

Relational Database Design Bootcamp

Mark Niebergall

Longhorn PHP 2019

<https://joind.in/talk/9d7a6>

<https://github.com/mbniebergall/database-bootcamp>

Mark Niebergall



- PHP since 2005
- Masters degree in MIS
- Senior Software Engineer
- Drug screening project
- Utah PHP Co-Organizer
- CSSLP, SSCP Certified and SME
- Father, long distance running, fishing, skiing



Survey

- Beginner - know of databases but haven't written SQL?
- Intermediate - know how to write queries, create tables?
- Advanced - database administrator, years of experience?

Relational Database Design Bootcamp

- 3 hour tutorial
- Hands-on learning
- Learn concepts
- Design databases



Relational Database Design Bootcamp

- 1:00-1:30 Key concepts
- 1:30-2:00 Class Project Design
- 2:00-2:10 Break
- 2:10-2:30 Class Project Design (continued)
- 2:30-3:00 Group Projects
- 3:00-3:10 Break
- 3:10-3:30 Group Projects
- 3:30-3:50 Group Project Presentations
- 3:50-4:00 Review and Discussion

Relational Database Design Bootcamp

- Ask questions anytime
 - Others probably have same question

Objectives

- Increased knowledge and experience with relational database design

Objectives

- Know how to create tables with relationships
- Ensure data integrity
- Understand database normalization
- Tweak database performance

Setup

- Access to any MySQL server
 - Free community server <https://dev.mysql.com/downloads/mysql/>
 - Ideally version 8, minimum 5.7+
 - MySQL client of your choice
 - ▶ Sequel Pro: Mac, free, <https://www.sequelpro.com/>
 - ▶ PhpStorm: <https://www.jetbrains.com/help/phpstorm/relational-databases.html>
 - ▶ Terminal or Cmd

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Use Cases

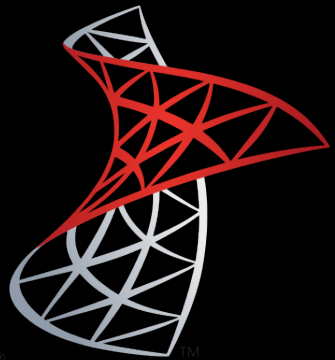
- Database
 - Storage of data
 - Organized into tables
 - Relationships represented with constraints
 - Replacement for spreadsheets

Use Cases



PostgreSQL

ORACLE



Microsoft®



Use Cases

- Online store
- Students at a school
- Warehouse inventory
- Medical providers
- Biology research

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Data Types

- Name all 8 PHP data types?
- Name 2 PHP pseudo data types?

Data Types

- Name all 8 PHP data types?
 - Scalar: boolean, integer, float, string
 - Compound: array, object
 - Special: resource, null
- Name 2 PHP pseudo data types?
 - Compound: callable, iterable

Data Types

- Database types
 - Numeric
 - Date and time
 - String
 - JSON

Data Types

- Numeric
 - BIT: BIT(4) 1001
 - BOOL: 0 or 1
 - INT: +/- 2147483647
 - DECIMAL: DECIMAL(6, 2) 123456.78

Data Types

- Date and Time
 - DATE
 - TIME
 - DATETIME
 - TIMESTAMP

Data Types

- String
 - CHAR
 - VARCHAR
 - BLOB
 - TEXT
 - ENUM

Data Types

- JSON
 - {"name": "widget", "description": "does something"}
 - [123, 4567]
 - {"id": 321, "codes": ["abc", "xyz"]}

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Create Tables

- Data modeling
 - Logical grouping
 - Reduce duplicate data
 - Table relationships

Create Tables

- `CREATE DATABASE warehouse;`

Create Tables

- USE warehouse;

```
CREATE TABLE item (  
    id INT NOT NULL AUTO_INCREMENT,  
    code VARCHAR(20) NULL,  
    quantity INT NOT NULL DEFAULT 0,  
  
    PRIMARY KEY (id),  
  
    CONSTRAINT UK_item_code  
        UNIQUE KEY (code)  
  
) ENGINE=INNODB;
```

Create Tables

- USE warehouse;

ALTER TABLE item

ADD COLUMN description VARCHAR(200) NULL
AFTER code;

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Constraints

- Ensure data
 - CONSTRAINT UK_key_name
UNIQUE KEY (column_name[, column_b, ...])
 - NULL value considerations

Constraints

- Ensure data
 - CONSTRAINT FK_name
FOREIGN KEY (table_column)
REFERENCES another_table (that_table_column)
ON DELETE NO ACTION
ON UPDATE CASCADE

