IT125 SQL: Designing a Database
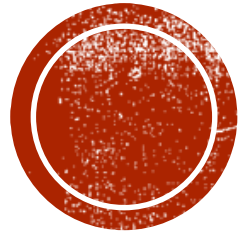
Bill Barry

TONIGHT

- Prepare for Midterm Exam
- Given client artifacts, identify:
  - Tables
  - Columns
  - Keys
  - Relationships
  - Indexes
- Understand and use normalization
  - And understand how it affects data, data actions, and performance
- Use MySQL Workbench to create an EER

PART I: DESIGN AND NORMALIZE A DATABASE

STARTING POINT

- You may start with interviews, you may start with artifacts

- As you ask questions, you'll learn there's more than meets the eye; the way the client presents the data is often *not* the way it needs to be structured in the database

- The first task is to jot down what you've got and start separating out entities, "One of these things is not like the others"

- You also watch for and eliminate repeated data

- Think about potential primary keys as you go; these will be also be a guide as all column data should depend on "the key, the whole key, and nothing but the key"

- Technically, this is part of Database Normalization

- With experience and practice, this will "just feel right"

NORMALIZATION BASICS

- First Normal Form (1NF)   "The key…"
  - Create separate tables for separate entities
  - Identify a unit PK, single column or composite
  - Ensure all cells contain atomic data (no repeats)
  - Eliminate columns that repeat similar sets of data

- Second Normal Form (2NF)   "…the whole key…"
  - If columns don't depend on the PK (which includes all parts of composite keys), move them out

- Third Normal Form (3NF)   "…and nothing by the key"
  - Eliminate columns that don't depend on the key alone, ("transitive dependencies")

EFFECTS OF NORMALIZATION #1

+ Minimizes duplicate data
  - Decreases storage (fewer rows → less storage)
  - Increases performance (less data to search)

+ Simplifies queries
  - Don't have to search multiple columns or expect multiple rows containing the same data
  - Sorting makes more sense; data isn't distributed over columns

- Requires more JOINs
  - Separate entities live in separate tables; joining is often required
  - There is some performance overhead vs. less-normalized tables
  - Note: indexes make this fast; we'll soon discuss those in more depth

EFFECTS OF NORMALIZATION #2

+ Avoids or minimizes data modification issues including…
  - Insert Anomaly: can't insert all data because information is too distributed among tables
  - Update Anomaly: must make changes on multiple rows because of dups
  - Deletion Anomaly: row removal causes loss of data (in other tables)

  See this site for some easy-to-understand examples of these:
 https://www.essentialsql.com/get-ready-to-learn-sql-database-normalization-explained-in-simple-english/

DESIGN: ORIGINAL

Member

Member	Phone	Zip	Office	Committees (Joined)	2016 Dues $20	2015 Dues $15
Dana Davenport	206-752-6207	98027		Academics (8/1/15)	$20 (1/12)	$15 (1/27)
Pat Portillo	206-697-9703	98029	President (16)	Recruitment (9/12/15), Annual Party (1/12/16)	$10 (1/5), $10 (1/19)	$10 (1/6), $5 (1/20)
Chris Chisholm	206-474-2908	98027	President (15)	Academics (10/05/15), Annual Party (1/5/16)		$15 (1/13)
Lupe Lopez	206-491-6713	98008	Vice-Pres (16)	Recruitment (1/10/16)	$20 (1/19)	$5 (1/6), $10 (1/13)
Marty Montoya	206-750-9175	98029	Secretary (16)		$5 (1/12), $15 (1/19)	n/a
Avery Archibald	206-626-4378	98027	Vice-Pres (15)	Academics (11/13/14)		$15 (1/27)

