Designing a Database to Track Inventory and Purchase Data for a Library

Mike Kastellec

Valdosta State University

The Oconee County Libraries (OLC) has to keep track of information on over 100 pieces of electronics (computers, monitors, printers, etc.) Currently, a combination of Excel spreadsheets and an online tool from webjunction.org are used to record inventory information. The result is an unwieldy system that is time consuming to keep updated. The goal of this project is to create a database to record essential inventory data for both branches. It is hoped that, if the system performs well, it may one day be expanded to cover the other libraries in the Athens Regional Library System (ARLS). Most frequently, the system will be used to record or update the location and relation (e.g., which monitor is hooked up to which computer) of pieces of equipment. In its current incarnation, the database will only be accessed by the author (OCL's sole IT staffer), but in part that will entail translating requests from the Library's Manager and Regional Business Office staff into database queries.

Business Rules

General Rules

- A computer consists of a laptop or a desktop and monitor. Computers *within a given library* have unique names.
- Every piece of equipment has a unique serial number and a non-unique ARLS purchase code. Equipment belongs to a type (e.g., "Desktop") and a model (e.g., "Optiplex 740.") Models are made by one manufacturer (e.g., "Dell.")
- Each purchase has a unique ARLS purchase code. A purchase may include many different models but one purchase comes from only one vendor and one date. Orders from more than one vendor on the same day, or from the same vendor over many days, will each have their own purchase (and ARLS purchase code.)

- Vendors have a phone number and/or website address. Vendors may have a sales rep attached them, who may have their own phone number and/or email address.

- A library contains one or more locations. While of course an actual location only exists in one library, libraries may share names of locations (e.g., the Watkinsville library may have a location called "Adult" and so may the Bogart library.) Therefore a location in the database may be contained by one or more locations.

- When equipment is disposed of, it must be removed from the active inventory but a historical record of its information must be maintained. It is *not* necessary, however, to retain previous libraries or location if a piece of equipment is moved.

Relationships

- Zero to many pieces of equipment are located in each library and location. Every piece of equipment has one library and location.

- A piece of equipment belongs to one model. Each model may define one or many pieces of equipment.

- A model belongs to a type (of equipment.) A type contains one or more models.

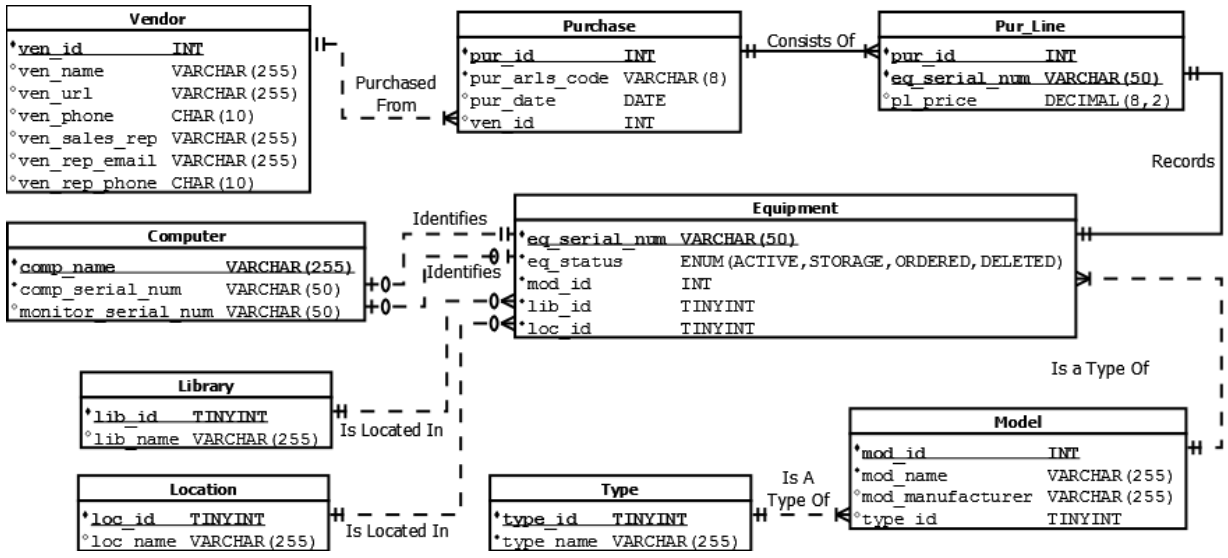
- Active computers (both desktops and laptops) are identified by a unique name. Active desktops are connected to a monitor.