Constraints

- Ensure data
 - CONSTRAINT FK_name
FOREIGN KEY (column_a, column_b)
REFERENCES another_table (column_a, column_b)
ON DELETE NO ACTION
ON UPDATE CASCADE

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Normalization

- Reduce data redundancy
- Increase data integrity

Normalization

- Levels of Normal Form
 - 1NF
 - 2NF
 - 3NF
 - EKNF (Elementary Key)
 - BCNF (Boyce-Codd)
 - 4NF
 - ETNF (Essential Tuple)
 - 5NF
 - DKNF (Domain Key)
 - 6NF

Normalization

- Considerations
 - Performance
 - Data size

Normalization



Food Inventory Sheet - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Team

Normal Page Layout Page Break Preview Custom Views Full Screen

Workbook Views

Ruler Formula Bar

Gridlines Headings

Show

Zoom 100% Zoom to Selection

Zoom

New Window Arrange All Freeze Panes

Split Hide Unhide

View Side by Side Synchronous Scrolling Reset Window Position

Window

Save Workspace Switch Windows

Macros

Macros

H8 fx 0

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Item #	Description	Vendor	Category	Size	Unit	Starting Qty	Starting Value	Wk 1 Qty	Wk 1 Cost	Wk 2 Qty	Wk 2 Cost	Wk 3 Qty	Wk 3 Cost	Wk 4 Qty
2	492229	TURKEY SLICED .5 OZ	Ben E Keith	2 - FROZEN FOOD	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
3	662371	DRESSING CAESAR CREAMY	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
4	779243	MARGARINE LIQUID OLEO	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
5	815306	LID PLAS SOUFFLE CLEAR	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
6	860055	LID PLAS 16SL SLOTTED	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
7	860060	CUP FOAM 16OZ 16J16	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
8	774704	PAPRIKA	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 5.79	0.00	\$ -	0.00
9	664005	Mustard Prepared	Ben E Keith	4 - GROCERY	512	fl oz	0.00	\$ -	1.00	\$ 3.75	0.00	\$ -	0.00	\$ -	0.00
10	750100	CHEESE PARMESAN SHRED	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	1.00	\$ 13.27	0.00
11	250025	EGG FRESH SHELL MED USDA AA	Ben E Keith	1 - PRODUCE	0	0	0.00	\$ -	1.00	\$ 15.89	0.00	\$ -	0.00	\$ -	0.00
12	686034	VINEGAR APPL CIDER 40GRAIN	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 17.77	0.00	\$ -	0.00
13	29078	LIME 12 CT	Ben E Keith	1 - PRODUCE	12	ct	0.00	\$ -	2.00	\$ 8.99	0.00	\$ -	0.00	\$ -	0.00
14	650547	TOMATO DICED W/GREEN CHILES	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	1.00	\$ 18.88	0.00	\$ -	0.00	\$ -	0.00
15	286500	Ice Cream Vanilla Cr 3 Gal	Ben E Keith	6 - DAIRY	384	fl oz	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
16	650474	KETCHUP FANCY 33% SOLIDS	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	1.00	\$ 20.69	0.00	\$ -	0.00	\$ -	0.00
17	140005	MUSHROOM WHITE SMALL BUTTON	Ben E Keith	1 - PRODUCE	0	0	0.00	\$ -	1.00	\$ 20.98	0.00	\$ -	0.00	\$ -	0.00
18	771131	CROUTON SEASONED HOMESTYLE	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 22.30	0.00	\$ -	0.00
19	660409	SAUCE LOUISIANA RED HOT	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	1.00	\$ 11.24	0.00	\$ -	1.00	\$ 11.24	0.00
20	150015	Onion Green Iceless W/Root	Ben E Keith	1 - PRODUCE	32	oz	0.00	\$ -	1.00	\$ 8.29	1.00	\$ 8.29	0.00	\$ -	0.00
21	780009	SUGAR BROWN LIGHT IN BAGS	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 27.69	0.00	\$ -	0.00
22	155030	Onion Yellow Jumbo	Ben E Keith	1 - PRODUCE	800	oz	0.00	\$ -	0.00	\$ -	1.00	\$ 13.99	0.00	\$ -	0.00
23	774173	Pepper Red Crushed	Ben E Keith	4 - GROCERY	52	oz	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00
24	920919	TUMBLER 20 OZ AMBER	Ben E Keith	8 - EQUIP & SUPPLY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 29.99	0.00	\$ -	0.00

