Ivey | Publishing

W18797

FOXCORE RETAIL (B): DATABASE IMPLEMENTATION

Liam Corrigan and Michael Gencarelli wrote this case under the supervision of Professor Derrick Neufeld solely to provide material for class discussion. The authors do not intend to illustrate either effective or ineffective handling of a managerial situation. The authors may have disguised certain names and other identifying information to protect confidentiality.

Foxcore Retail (Foxcore) was facing a time crunch. With only one month before the festival season began, the company's founders, Liam Corrigan and Mitchell Fox, were looking for an efficient and cost-effective solution to resolve their event and sales tracking problem. However, their budget was already stretched to cover costs related to the company's recent expansion. Therefore, Corrigan and Fox planned to implement a new company database on their own, rather than investing in high-cost technical services. The night before the start of the project, they had stayed up late designing the initial data model and making refinements (see Exhibit 1).

The data model helped bring the database from idea to reality. They could see exactly what they needed to implement before creating any of the data tables. Corrigan knew that this would make the implementation process much less prone to error. All of the entities, attributes, and relationships were defined correctly in the model. Everything was ready to begin the implementation. The only task left was to select the tools.

IMPLEMENTATION TOOLS

Corrigan recalled learning about XAMPP[1] in his data management class. He knew that he could create a MariaDB database using the XAMPP platform, using one of two approaches: Structured Query Language (SQL)[2] or phpMyAdmin.

Structured Query Language

The first option was to write and execute the database queries using SQL. This involved writing CREATE statements to convert each entity in the data model to a physical table in the database, convert attributes to fields, and to establish relationships and constraints between the various tables. For example, the

[1] XAMPP is a free Apache software distribution that makes it easy to set up a local web server with a database back end, named for its features that include cross-platform (X), Apache (A), MariaDB (M), PHP (P), and Perl (P); "XAMPP Apache + MariaDB + PHP + Perl," Apache Friends, accessed October 17, 2018, www.apachefriends.org/index.html.

[2] SQL is a standard language for storing, manipulating, and retrieving information from a database; "SQL Tutorial," W3Schools, accessed October 17, 2018, www.w3schools.com/sql.

following CREATE statement would create a table called "venue," in a database named "db_foxcore," with four attributes including a primary key.

```
CREATE TABLE db_foxcore.venue (
        id INT(4) NOT NULL AUTO_INCREMENT,
        name VARCHAR(30) NOT NULL,
        address VARCHAR(100) NOT NULL,
        description VARCHAR(300) NULL,
        PRIMARY KEY (id));
```

phpMyAdmin

Another option was to use a front-end software tool to assist with the implementation. One popular option was called phpMyAdmin[3] (see Exhibit 2). With point-and-click typing, this software could be used to create all the tables, attributes, and relationship constraints. Although making changes and revisions to the database over time could be a bit more challenging, compared to establishing an SQL codebase with the first method, phpMyAdmin offered a fairly quick and painless mechanism for setting up a first iteration.