- A desktop (computer) may be connected to zero or one monitors. A monitor may be connected to zero or one desktops.

- A purchase code identifies one or more pieces of equipment purchased at the same time. Each piece of equipment is identified by only one purchase code.

- Each purchase is purchased from one vendor. Vendors may supply equipment for many different purchases.

Entity Relationship Diagram



Data Dictionary

See Appendix.

Initializing the Database

Create Tables

```
CREATE TABLE Equipment (
eq_serial_num VARCHAR(50) NOT NULL PRIMARY KEY,
eq_status ENUM('ACTIVE','STORAGE','ORDERED','DELETED') NOT NULL,
eq_price DECIMAL(8,2),
mod_id int NOT NULL,
lib_id int NOT NULL,
loc_id int NOT NULL,
pur_arls_code VARCHAR(8) NOT NULL,
FOREIGN KEY (mod_id) REFERENCES Model(mod_id),
FOREIGN KEY (lib_id) REFERENCES Library(lib_id),
FOREIGN KEY (loc_id) REFERENCES Location(loc_id),
FOREIGN KEY (pur_arls_code) REFERENCES Purchase(pur_arls_code)
);

CREATE TABLE Vendor (
ven_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
ven_name VARCHAR(255),
ven_url VARCHAR(255),
```

```
ven_phone CHAR(10),
ven_sales_rep VARCHAR(255),
ven_rep_email VARCHAR(255),
ven_rep_phone CHAR(10)
);

CREATE TABLE Purchase (
pur_arls_code VARCHAR(8) NOT NULL PRIMARY KEY,
pur_date DATETIME,
ven_id INT,
FOREIGN KEY (ven_id) REFERENCES Vendor(ven_id)
);


CREATE TABLE Computer (
comp_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
comp_name VARCHAR(20) NOT NULL,
comp_serial_num VARCHAR(50) UNIQUE NOT NULL,
monitor_serial_num VARCHAR(50),
FOREIGN KEY (comp_serial_num) REFERENCES Equipment(eq_serial_num),
FOREIGN KEY (monitor_serial_num) REFERENCES Equipment(eq_serial_num)
);

CREATE TABLE Library (
lib_id TINYINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
lib_name VARCHAR(255)
);

CREATE TABLE Location (
loc_id TINYINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
loc_name VARCHAR(255)
);

CREATE TABLE Model (
mod_id int NOT NULL PRIMARY KEY AUTO_INCREMENT,
mod_name VARCHAR(255) NOT NULL,
mod_manufacturer VARCHAR(255),
type_id TINYINT,
FOREIGN KEY (type_id) REFERENCES Type(type_id)
);

CREATE TABLE Type (
type_id TINYINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
type_name VARCHAR(255)
);
```

