DWDC

SQL: the Sequel

More SQL in the database, and using SQL in data science contexts

Ryan B. Harvey

August 6, 2014

DATAWRANGLERSDC

Review: Relational Data

DWDC

- Relational data is organized in tables consisting of columns and rows
- Fields (columns) consist of a column name and data type constraint
- Records (rows) in a table have a common field (column) structure and order
- Records (rows) are linked across tables by key fields

Relational Data Model: Codd, Edgar F. "A Relational Model of Data for Large Shared Data Banks" (1970)

Review: Why should I use a database system?

DWDC

1. You care about strong data types, type validation and data access controls
2. You need to relate multiple tables together via common fields
3. Your data is larger than a few 10s to 100 MB, making file parsing onerous
4. You need to subset or aggregate your data often based on field values

The above are my opinions based on experience. Others may disagree, and that's OK.

Review: Intro to SQL

DWDC

- SQL ("Structured Query Language") is a declarative data definition and query language for relational data
- SQL is an ISO/IEC standard with many implementations in common database management systems (a few below)

Structured Query Language: ISO/IEC 9075 (standard), first appeared 1974, current version SQL:2011

Review: Which database system should I use?

DWDC

1. Use the one your data is in
2. Unless you need specific things (performance, functions, etc.), use the one you know best
3. If you need other stuff or you've never used a database before:
   A. SQLite: FOSS, one file DB, easy/limited
   B. PostgreSQL: FOSS, Enterprise-ready

The above are my opinions based on experience. Others may disagree, and that's OK.

SQL: Working with Objects

DWDC

- Data Definition Language (DB Objects)
  - CREATE (table, index, view, function, ...)
  - ALTER (table, index, view, function, ...)
  - DROP (table, index, view, function, ...)

SQL: Working with Rows

DWDB

- Query Language (Records)
  - SELECT ... FROM ...
  - INSERT INTO ...
  - UPDATE ... SET ...
  - DELETE FROM ...

SQL: SELECT Statement

DWDC

- SELECT <col_list> FROM <table> ...
  - Merging: JOIN clause
  - Row binding: UNION clause
  - Filtering: WHERE clause
  - Aggregation: GROUP BY clause
  - Aggregated filtering: HAVING clause
  - Sorting: ORDER BY clause

You'll remember this from last time

SQL: Views from SELECTs

DWDC

- CREATE VIEW <name> AS ...
- SELECT <col_list> FROM <table> ...
  - Merging: JOIN clause
  - Row binding: UNION clause
  - Filtering: WHERE clause
  - Aggregation: GROUP BY clause
  - Aggregated filtering: HAVING clause
  - Sorting: ORDER BY clause

Same as before!

SQL: Functions from Views

DWDC

- CREATE FUNCTION <name> (<params>) AS ...
- SELECT ... <params> ...
  - Merging: JOIN clause
  - Row binding: UNION clause
  - Filtering: WHERE clause
  - Aggregation: GROUP BY clause
  - Aggregated filtering: HAVING clause
  - Sorting: ORDER BY clause

Almost same as before!

SQL: Tuning with EXPLAIN

DWDC

- EXPLAIN <options> SELECT ...
  - rows scanned: COST option
  - wordy response: VERBOSE option
  - output formatting: FORMAT option
  - actually run it: ANALYZE option
  - runtime (only with ANALYZE): TIMING option
- (EXPLAIN is not part of the SQL standard)

Same as before!

