Database Pitfalls

Christopher Gallo

NoSQL v SQL

NoSQL

- LARGE DATA
- variety of columns in a data set
- Documents Store
  - Usually has a version system
  - Eventually consistent
- Key-Value
- Graph
  - Nodes, Edges, Relations
- Wide-Column Store
  - Like a relational DB, but column size can very in a table

SQL

- RELATIONAL data
- Tables
- Predefined schema
- Good for complex queries
- ACID compliant
  - Atomicity, Consistency, Isolation, Durability

Indexes

- Explain explained

```
EXPLAIN SELECT state_name FROM State WHERE state_cd = 'TX';
+-------+------+---------------+--------+---------+-------+------+------------+
| table | type | possible_keys | key    | key_len | ref   | rows | Extra      |
+-------+------+---------------+--------+---------+-------+------+------------+
| State | ref  | st_idx        | st_idx | 2       | const | 1    | where used |
+-------+------+---------------+--------+---------+-------+------+------------+
```

- Dont over index
- NULLS
  - use anything else to indicate the absense of a value
- Appropriate use of datatypes
  - varchar for date or ip is bad
- It is OK to have un-normalized data
  - Joins have a performance cost

Triggers and Constraints

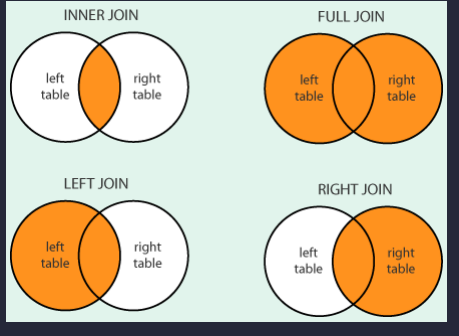
- Obscurity
  - Trigger logic should be in application
  - Can be easily forgotten if not documented
- Complexity
  - A good source of unexpected behavior, aka bugs
- Perforamace
  - Triggers run every time a Data Manipulation Language (DML) operation runs

What not to store

- Images
- unstructured data you want to search
- Unencrypted Sensitive data
  - Hash AND salt passwords
  - Don't use MD5 or SHA1

Joins

- Joins > subquery



Performance Tuning

- acquire connection only when needed, and immediately close
- db files should live on their own storage
- understand the query cache

Big data and scaling up

- table partition based on timestamp
  - only partition on something else if it REALLY makes sense
- sharding
- multi master setups
- having a read sql string and a write sql string