

OKC MySQL Users Group



OKC MySQL

- Discuss topics about MySQL and related open source RDBMS
- Discuss complementary topics (big data, NoSQL, etc)
- Help to grow the local ecosystem through meetups and events

Sponsored By...



MySQL Schema Basics

Some General Terms

- Primary Key (PK)
 - Unique identifier for a row
- Secondary Index
 - Index on a non-PK column in a table
 - Can be unique
 - Allows for faster access to row (i.e. phonebook)
- Normalized
 - Removing duplicate data from tables
 - Leverages JOINS to prevent storing data in multiple places

Topics

- Normalization
 - What is it and why does it matter?
 - Caveats to normalization
- Indexing
 - Basic concepts
 - Things to consider
- Data Types
 - Why do they matter?
 - How should I choose what to use?

Normalization

*Database normalization (or normalisation) is the process of organizing the columns (attributes) and tables (relations) of a relational database to **minimize data redundancy**.*

https://en.wikipedia.org/wiki/Database_normalization

Normalization (cont.)

- First Normal Form (1NF)
 - Table for each type of data
 - No repeating data for a field
 - Each row has a primary key
- Second Normal Form (2NF)
 - Table meets 1NF requirements
 - No potential “candidate key” can be duplicated
- Third Normal Form (3NF)
 - Table meets 2NF requirements
 - Every non-key attribute must give a fact about the whole key and only the key

First Normal Form

Customer

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659 555-776-4100
789	Maria	Fernandez	555-808-9633

Non-1NF

Customer

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659
456	Jane	Wright	555-776-4100
789	Maria	Fernandez	555-808-9633

→ 1NF

Second Normal Form

Electric Toothbrush Models

Manufacturer	Model	<u>Model Full Name</u>	Manufacturer Country
Forte	X-Prime	Forte X-Prime	Italy
Forte	Ultraclean	Forte Ultraclean	Italy
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush	USA
Kobayashi	ST-60	Kobayashi ST-60	Japan
Hoch	Toothmaster	Hoch Toothmaster	Germany
Hoch	X-Prime	Hoch X-Prime	Germany

Non 2NF

→ 2NF

Electric Toothbrush Manufacturers

<u>Manufacturer</u>	Manufacturer Country
Forte	Italy
Dent-o-Fresh	USA
Kobayashi	Japan
Hoch	Germany

Electric Toothbrush Models

<u>Manufacturer</u>	<u>Model</u>	Model Full Name
Forte	X-Prime	Forte X-Prime
Forte	Ultraclean	Forte Ultraclean
Dent-o-Fresh	EZbrush	Dent-o-Fresh EZbrush
Kobayashi	ST-60	Kobayashi ST-60
Hoch	Toothmaster	Hoch Toothmaster
Hoch	X-Prime	Hoch X-Prime

Third Normal Form (normalized)

Tournament Winners

<u>Tournament</u>	<u>Year</u>	<u>Winner</u>	<u>Winner Date of Birth</u>
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Non 3NF

Tournament Winners

<u>Tournament</u>	<u>Year</u>	<u>Winner</u>
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner Dates of Birth

<u>Winner</u>	<u>Date of Birth</u>
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

→ 3NF

Where it breaks down?

- Normalization is very important for OLTP workloads – minimizes writes on non-key data
- For analytics, can cause problems
 - JOINS are slower for aggregate
 - Star schema (duplication) often preferred
 - Duplicate keys are repeated for ease of reporting

Indexing

CARTIN—CASS 45	
CASEY Robt L 32 Scott Cir Ded.....	326-2370
Robt & Lauren 8 Otis Av Ded.....	326-0635
Robt W 71 Herbert Rd Bra.....	848-5137
Rose 617 Broad Wey.....	331-6948
Ruth B 208 Atlntc Av Hul.....	925-2525
Sean 141 Carroll Av Wswd.....	329-9412
Stacie 30 Chapman Wey.....	331-3652
Stephen & Katherine 590 Middle Wey.....	340-6658
Stephen R 17 Draper Can.....	828-1108
Theresa 17 Binnacle Ln Qui.....	479-4923
Thos 65 Dickens Qui.....	328-6078
Thos A Jr 194 School Wswd.....	828-2474
Thos R 1 Wentworth Rd Can.....	828-6969
Thos W 20 Westchester Dr Wswd.....	326-1140
Timothy P 30 French Qui.....	328-4662
Timothy P 637 Pleasant Mil.....	698-3922
Tom & Judy 63 Hollis Av Qui.....	328-0734
V 9 Fore River Av Wey.....	331-5746
Walter J Jr 17 Autumn Cir Hin.....	740
Wm C 212 Central Av Humrck.....	
Wm L 36 Bayley Wswd.....	
Ben & Bill 11 King.....	
96B Atlantic.....	
a A.....	

Concepts

- Same as a phonebook or index in a book – help you find your data faster based on a pointer
- Primary Key is main index – fastest way to look up data and definition of location on disk
- Secondary index is attached to other column(s)
- Clustered index (InnoDB)
 - PK is appended to the end of the secondary index for faster lookup
 - Great for performance, impact on size

Types of Indexes

- Primary Key
- Unique Index
- B-Tree or R-Tree Index (normal secondary index)
- Index on numeric, string, or date columns (and spatial columns as of MySQL 5.7)
- Foreign Keys
 - Not explicit index, but requires column to be indexed for performance
 - Enforces referential integrity
 - Handles cascading updates/deletes

Index Basics

- Index can be comprised of one or more columns in the table
- Indexes are read from left to right to utilize multiple columns
- Used for:
 - Lookups
 - JOINS
 - Ordering
- Can be equality or range
 - Once range field is used, rest of index can't be used

Index Example

```
CREATE TABLE `ip_varchar` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `t1_id` int(11) DEFAULT NULL,  
  `data` varchar(255) DEFAULT NULL,  
  `ip_address` varchar(15) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `t1_id` (`t1_id`),  
  KEY `ip_address` (`ip_address`)  
) ENGINE=InnoDB
```

- Primary Key is an integer (optimal)
- 2 Secondary indexes, one on related table and one on ip_address

Index Caveats

- Indexes add overhead to write queries
- In InnoDB, indexes can also add significant space to the table due to the clustered index (discussed before)
- Indexes do not need to use the entire set of columns
 - Index1 (col1, col2) will cover these:
 - Where col1 = x
 - Where col1 = x and col2 = y
 - Index1 (col1, col2) will not cover:
 - Where col2 = y

Data Types

- Data types are frequently overlooked, but are very important in schema design and optimization
- Some data types are fixed in size while others are variable
- Types include:
 - Numeric (int, float, double, decimal)
 - String (varchar, char, text)
 - Date (timestamp, datetime)
 - Binary (blob)

General Rules

- Use the smallest data type you can comfortably use for a reasonable amount of growth
 - i.e. don't use bigint unless you absolutely have to!
- Use timestamps to record activity, use dates for historical purposes
- Don't store binary data in the database
 - i.e. just because you can store a PDF in a blob field, don't do it

Open Discussion...

Thanks for coming!

Website: <http://okcmysql.org>

Twitter: @okcmysql

Email: mike@okcmysql.org