Indexing and Views – 3/6

Several important announcements regarding the upcoming midterm, see slides for details.

Reading Quiz:

- Q1: b) CREATE INDEX index_name ON table_name (column_name)
- Q2: d) All of the above
- Q3: d) Space saving: to reduce the storage of database tables
- Q4: c) SELECT
- Q5: a) True

B+ trees (often just called B trees) are the ubiquitous indexing structure
Start at the root, follow the links based on the search key (not to be confused with primary and foreign key)
Leaf nodes have pointers to the rows in the associated table and to their successor leaf node in the B tree

B trees can be on single columns or multiple columns. A multi-column B tree (aka a composite index) concatenates the values in the indexed columns in the order specified by the create index statement. For example, an index on (eventid, qtysold) first sorts by eventid, and then sorts by qtysort.

B trees are great for doing exact match searches and also doing range queries as long as the index is selective enough (if the range comprises the majority of the records in the table, the query optimizer will choose to do a full table scan to avoid disk seeks)
Query optimizer evaluates the cost of doing a full table scan versus an index scan for a given query.
Chooses lowest access path based on the cost
Predominant cost of querying is Read I/O

Covered Index - when the index that contains all the columns you need to answer your query. For example the index on (eventid, qtysold) on the Sales table is a covered index for the query "select sum(qtysold) from Sales where eventid = 800" because both the eventid and qtysold values are in the index and therefore the query can be answered just by looking at the index.

To see what access path chosen by the query optimizer (e.g. index scan or full table scan), look at the query plan as follows:
"explain" follow by the statement
For example, “EXPLAIN SELECT * FROM Actors WHERE lname = 'Amanda'"

Practice Problem 1
A) False, the composite index with columns in the order of eventid then qtysold will not speeds up the query as qtysold is not the first argument.

Practice Problem 2
B) False, the query would be slowed down by the index because neither of the columns in the index helps with the retrieval process.

A View can be used to encapsulate data. Virtual views are computed on-demand, always up-to-date with the base table(s). Materialized views are pre-computed, require more storage, and may be out-of-date with the base table(s). Materialized views are often used to pre-compute joins over large tables, virtual views are often used to hide sensitive columns and/or records and simplify the SQL for complex queries.

Practice Problem 3

B) Event and Category, because that is the common columns involved in those queries.