Unit 5 Module 1: Introduction to mySQL

Dylan Lane McDonald

CNM STEMulus Center
Web Development with PHP

September 22, 2014

Outline

1. Set Theory
2. Introduction to Relational Database Management Systems
    - Introduction to Relational Database Management Systems
    - Indexes & Keys
    - Relations
3. SQL
    - Core Operations
    - Joins

Set Theory

A **set** is a collection of distinct objects. Sets and their operations serve as a mathematical motivation to how databases store, access, and relate data.

Operation	**Name**	**Comment**
$A \cap B$	Intersection	All items in common in A & B
$A \cup B$	Union	All items in A & B
$A \times B$	Cartesian Product	All items in A paired with B
$A \backslash B$	Compliment	All items in A not in B
$A \subset B$	Subset	All items in A exist in B
$\lvert A \rvert$	Cardinality	Number of items in A
$x \in A$	Element Of	x exists in A

Table 1: Set Operations

Example Set Operations

Let $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$.

$$\begin{aligned}
A \cap B &= \{3\} \\
A \cup B &= \{1, 2, 3, 4, 5\} \\
A \times B &= \{(1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 3), (3, 4), (3, 5)\} \\
A \backslash B &= \{1, 2\} \\
A \subset B &= \text{false} \\
|A| &= 3 \\
2 \in A &= \text{true}
\end{aligned}$$

These examples of all the operations in Table 1 will motivate how to visualize database operations and queries.

Common Sets

Set	Comment
$\emptyset$	Empty set $= \{\}$
$\mathbb{P}$	Prime numbers $= \{2, 3, 5, \ldots\}$
$\mathbb{N}$	Natural numbers $= \{1, 2, 3, \ldots\}$
$\mathbb{Z}$	Integers $= \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$\mathbb{Q}$	Rational numbers (all numbers $\frac{a}{b}$ where $b \neq 0$)
$\mathbb{R}$	Real numbers (all numbers without a complex part)
$\mathbb{C}$	Complex numbers (all numbers in the form $a + bi$)

Table 2: Common Sets

Note the row above is a subset of the row below in Table 2. i.e.,

$$\emptyset \subset \mathbb{P} \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

Introduction to Relational Database Systems

Definition

Relational Database Management Systems *(RDBMS) are systems that store and relate data using first order predicate logic. RDBMS allow concurrent access to the same data and maintain the integrity and relations of the data they contain.*

Introduction to Relational Database Systems

Definition

Relational Database Management Systems *(RDBMS) are systems that store and relate data using first order predicate logic. RDBMS allow concurrent access to the same data and maintain the integrity and relations of the data they contain.*

The key to RDBMS's success is the relationships shared among all the entities in the database, making the database flexible and searchable. The relations give multiple deterministic, efficient paths to the data required.

RDBMS Elements

RDBMS have the following elements:

1. **Database**: In this context, a database is a collection of related entities.
2. **Entity**: A particular model in a database. This is typically a class in an object oriented program. This is a set in set theory.
3. **Row**: A specific member of an entity. This is an object is object oriented programming. This is an element in set theory.
4. **Attribute**: A specific field of an entity. This is a state variable in object oriented programs.
5. **Relation**: A relationship two entities have on one or more attributes. This is implemented as a method in object oriented programming.

Indexes & Keys

Definition

*An **index** on an attribute in an entity that is stored in a data structure that is faster to search.*

Definition

*A **key** an index that uniquely identifies a row in an entity. If the key is the main identifier for the row, it is known as a **primary key**.*

Every entity should have a primary key defined. The primary key is critical in disambiguating among the other rows as well as serving as the typical identifier for relations.

Relations

Definition

A **relation** *is a semantically defined link between two entities. The relation will occur on one or more attributes.*

The three types of relations are:

1. 1-to-1: where one row in an entity refers to another entity and gets 0 or 1 results.
2. 1-to-n: where one row in an entity refers to another entity and gets 0 or more results.
3. m-to-n: where multiple rows can refer to another entity and get 0 or more results.

Facebook Relations

Consider a small cross-section of Facebook as an example database. Suppose we have the following entities:

- **User**: A user's username & password.
- **Timeline**: A user's posts, photos, etc.
- **Groups**: A group of users with common interests.

The relations in our small Facebook database would be:

- 1-to-1: One user can have only one timeline.
- 1-to-n: One user can post n items to her/his timeline.
- m-to-n: Many users can belong to multiple groups.

Core SQL Operations

The core SQL operations are:

- **Select**: Extract row(s) from an entity based on search criteria.
- **Update**: Update row(s) in an entity based on search criteria.
- **Delete**: Delete row(s) from an entity based on search criteria.
- **Insert**: Insert a new row into an entity.

These four operations are at the core of any SQL-aware application. In the Model-View-Controller design pattern, these four operations are the ones that are repeated continuously. In fact, in accordance of the Principle of Least Privilege, most database users are limited to these four operations.

Joins

Definition

A **join** *is the process of gathering data from two entities. A join is executed on a field that is common to both entities.*

The major types of joins are:

- **Inner Join**: All rows where field x is common to A and B - $A \cap B$
- **Left Outer Join**: All rows where field x is in A and not in B - $A \backslash B$
- **Right Outer Join**: All rows where field x is in B and not in B - $B \backslash A$