Inventory Graphs DescriptionLookup CategoryLookup Week1 Week2 Week3 Week4 Week5

Select destination and press ENTER or choose Paste

100%

<https://i.stack.imgur.com/7JoKT.jpg>

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Design Principles

- Practical data mapping

Design Principles

- Naming that makes sense

Design Principles

- Correct data types

Design Principles

- Accurately reflects relationships
 - Boil down to simplest form

Design Principles

- S - Single Responsibility: do one thing well
- O - Open-Closed: open for extension, closed for modification
- L - Liskov Substitution: replaceable with subtypes
- I - Interface Segregation: break apart interfaces logically
- D - Dependency Inversion: dependency injection

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Performance

- Decreases as data volume increases

Performance

- Denormalization
 - Use constraints to retain data integrity

Performance

- Database indexes
 - `INDEX IX_name (column_a)`
 - `ALTER TABLE some_table
ADD INDEX IX_name (column_a, column_b);`

Performance

- Avoid
 - LIKE '%abc%'
 - JOIN table t ON t.a = x.a OR t.b = y.b
 - WHERE (t.a = x.a OR t.b = y.b)
 - COUNT(*) [no limit set]
 - ORDER BY CONCAT(non_index_varchar_a, b)
 - GROUP BY a, b, c

Performance

- EXPLAIN
 - EXPLAIN SELECT...
 - EXPLAIN DELETE...
 - EXPLAIN INSERT...
 - EXPLAIN REPLACE...
 - EXPLAIN UPDATE...

Performance

- EXPLAIN
 - Fast: Using index
 - Okay: Using where
 - Slow: Using filesort
 - Slow: Using temporary

Performance

```
EXPLAIN
SELECT COUNT(b.code) AS building_code_count,
       b.code,
       i.description
FROM building b
JOIN rack r
    ON r.building_id = b.id
JOIN shelf s
    ON s.rack_id = r.id
JOIN location l
    ON s.id = l.shelf_id
JOIN location_item li
    ON l.id = li.location_id
JOIN item i
    ON i.barcode = li.item_barcode

WHERE b.code LIKE '%5%'
AND i.description LIKE '%pig%'

GROUP BY b.code, i.description;
```

Performance

1,SIMPLE,b,,index,PRIMARY,UK_building_code,22,,8,12.5,Using where; Using index; Using temporary; Using filesort
1,SIMPLE,i,,ALL,PRIMARY,,,,10,11.11,Using where; Using join buffer (Block Nested Loop)
1,SIMPLE,r,,ref,"PRIMARY,FK_rack_building",FK_rack_building,4,warehouse.b.id,9,100,Using index
1,SIMPLE,s,,ref,"PRIMARY,UK_shelf",UK_shelf,4,warehouse.r.id,9,100,Using index
1,SIMPLE,l,,ref,"PRIMARY,FK_location_shelf",FK_location_shelf,4,warehouse.s.id,9,100,Using index
1,SIMPLE,li,,eq_ref,"PRIMARY,UK_item_location",PRIMARY,46,"warehouse.l.id,warehouse.i.barcode",1,100,Using index

Performance

- Make it work, make it fast, make it right
- Performance tuning during project building

Performance

- Considerations
 - Disk space
 - Breaking queries apart
 - Combining queries
 - Stored procedures
 - Views
 - Functions
 - User workflows

Overview

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Project

- Warehouse inventory on shelves

Project



<http://cdn.idplate.com/images/warehouse-rack-bar-code-labels.jpg>

Project

- <https://github.com/mbniebergall/database-bootcamp>



Team Project

- Groups of about 4
- Work together on design
 - Data model with normalization
 - Create database
 - Create tables
 - Populate tables
 - Add indexes

Team Project

- Parking lot
- Elephant collections
- Family Tree
- Exercise tracker
- Messenger queue
- Pet daycare
- Sports roster
- Driver licenses
- Airline seating

Team Project

- Share
 - Project chosen
 - Tables
 - Relationships
 - Normalization
 - Performance
 - Difficulties

Team Project



Review and Discussion

- Use cases
- Data types
- Create tables
- Constraints
- Normalization
- Design principles
- Performance

Review and Discussion

- Class project
- Group projects

Review and Discussion

- Discussion
 - Denormalization
 - Performance tuning
 - Security
 - Business Logic

Review and Discussion

- Discussion
 - Stored Procedures
 - Functions
 - Triggers
 - Views

Review and Discussion

- Anything PHP

Questions?

- Rate on joind.in <https://joind.in/talk/9d7a6>