

Data Management

OR
GETTING PUNCHED IN THE FACE BY SQL AND PANDAS



BY
NICHOLAS ROSS, PHD

PROFESSOR OF DATA SCIENCE
UNIVERSITY OF CHICAGO

© 2020-2023 All Rights Reserved

Cover Art By: Megan Carlsen

Version: 2023-08-21 02:33:21

Contents

Introduction and Errata

i

Relational Databases

1	Rows and Columns	1
1	What is a Relational Database	3
2	Selecting Columns	6
3	WHERE: Filtering rows	7
4	Null	9
5	ORDER BY and LIMIT	11
6	Column Numbering	16
7	Where are we: A Note on Scope	17
2	Basic Manipulations	19
1	Types	21
2	Renaming a Column	23
3	Basic Mathematical Manipulations, ABS and LEAST/GREATEST	24
4	Queries without a FROM Clause and Singletons	28
5	String Functions: LEFT, RIGHT, LOWER, UPPER, LENGTH, TRIM and CONCAT	29
6	ROUND and Changing Types (CAST)	33
7	CAST and changing types	33
3	Subqueries, Distinct & Case	41
1	Query Evaluation Order: SELECT and WHERE	43
2	Comparisons: BETWEEN, LIKE and ILIKE	45
3	CASE: Conditional Logic	47
4	The DISTINCT Operator	52
5	Subqueries (IN, ANY, ALL)	55
6	Correlated Subqueries	58
4	Database Internals: Transactions	63
1	REDO / COMBINE NEXT SECTIONS	65
2	Table Creation and Deletion	65
3	Database Operations: CRUD	65
4	Creating Tables, Constraints and Deleting tables	66
5	Altering Tables	68
6	Inserting, Copying, Updating and Deleting	68
7	Transactions and ACID	69
8	Isolation Levels in Relational Databases	73
9	Why do we care (NoSQL)?	78

10	NoSQL	80
11	Transaction Implementations [TBD]	81
5	Aggregations	83
1	Introduction to MTA data set	85
2	GROUP BY clause	86
3	Column numbering syntax	91
4	Aggregates and CASE Statements	93
5	Named Subqueries	95
6	Dates and Types	101
1	Date Types	103
2	Date Functions	104
3	Hard GROUP BY problems	110
7	Averages	115
1	The Trouble with Averages	117
2	HAVING	119
3	COALESCE and NVL	120
8	Joins	123
1	Joins	125
2	UNION and UNION ALL	132
3	Best Practices when Combining Tables	134
4	Intermediate Joins	136
4.1	Aggregations on-self	136
4.2	Cross Joins for missing values	137
5	Statistical Analysis in SQL	138
9	Advanced Joins	147
1	The Shape of Data	149
2	Revenue over time & Advanced Joins	151
2.1	First Value	152
2.2	Most common value by group	156
2.3	Cumulative Sum	158
2.4	Rolling 90 day Calculation	160
2.5	Cohorted Monthly Revenue	161
10	Analytic Functions & CTE's	165
1	Analytic Functions	167
2	Using Analytic Functions with Transaction Data	174
3	Common Table Expressions ("CTE")	176
4	CTEs with the transaction data	178
11	Database Internals: Performance Evaluation	181
1	Normalization	183
2	Views	185
3	Information Schema	189
4	Performance Considerations	190
5	Index	194
6	Distributed Systems and the CAP Theorem	195

12 Extensions [TBD]	199
1 More Advanced Joins	204
2 OLAP: Cube and Rollup	210
3 Schemas	210
4 Keys	210
5 Data Exploration Strategies	210
6 Query Strategies	210
13 Interview Hints	211
1 Interview Hints	212
2 Example Interview #1	215
3 Example Interview #2	217
4 Example Interview #3	219
5 Example Interview #4	220
 Pandas	
14 Introduction	223
1 What is Pandas	225
2 Data structures	226
3 Selecting Columns and Rows	231
4 Column Types Conversion	237
5 Dealing with NaN	237
6 Choosing the largest and smallest values	239
7 Manipulating Data & Method Chaining	240
8 Indexes: Creating and Dropping	244
9 Views and Copies	246
15 More Manipulations and Types	251
1 Sorting DataFrames	253
2 Dealing with Duplicates	256
3 Using Type specific functions	258
3.1 Dates	258
3.2 Strings	260
4 CASE style statements and the “isin” operator	264
5 Regex Pattern Matching	265
16 Aggregations	269
1 Introduction to the MTA dataset	271
2 Simple Aggregations	271
3 GroupBy Objects	274
4 Advanced Index / Multiindex	279
5 If not indexes...	285
6 Indexing with aggregations, a big Gotcha	286
17 Joins	289
1 Helpful Table / Review	291
2 Merging data in Pandas	292
3 Complex Join Conditions	294
4 Stacking Data	294

