Advanced Data Modelling Techniques

Chris Travers

April 17, 2018

Relational Databases

- Inferential systems
- Based on function composition
- Functions and function composition are the basis of normalization

Object-Relational Databases

This course is about

- Intelligent datatypes
- Derived information
- Can extend to derive complex things

Joins as Function Composition

- Relations are Functions
- In relations, fields are functions of keys.
- Foreign keys define function compositions
- Therefore we can define relations from joins

Exercise 1

First exercise.

what does this do?

select id, lock_record(r) from my_table r where id = 10;
What rows have lock_record() run against them?

Functional Programming

- FP is about passing functions around
- FP can optimize by composing functions
- FP can optimize when to run functions

(Go through Create Function syntax)

Volatility

Volatility levels:

- Immutable
- Stable
- Volatile

(Live demo of stable and immutable on random).

Functions and Indexes

- Functions can be indexed
- Exressions can be indexed
- These are run on write and can then sometimes be skipped on read

Exercise 2

Second Exercise. Here we play around and look at expression-based indexes

Discussion 2

Does it ever make sense to have an immutable function that reads from a table?

Table Methods

PostgreSQL provides “table method” syntactic sugar

1. t.val checks to see if an attribute exists
2. if not, checks to see of a val(t) function exists

(Live Demo)

Best Practices

- Make table methods immutable
- Since you have the whole row, use it.
- For compatibility with Pg before 9.4 (iirc) use $1
- Understand the limits of function composition in Postgres

Exercise 3

Imagine search that spans fields! we do this.

Discussion 3

What are the pros/cons of this approach?

Semistructured Data

- Often data is not perfectly structured
- Data may be present or not
- Structured text (EMBL or Uniprot in life sciences)
- Could be XML or JSON

Use Cases

- Search on dates extracted from text files
- Search on emails as values in JSONB

(Live Demo)

Exercise 4

Now let's do some text processing

Discussion 4

Difficulties and dangers of semi-structured data

Defining 1NF

- All relations have same number of fields
- All fields are atomic.

What does atomic mean? Discussion....

Reasons to break 1NF

- Optimizing access patterns
- Ingestion
- Others?

Design Options

- Arrays of Tuples
- JSON/JSONB
- Custom data types

(Life demos of all)

Exercise 5

Now your turn to play with Non-1NF designs

Discussion: Problems and solutions of non-1NF designs.

What is Inheritance in Pgsql

- Implicit casts
- Consistent interfaces

(Go through list of caveats regarding indexes)

What is Inheritence Good For?

- Collective Management of Similar Tables
- Consistency in Data Types
- Common functional interfaces

(Live Demo)

Exercise 6

Now to play with inheritance and notes.

Multiple Inheritance

We can also use multiple inheritance for “reverse partitioning” (discuss use cases)

Beware

This is not usual however, so:

- You may hit bugs (demonstrate type change bug)
- Column changes are hard
- Inheritance diamonds become actually useful

Final Exercise

Q/A on inheritance and multiple inheritance

Thanks everyone! Email me at chris.travers@adjust.com if you have any more questions or suggestions.
(Further discussion available during hallway track).