Insert Data

```
INSERT INTO Library(lib_name)
      VALUES
            ('WAT'),
            ('BOG');

INSERT INTO Location(loc_name)
VALUES
```

```
      ('Adult'),
      ('Reference'),
      ('Kids'),
      ('Circ Desk'),
      ('Workroom');

INSERT INTO Type(type_name)
VALUES
      ('Desktop'),
      ('Laptop'),
      ('Monitor'),
      ('Printer'),
      ('Copier');

INSERT INTO
Vendor(ven_name,ven_url,ven_phone,ven_sales_rep,ven_rep_email,ven_rep_phone)
VALUES ('Dell','dell.com','8001234567','Lumumba
Weems','lweems@dell.com','8001234568');

INSERT INTO Vendor(ven_name,ven_url,ven_phone)
VALUES ('Gordon Docs','gordondocuments.com','7701234567');

INSERT INTO Vendor(ven_name,ven_url)
VALUES
      ('Amazon','amazon.com'),
      ('HP','HP.com/gov');

INSERT INTO Purchase(pur_arls_code,pur_date,ven_id)
VALUES
      ('S-08-02','2007-08-01','2'),
      ('S-02-04','2002-11-19','2'),
      ('S-05-12','2005-06-22','3'),
      ('S-08-06','2008-03-18','4'),
      ('S-07-01','2006-07-28','4'),
      ('S-03-04','2002-09-01','4'),
      ('S-07-05','2005-01-30','1'),
      ('S-08-01','2007-08-01','1'),
      ('S-08-08','2008-04-11','1'),
      ('S-08-09','2008-05-03','1');

INSERT INTO Model(mod_manufacturer,mod_name,type_id)
VALUES
      ('Konica Minolta','C252',5),
      ('Sharp','1000k',5),
      ('HP','lj2420dn',4),
      ('HP','clj2600n',4),
      ('HP','lj4050',4),
      ('HP','dc5000',1),
      ('HP','7550',3),
      ('Dell','Optiplex 740',1),
      ('Dell','M150',3),
      ('Dell','Optiplex 755',1),
      ('Dell','M170-DX',3),
      ('Dell','XRT26',3),
      ('Dell','Latitude D260',2);
```

```
-- broken into two inserts b/c VSU server won't accept such a long imput
INSERT INTO Equipment
(eq_serial_num,eq_status,eq_price,mod_id,lib_id,loc_id,pur_arls_code)
VALUES
('1278433','ACTIVE',3125.25,2,2,1,'S-02-04'),
('1278432','ACTIVE',3125.25,2,1,2,'S-02-04'),
('YW-987-100012-545','ACTIVE',NULL,7,2,3,'S-03-04'),
('YW-987-100012-546','ACTIVE',NULL,7,2,3,'S-03-04'),
('YW-987-100012-547','ACTIVE',NULL,7,1,3,'S-03-04'),
('YW-987-100012-548','ACTIVE',NULL,7,1,3,'S-03-04'),
('YW-987-100012-549','ACTIVE',NULL,7,1,3,'S-03-04'),
('YW-987-100012-550','ACTIVE',NULL,7,1,3,'S-03-04'),
('YW-987-100012-551','STORAGE',NULL,7,1,2,'S-03-04'),
('YW-987-100012-552','STORAGE',NULL,7,1,2,'S-03-04'),
('mv-010-432478','ACTIVE',989.85,6,1,3,'S-03-04'),
('mv-010-432479','ACTIVE',989.85,6,1,3,'S-03-04'),
('mv-010-432480','ACTIVE',989.85,6,1,3,'S-03-04'),
('mv-010-432481','ACTIVE',989.85,6,1,3,'S-03-04'),
('mv-010-432482','ACTIVE',989.85,6,2,3,'S-03-04'),
('mv-010-432483','ACTIVE',989.85,6,2,3,'S-03-04'),
('mv-010-432484','STORAGE',989.85,6,1,2,'S-03-04'),
('mv-010-432485','STORAGE',989.85,6,1,2,'S-03-04'),
('h1-8909-3213','ACTIVE',481.32,3,1,4,'S-05-12');