5	Lags and Leads	296
6	Apply, map and applymap: Advanced Transformations	297
18	Window Functions	301
1	Window Functions in Pandas	303
2	Some gotchas	307
3	Reshaping Data: Transpose, Stack and Unstack	308
4	A Bunch of stuff to clean up	312
5	Combining with the original DataFrame	312
6	Moving the Window	316
7	Pivot / Melt	316
 Appendix		
Appendix A	Data Dictionaries	317
1	Introduction	318
2	Iowa Fleet data	318
3	NY MTA Data	319
4	Daily Stock Data: s2010 and s2011	321
5	Annual Fundamental Financial information: fnd	322
6	Soap Transaction Data	325
Appendix B	Connecting SQL to Python or R	327
1	Connecting to any database: ODBC and JDBC	327
2	Connecting only to PostgreSQL	327
Appendix C	Assignments	329
1	HW #0A: PostgreSQL Installation	330
2	HW #0B: Pandas Installation	331
3	HW #0C: MS CAPP Installation instructions	332
4	HW #1A: Basic SQL Querying	333
5	HW #1B: Basic Pandas	335
6	HW #2A: Basic Functions	337
7	HW #3A: Subqueries	339
8	HW #3B: Subqueries in Pandas	342
9	HW #4A: Aggregation	345
10	HW #4B: Aggregation in Pandas	347
11	HW #5A: Aggregate Functions and Dates	349
12	HW #5B: Aggregate Functions and Dates	351
13	HW #6A: SQL Joins (I)	353
14	HW #6B: Pandas Joins (I)	356
15	HW #7A: SQL Joins (II)	358
16	HW #7B: Pandas Joins (II)	360
17	HW #8A: SQL Window Functions: OLD	362
18	HW #8A: SQL Window Functions	364
19	HW #8B: Pandas Window Functions	365
20	BART Project	367
21	HW #5A: Info Schema and Price-Volume Relationship*	370
Appendix D	Example Exams	373

1	2023 CAPP Databases Final: A	374
2	2023 CAPP Databases Final: B	380
3	2023 CAPP Databases Midterm: A	386
4	2023 CAPP Databases Midterm B	390
7	2017 SQL Final	412
7	2017 SQL Final	412
7	2017 SQL Final	412
8	2018 SQL Final	421
9	2019 Exams	432
10	USF's student table	451
11	F&F Example	458
12	The Sales Rollup	465
13	Sales Work Through	468
14	Sales Work Through, part 2	473

DRAFT

Introduction and Errata

DRAFT

DRAFT

Introduction & Errata

Thank you for your interest in learning Data Management via SQL and Python. The material in these lecture notes covers the vocational aspects of learning these tools in a systematic and consistent manner.

Thank you for your interest in learning SQL! At the end of this course, you will be familiar with SQL and comfortable using it in a variety of real-world situations. While we directly use PostgreSQL in the notes, nearly all of the syntax presented is compatible with alternative SQL implementations. In those cases where there are compatibility issues, we try to call them out and address them.

Neither Pandas or SQL is *difficult*, but like learning any other language it requires time and practice. The purpose of these notes and problems are not to be a readable book, but instead a set of notes which are both a reference and guide. The majority of the learning that occurs is not within the text, but within the problem sets and their solutions.

Each module within this text is designed to be a (roughly) one hour lecture. At different levels and experience it is possible for some to run short and others to run long. This course has been taught in as little as 7 weeks to masters level students and taken as long as an entire semester at the undergraduate level. Dependencies between different modules are fairly obvious and quite a bit of the more technical material can be treated as an extension (specifically Modules 4, 10, 12 and 13 and easily skip-able).

The course material is designed to be amenable to a few different environments. It has been taught at the undergraduate level, undergraduate level as well as a free-standing executive certificate. While the primary learning objectives are the same in each of these environments there are (obviously) different expectations around this course at each of these levels.

Undergraduate

At the undergraduate level this has been taught as a semester long course which was paired with an applied machine learning section. When teaching at the undergraduate level students were provided with access to a cloud-based relational database, with limited permissions, that contained the databases used in this course.