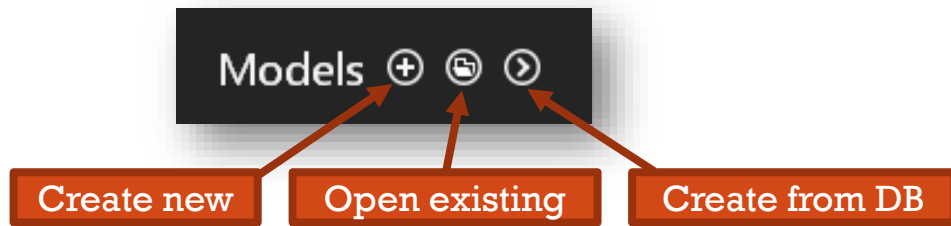What issues catch your eye? Let's work through each issue and normalize as we go…
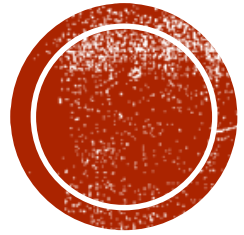
PART II: USE MYSQL TO DRAW AN EER

EER MISCELLANEOUS

Models ⊕ ⊡ ⊙

Create new   Open existing   Create from DB

Models (including table/column definitions and EERs) are saved in their own file format, extension .mwb
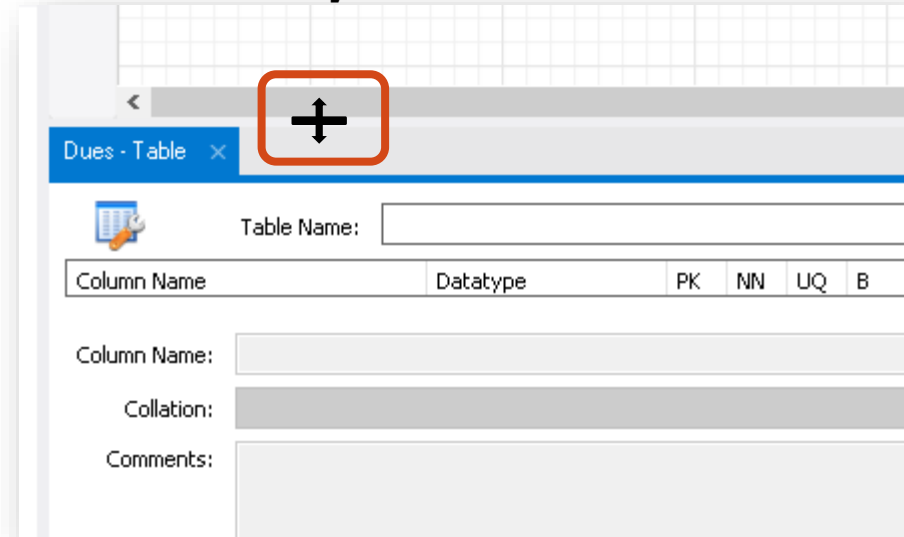
Two ways to open: from inside Workbench, or by double-clicking in the file explorer; try the other if one fails

Keys to know:
<Ctrl+NumPadPlus>   Zoom In
<Ctrl+NumPadMinus>   Zoom Out

"I can't see my columns!"


Identifying Relationships

- An **identifying relationship**: identified by a solid line between tables

An identifying relationship is one where the child table cannot be uniquely identified without its parent. Typically this occurs where an intermediary table is created to resolve a many-to-many relationship. In such cases, the primary key is usually a composite key made up of the primary keys from the two original tables.

- A **non-identifying relationship**: identified by a broken (dashed) line between tables

