Lecture 26:

- Query plans (wrap up)
- Open Data Formats (intro)

Announcements:

- Project out
- HW-4 out
- R-4 out

Query Plans

Putting it all together for Select-Project-Join Queries

- given a query tree, generate and estimate different query plans
- select and project often “on the fly” (in buffer) when many joins needed

Left-deep query trees  ...  i.e., fully pipelined trees

- helps restrict search space for plans
- left-deep join trees generate “fully pipelined plans” (pull a row-at-a-time)
- intermediate results don’t need to be written to / read from temp files
- can use indexes, different operator placements (e.g., push selects)
Open Data Formats

Basics:

Open versus Proprietary

On-disk storage vs in-memory representation

Designed for efficiency (compression, encoding/decoding)

Interoperability w/out “Copy & Convert”

• to get data from one system to another
• convert between value representations (like integers, etc.)
• convert between how rows are layed out

Some Examples: Handling structured data within Hadoop(-like) ecosystems

Apache ORC & Parquet: On-disk formats (circa 2013)

• ORC originally by Meta, Parquet by Twitter & Cloudera (one-month apart)
• Both use columnar storage (more later)

Apache Avro: On-disk format, created by Hadoop (circa 2009)

• Similar to protocol buffers (for RPCs / serialization), supports JSON

Apache Arrow: In-memory format (circa 2015) by various (including Pandas)

• avoids having to serialize/deserialize between systems
• e.g., query result followed by post processing

★ Parquet & ORC have become de facto standards in data warehouses/lakes
Open Data Formats

Storage Models:

Row-oriented (one row after another) ... e.g., slotted pages

• also called the *N-ary storage model*

Column-oriented (one column after another) ... columnar

• also called the *decomposition storage model*

• a challenge is to “get back” the rows!

Hybrid: “batches” (i.e., groups) of columns ... also columnar

• partition table horizontally into row groups

• store tuples column-by-column within each group