-- broken into two inserts b/c VSU server won't accept such a long imput
INSERT INTO Equipment
(eq_serial_num,eq_status,eq_price,mod_id,lib_id,loc_id,pur_arls_code)
VALUES
('321-dsa2-32190','ACTIVE',788.09,5,2,1,'S-07-01'),
('1338-QPZ-983S32','STORAGE',1089.98,13,1,5,'S-07-05'),
('1338-QPZ-983T98','STORAGE',1089.98,13,1,5,'S-07-05'),
('10303-OJP-13','ACTIVE',NULL,9,1,4,'S-07-05'),
('10303-OJP-14','ACTIVE',NULL,9,1,4,'S-07-05'),
('10303-OJP-11','ACTIVE',NULL,9,1,2,'S-07-05'),
('10303-OJP-12','ACTIVE',NULL,9,1,2,'S-07-05'),
('10303-OJP-10','ACTIVE',NULL,9,2,5,'S-07-05'),
('GX-321h890-F3','ACTIVE',770.33,8,1,4,'S-07-05'),
('GX-321h890-G3','ACTIVE',770.33,8,1,4,'S-07-05'),
('GX-321h890-D3','ACTIVE',770.33,8,1,2,'S-07-05'),
('GX-321h890-E3','ACTIVE',770.33,8,1,2,'S-07-05'),
('GX-321h890-H3','ACTIVE',770.33,8,2,5,'S-07-05'),
('21398-UUP-992','ACTIVE',NULL,11,1,2,'S-08-01'),
('21398-UUP-993','ACTIVE',NULL,11,1,2,'S-08-01'),
('21398-UUP-990','STORAGE',NULL,11,1,5,'S-08-01'),
('21398-UUP-991','ACTIVE',NULL,11,1,5,'S-08-01'),
('GX-923k987-025','ACTIVE',722.01,10,1,2,'S-08-01'),
('GX-923k987-026','ACTIVE',722.01,10,1,2,'S-08-01'),
('GX-923k987-023','ACTIVE',722.01,10,1,5,'S-08-01'),
('GX-923k987-024','ACTIVE',722.01,10,1,5,'S-08-01'),
('CLM1201','ACTIVE',5285.05,1,1,5,'S-08-02'),
('312-fds324-32132','ACTIVE',654.21,4,1,4,'S-08-06'),
('20123-UXP-490','ACTIVE',NULL,11,1,5,'S-08-08'),
('20123-UXP-491','ACTIVE',NULL,11,1,5,'S-08-08'),
```

```
('GX-923k987-343','ACTIVE',626.09,10,1,5,'S-08-08'),
('GX-923k987-344','ACTIVE',626.09,10,1,5,'S-08-08'),
('hidw395-32hj-ii8988','ACTIVE',399.99,12,1,5,'S-08-09');


INSERT INTO Computer(comp_name,comp_serial_num,monitor_serial_num)
VALUES
      ('CHL-01','mv-010-432482','YW-987-100012-545'),
      ('CHL-02','mv-010-432483','YW-987-100012-546'),
      ('CIRC-01','GX-321h890-H3','10303-OJP-10'),
      ('CHL-01','mv-010-432478','YW-987-100012-547'),
      ('CHL-02','mv-010-432479','YW-987-100012-548'),
      ('CHL-03','mv-010-432480','YW-987-100012-549'),
      ('CHL-04','mv-010-432481','YW-987-100012-550'),
      ('CIRC-01','GX-321h890-F3','10303-OJP-13'),
      ('CIRC-02','GX-321h890-G3','10303-OJP-14'),
      ('REF-01','mv-010-432484','YW-987-100012-551'),
      ('REF-02','mv-010-432485','YW-987-100012-552'),
      ('REF-01','GX-321h890-D3','10303-OJP-11'),
      ('REF-02','GX-321h890-E3','10303-OJP-12'),
      ('REF-03','GX-923k987-025','21398-UUP-992'),
      ('REF-04','GX-923k987-026','21398-UUP-993'),
      ('STAFF-CHL','GX-923k987-344','20123-UXP-491'),
      ('STAFF-IT','GX-923k987-023','hidw395-32hj-ii8988'),
      ('STAFF-MANAGER','GX-923k987-024','21398-UUP-991'),
      ('STAFF-REF','GX-923k987-343','20123-UXP-490'),
      ('LAPTOP-01','1338-QPZ-983T98',NULL),
      ('LAPTOP-02','1338-QPZ-983S32',NULL);
```

