Lecture 13A (April 9, 2018)

CS 410 Spring 2018

Previously

- Homeworks
- Transactions

Looking for Data

Example data: HCIBIB

- Started talking about them on Wednesday
- Data is in a big pile - a column of tuples
    - Pointers to other storage (BLOB, etc.)
- How to find things?
    - Scan rows
    - Look up quickly

Library: we want to find books by Cathy O'Neil. But the shelves are organized by topic.

Indexing to the Rescue

Index is a separate storage that stores organized by 'key'. MySQL calls indexes 'keys' sometimes.

It is a map from key values to (lists of) record positions. Like a hashtable.

Hashtables are one way to implement indexes! A massive hashtable stored on disk.

Another common storage mechanism: the B+ tree

- Nodes have up to n keys
- Child nodes are between keys
- Final nodes point to leaf data

B+ tree advantages:

- In order, so order-by and range queries work
- Can support efficient joining - linear scan of two indices in parallel

Other index types

- Full-text indexes - store words in a hash table or b+-tree
- Spatial indexes - R-tree

What to index?

- Things you look up often
- Things with high selectivity (each value only pulls up a smallish fraction of records)
- Unique columns (this is how uniqueness constraints work)
- Foreign keys! (MySQL does this automatically)
- Direct lookup is good for hash tables
- In-order scans (including possibly joins, depending on your database engine) and range queries are good for B+ trees

Updating indexes makes inserts and updates slower, and indexes take disk space, so we don't just want to index everything.

Creating Indexes

- The INDEX clause in CREATE TABLE
- CREATE INDEX author_name_idx ON author (author_name);

Indexing and Bulk Import

When you are importing a lot of data, indexing can make things slower. It is faster to build an index on a bunch of data than to incremental update one.

So, when importing a lot of data:

- Create tables with minimal indexes and FK constraints
- Import data
- Add FK constraints (with ALTER TABLE) and indexes (with CREATE INDEX)