SQL: Tuning using Indexes

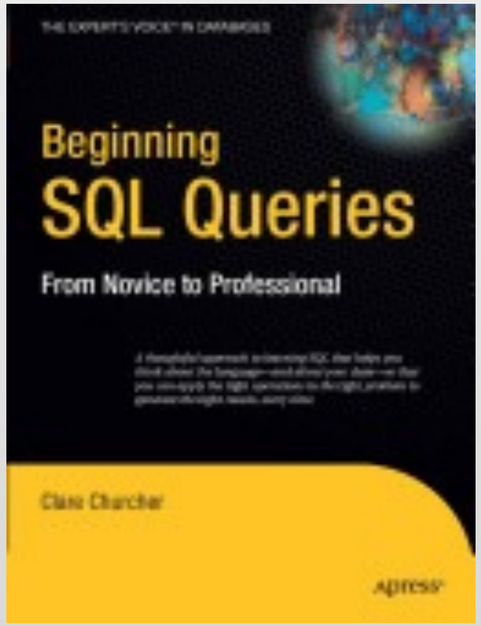
DWDC

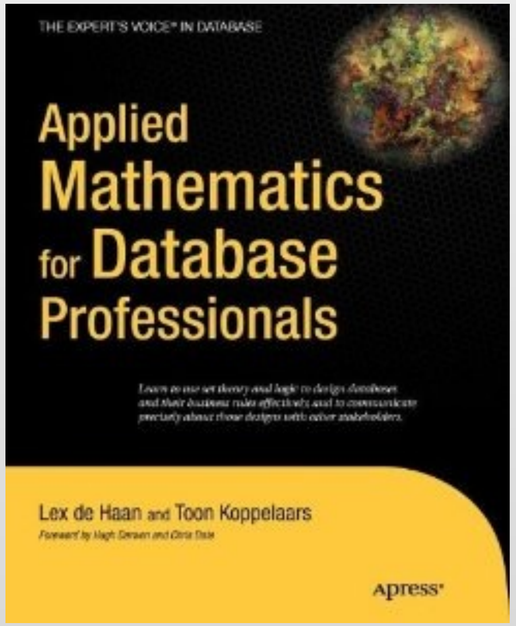
- CREATE INDEX <name> ON <table> (<col_list|expression>) ...
  - UNIQUE indices for key fields
  - Use functions in expressions: LOWER(<text_col>), INT(<num_col>)
  - Specify ordering (ASC, DESC, NULLS FIRST, etc.) and method (BTREE, HASH, GIST, etc.)
  - Partial indexes via WHERE clause

What's in your WHERE clause?

SQL Beginner Resources

- Basic SQL Commands Reference: http://www.cs.utexas.edu/~mitra/csFall2013/cs329/lectures/sql.html





Same as before!

Still useful!

Intro to Relational Algebra

TEXT

- Basic operators

SELECT	σ	WHERE, HAVING
PROJECT	$\prod$	<COL_LIST>
RENAME	ρ	AS

- Join operators: inner/outer, cartesian
- Set operators: union, intersect, set minus, and, or, etc.
- SELECT name, id FROM t1 WHERE id<3 AND dob<DATE '2004-01-01'

$$\Pi_{NAME,ID}\ \sigma_{ID<3 \wedge DOB<(1/1/2004)}\ (T1)$$

For a very detailed Intro to Relational Algebra, see lecture notes from 2005 databases course, IT U of Copenhagen

SQL in other languages

(or, accessing data in databases via SQL in other languages)

DWDC

- R with libraries
  - RPostgreSQL, dplyr
- Python with modules
  - psycopg2, SQLAlchemy
- Julia with packages (in dev)
  - PostgreSQL, DBI

You'll remember this from last time

SQL in other languages

(or, operating on other languages' data structures via SQL)

- R with libraries
  - RSQLite, sqldf
- Python with modules
  - Pandas, PandaSQL
- Julia with packages (in dev)
  - SQLite, DBI

Mostly, Data Frames.

DWDC

Now, let's look at

<u>some code</u>!

Ryan B. Harvey

http://datascientist.guru
ryan.b.harvey@gmail.com
@nihonjinrxs
+ryan.b.harvey

DWDC

Day Job

IT Project Manager
Office of Management and Budget
Executive Office of the President

Side Job

Data Scientist & Software Architect
Kitchology Inc.