FINAL EER
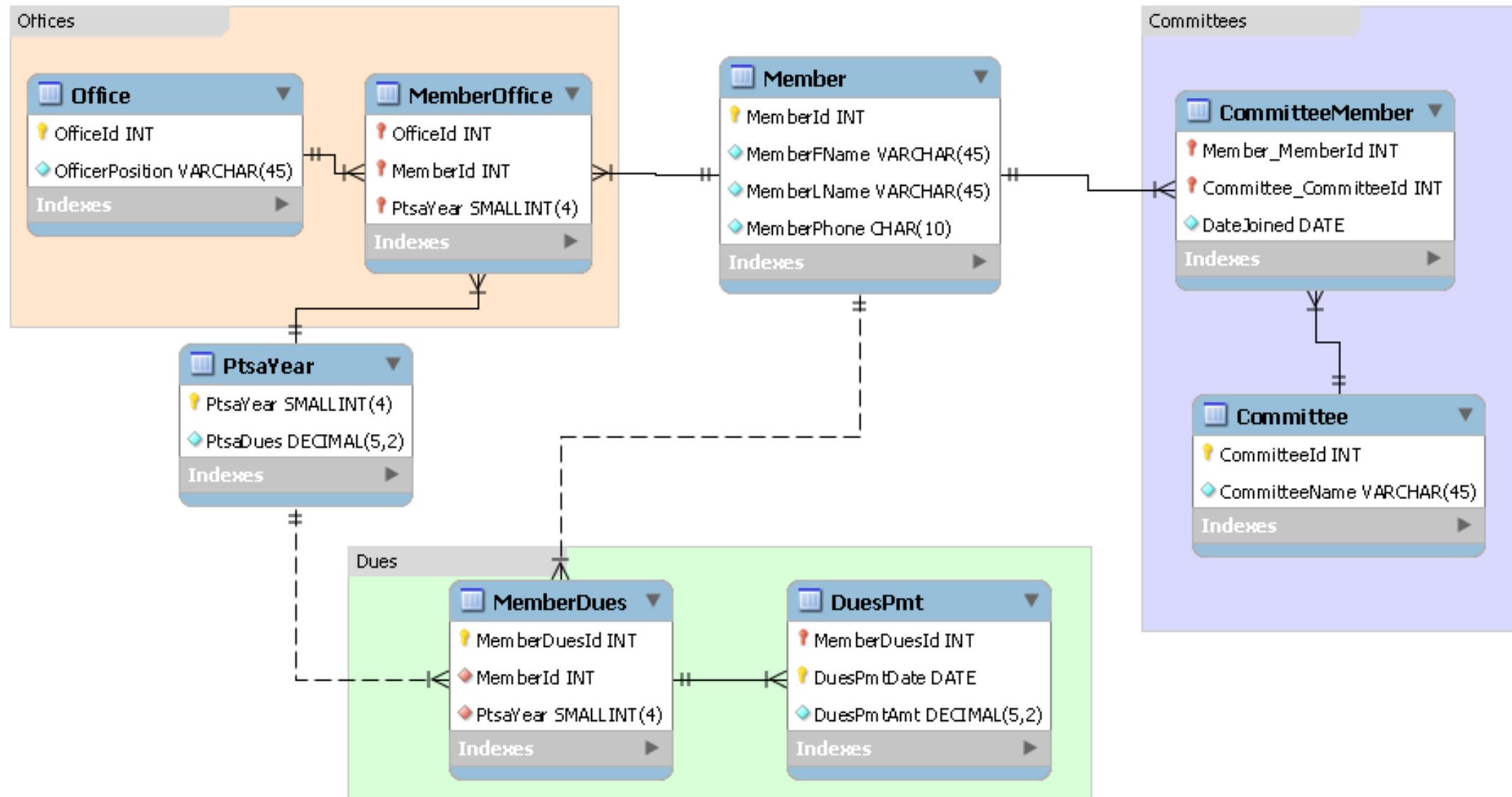
REFERENCES: DATABASE NORMAL FORMS

- EssentialsSQL – decent, simple overview with examples
  - Click through to specific pages discussing 1NF, 2NF, 3NF

- LifeWire

- ThoughtCo

- BeginnersBook

- Dummies.com

- Techopedia

- Transitive Dependencies

WHAT SHOULD I DO NEXT?

- Before **next week** (Midterm Exam)
  - Carefully read the exam preparation document
  - Consider re-watching key videos, including perhaps…
    - Understanding EER Diagrams (Modules, Week01), 14:29
    - Basics of Table Design (Modules, Week03), 14:53
  - Study; ask questions, if you have them

- Before **two weeks** from tonight…
  - Take the quiz on Chapter 10
  - Submit Proj05
    - Designing a Database

QUESTIONS?