The coursework in these notes was paired with a group project and research paper write-up. Students were required to load their data into a database and then created a set of jupyter notebooks and Python libraries to access the data and execute on their own research plan and agenda. Groups of students then did final presentations and long-form write-ups.

Graduate

At the graduate level this has been taught as both a once-a-week, 7-week long introduction to SQL as well as a twice a week quarter length course covering both SQL and Pandas. In the former situation only

the core modules regarding SQL syntax were covered while the later included all information presented in these notes.

When teaching at the graduate level, the raw data was provided to the students with the expectations that they would be able to load it into their own SQL (local) instances and work from there. Homework problems are lightly graded and quizzes are given each week in order to assess current knowledge retention.

Executive Certificate

This course material was also taught as an Executive Certificate in a once-a-week, 3 hours per week, 7 week long format. During each week, save the first and last, students were given a short self-assessed quiz. An hour long lecture was then done followed by working on problems from the assignments with the goal to finish the “first five” for the sections covered.

When teaching in this format, only the core SQL syntax was covered and students were provided access and credentials to a cloud-based server which contained the data for the course.

Errata and WIPs

This document is a work in progress and contains quite a few known issues. This preface contains known issues and places where improvements are required.

Overall

1. Remove every reference to Module and change to chapter.
2. Fix the interview notes.
3. DDL
4. Categorical Data
5. Add a section to the start of the Pandas regarding “state” and how, unlike SQL, there is a current “state” of a DataFrame. E.g. row numbers matter a lot.
6. Vector DB Discussion: <https://www.ethanrosenthal.com/2023/04/10/nn-vs-ann/>

SQL

1. Rewrite cast section. Currently confusing.
2. Queries with out a from clause and discussion of select 1 put at start mod 2.
3. Add to the start of the book <https://www.amazingcto.com/postgres-for-everything/>
4. Simple correlated subquery example. Current example is far too complex.
5. Rewrite NoSQL section adding information about a vector and graph databases:
 - Look at: https://www.theregister.com/2023/03/08/great_graph_debate_wednesday/
6. Add a resources section to the introduction which contains information on different books to consider.
 - <https://postgrespro.com/community/books/internals>
7. Online PostgreSQL explainer: <https://explain.dalibo.com/>
8. Add to performance consideration section. Discussion on the extreme case of super wide tables and how it effects performance: <https://www.cybertec-postgresql.com/en/column-order-in-postgr>
9. Add more examples of aggregation with case statements.
10. Add more formality to the discussion on what is returned and how it can be used in table vs. scalar.
11. <https://carlineng.com/?postid=sql-critique#blog>
12. Use MTA data and add hour to create timestamp in the date/time section. Add more date/time examples to date and time section, including intervals.
13. Language around analytic functions and LTV incorrect and needs to be fixed.
14. Check GSN and Zynga Dates. Add photos from Zynga as well as their MySQL solution.

Pandas

1. Add rank aggregation.
2. In section #1 the way that value_counts and column selection occurs is awkward. Maybe change the ordering to move value counts to after column selection.

3. Add `reindex` to index discussion in pandas. Overall discussion of multi-index and `reindex` needs to be updated.
4. Move `loc` not accepting `NaN` to first module
5. Add example for `duplicated`
6. Date stuff needs to be redone, both as index (additional section) and as regular/type discussion.
7. `MTADF` comes out of nowhere in the module 2. When is it first introduced?
8. Other resources to look into:
 - Effective Pandas <https://store.metasnake.com/effective-pandas-book>
 - <https://betterprogramming.pub/pandas-illustrated-the-definitive-visual-guide->
 - copy warning: <https://stackoverflow.com/questions/32573452/settingwithcopywarning>
9. `day_name` vs. `weekday_name`
10. https://www.practicaldatascience.org/html/views_and_copies_in_pandas.html
11. Add a short section on creating simple dataframes with dictionaries or lists
12. Move `applymap` `map` and `apply` to the next module and then take all the slice stuff and move it to the module with the `groupby` object.
13. Cuts / bins
14. `iterrows`
15. Any / All
16. Pivot
17. Add a section on analyzing the transaction data using Window Functions. Specifically, mimic the functions in the advanced joins.
18. HW #8B needs work, add more.
19. Loading and Saving Data
20. Time Series stuff
21. There are some questions in the HW about correlation, just go over this.