.e., linear scan) through the table Rel1.
Hint: How many pages are needed to make up the space occupied by the table? How many pages will be read from disk during a table scan?

1c) Calculate the time taken, using the primary index, to execute the following query. Assume the selectivity of range condition on B is 10%.

```
SELECT C
FROM Rel1
WHERE A = 'AQG' AND (B BETWEEN 'WPQ' AND 'XYZ');
```

Exercise 2. Index creation with PostgreSQL

Find which indices already exist for the tables you defined in early tutorials and assignments.
Identify an extra index, which would be useful for one of the queries you wrote (supposing that the database became much bigger, with many more records in each table!). Create this index.
Note: To check which indexes are present in your schema (plus some details), you can use the following SQL commands in PostgreSQL:

```
SELECT * FROM PG_INDEXES;
```