GETTING STARTED

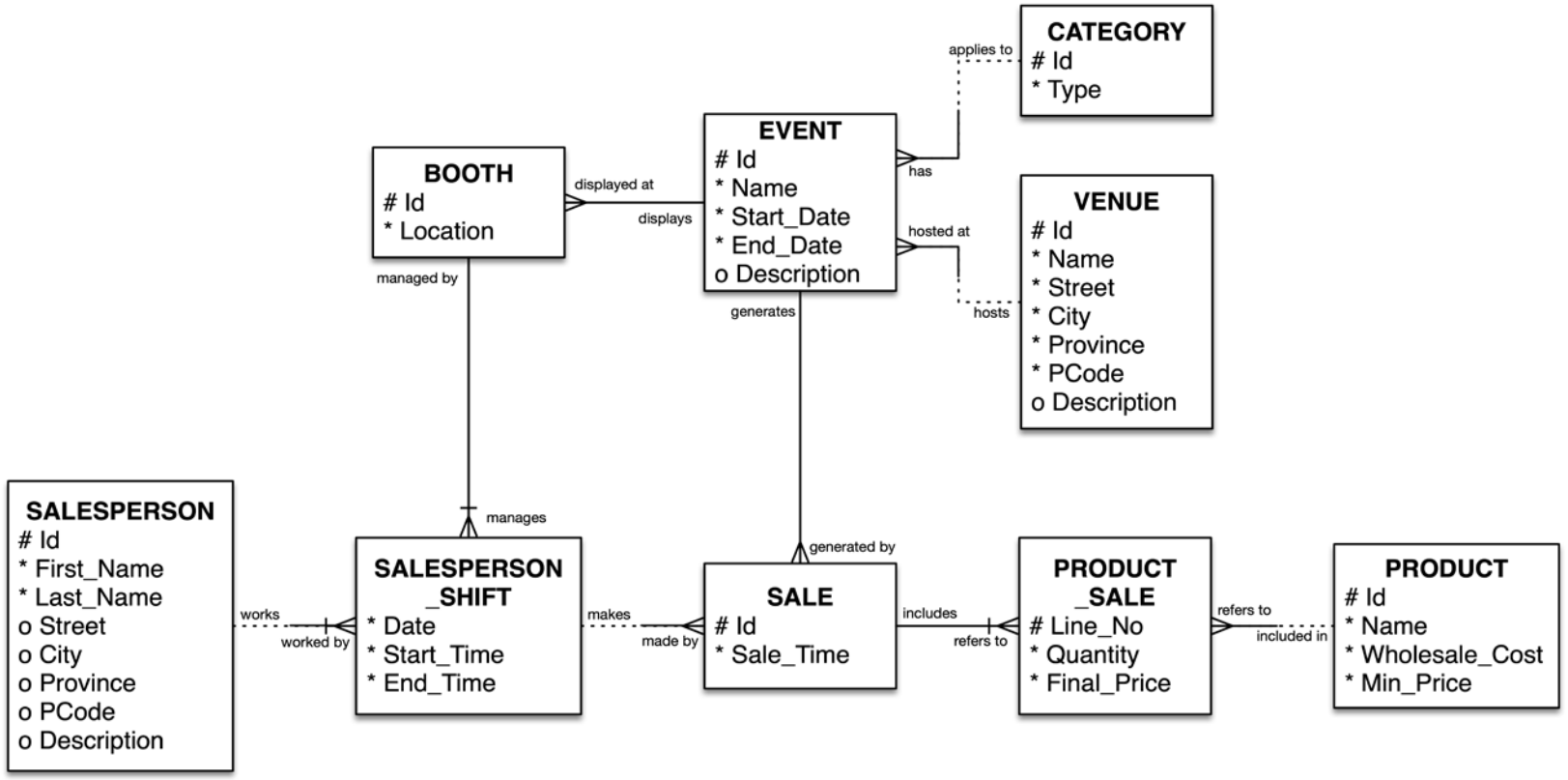
Whether SQL or phpMyAdmin was chosen to implement the data model, Corrigan needed to pay special attention to the unique identifiers in each table (i.e., captured as a *primary key constraint*) and to the relationships between entities (i.e., denoted using *secondary key constraints*). For example, from the data model (see Exhibit 1), the Event table would have to have both a primary key (called "id") and a secondary key to capture the relationship between Venue and Event (e.g., which might be called "venue.id").

A database would use these relationship constraints to ensure control of the data at all times. For example, whenever a new Event record was added to the database, it had to be assigned a unique Id; attempting to assign it an Id that already existed would violate its primary key constraint. Likewise, when entering that new Event, the related "venue.id" had to already exist in the Venue table; attempting to assign it a Venue Id that did not exist would violate its secondary key constraint. Therefore, the order of creating tables, and the relational constraints to other tables, were important and had to be carefully considered.

With these considerations in mind, Corrigan was ready to start creating the physical version of his database. He pulled out his laptop and got to work.

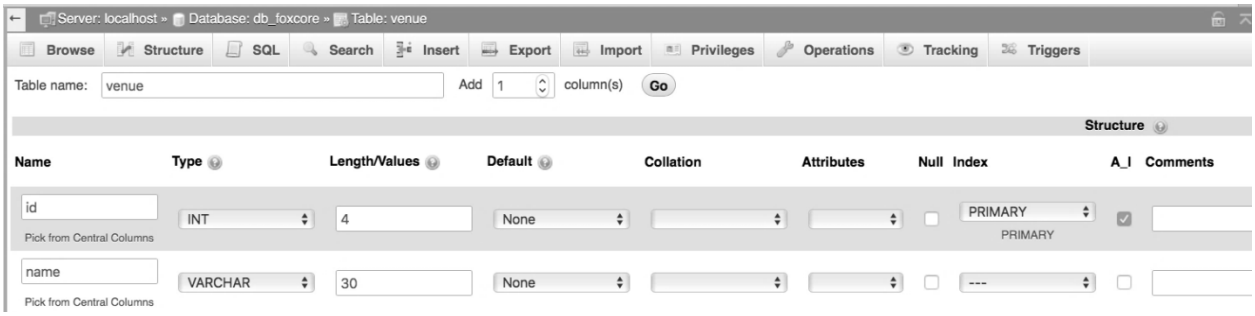
[3] phpMyAdmin is an administration tool for MySQL and MariaDB, included in the XAMPP software distribution, that provides an intuitive interface to easily interact with a database without extensive knowledge of SQL; "Bringing MySQL to the Web," phpMyAdmin, accessed October 17, 2018, www.phpmyadmin.net.

EXHIBIT 1: FOXCORE DATA MODEL



Source: Created by the case authors.

EXHIBIT 2: PHPMYADMIN



Source: Created by the case authors, with information from "phpMyAdmin GNU General Public License, Version 2," GNU Operating System, accessed October 17, 2018, www.gnu.org/licenses/old-licenses/gpl-2.0.en.html.