Sample SQL Statements

Selects

```
-- List the name, manufacturer, model, serial numer and location of all
active computers at the Watkinsville Library
SELECT
      C.comp_name AS Computer,
      M.mod_manufacturer AS Manufacturer,
      M.mod_name AS Model,
      C.comp_serial_num AS "S/N",
      Loc.loc_name AS Location
FROM
      Model AS M
      NATURAL JOIN
      (Equipment AS E
      INNER JOIN
      Computer AS C
            ON C.comp_serial_num=E.eq_serial_num
      NATURAL JOIN
      Location AS Loc)
WHERE
      E.lib_id = (SELECT Lib.lib_id FROM Library AS Lib WHERE lib_name="WAT")
      AND
      E.eq_status = "ACTIVE"
```

```
ORDER BY Loc.loc_name;

-- List the owning library, location, manufacturer, and model for all
printers and copiers
SELECT
      Lib.lib_name AS Library,
      Loc.loc_name AS Location,
      M.mod_manufacturer AS Manufacturer,
      M.mod_name AS Model
FROM
      Equipment AS E
      NATURAL JOIN
      Model AS M
      NATURAL JOIN
      Type AS T
      NATURAL JOIN
      Location AS Loc
      NATURAL JOIN
      Library AS Lib
WHERE
      T.type_name IN ('Printer','Copier')
      AND
      E.eq_status = "ACTIVE"
ORDER BY M.mod_name;

-- List the name, owning, library, manufacturer, and model of all public
computers
SELECT
      C.comp_name AS Computer,
      Lib.lib_name AS Library,
      Loc.loc_name AS Location,
      M.mod_manufacturer AS Manufacturer,
      M.mod_name AS Model
FROM
      (Equipment AS E
      NATURAL JOIN
      Model AS M
      NATURAL JOIN
      Location AS Loc
      NATURAL JOIN
      Library AS Lib)
      INNER JOIN
      Computer AS C
            ON C.comp_serial_num=E.eq_serial_num
WHERE
      Loc.loc_name NOT IN ('Circ','Workroom')
      AND
      E.eq_status = "ACTIVE"
ORDER BY Lib.lib_name,Loc.loc_name,C.comp_name;

-- Get detailed info on an active computer (and attached monitor), given only
the name and library
SELECT
      Loc.loc_name AS Location,
      C.comp_name AS Computer,
```

```
	CONCAT(M.mod_manufacturer," ",M.mod_name) AS Model,
	C.comp_serial_num AS "S/N",
	E.pur_arls_code AS "ARLS Code",
		CONCAT(
			(SELECT M.mod_manufacturer
				FROM
					Model AS M
					NATURAL JOIN
					(Equipment AS E
					INNER JOIN
					Computer AS C
						ON C.monitor_serial_num=E.eq_serial_num
					)
				WHERE
					C.comp_name = "STAFF-IT"
					AND
					E.lib_id = (SELECT Lib.lib_id FROM Library AS
Lib WHERE lib_name="WAT")
					AND
					E.eq_status = "ACTIVE"
			),
			" ",
			(SELECT M.mod_name
			FROM
			Model AS M
			NATURAL JOIN
				(Equipment AS E
				INNER JOIN
				Computer AS C
					ON C.monitor_serial_num=E.eq_serial_num)
			WHERE
				C.comp_name = "STAFF-IT"
				AND
				E.lib_id = (SELECT Lib.lib_id FROM Library AS Lib
WHERE lib_name="WAT")
				AND
				E.eq_status = "ACTIVE"
			)
			) AS "Monitor Model",
		C.monitor_serial_num AS "Monitor S/N",
			(SELECT E.pur_arls_code
				FROM
					Equipment AS E
					INNER JOIN
					Computer AS C
					ON C.monitor_serial_num=E.eq_serial_num
				WHERE
					C.comp_name = "STAFF-IT"
					AND
					E.lib_id = (SELECT Lib.lib_id FROM Library AS
Lib WHERE lib_name="WAT")
					AND
					E.eq_status = "ACTIVE"
				)	AS "Monitor ARLS Code"
FROM
```

```
      Model AS M
      NATURAL JOIN
      (Equipment AS E
      INNER JOIN
      Computer AS C
            ON C.comp_serial_num=E.eq_serial_num
      NATURAL JOIN
      Location AS Loc)
WHERE
      C.comp_name = "STAFF-IT"
      AND
      E.lib_id = (SELECT Lib.lib_id FROM Library AS Lib WHERE lib_name="WAT")
      AND
      E.eq_status = "ACTIVE";


-- List purchasing information for every unit purchased in Fiscal Year 2008
SELECT
      LEFT(P.pur_date,10) AS "Purchase Date",
      P.pur_arls_code AS "ARLS Purchase Code",
      V.ven_name AS Vendor,
      T.type_name AS Category,
      M.mod_name AS Item,
      E.eq_serial_num AS "S/N"
FROM
      Equipment AS E
      NATURAL JOIN
      Purchase AS P
      NATURAL JOIN
      Vendor AS V
      NATURAL JOIN
      Model as M
      NATURAL JOIN
      Type AS T
WHERE
      P.pur_date BETWEEN '2007-07-01' AND '2008-06-30'
ORDER BY P.pur_arls_code,P.pur_date

-- List summary of purchases made in FY2008, broken down by models ordered
SELECT
      LEFT(P.pur_date,10) AS "Purchase Date",
      P.pur_arls_code AS "ARLS Purchase Code",
      V.ven_name AS Vendor,
      M.mod_name AS Item,
      COUNT(M.mod_name) AS "Units Purchased",
      E.eq_price AS "Price per Unit",
      SUM(E.eq_price) AS "Total Price"
FROM
      Equipment AS E
      NATURAL JOIN
      Purchase AS P
      NATURAL JOIN
      Vendor AS V
      NATURAL JOIN
      Model as M
```

```
WHERE
      P.pur_date BETWEEN '2007-07-01' AND '2008-06-30'
GROUP BY M.mod_name,P.pur_arls_code
ORDER BY P.pur_arls_code,P.pur_date;

-- List total spending by year, rounded to nearest dollar
SELECT
      YEAR(P.pur_date) AS "Purchase Year",
      ROUND(SUM(E.eq_price),0) AS "Total Spending"
FROM
      Equipment AS E
      NATURAL JOIN
      Purchase AS P
      NATURAL JOIN
      Vendor AS V
GROUP BY YEAR(P.pur_date)
ORDER BY P.pur_date
```

Operations

```
-- Mark a piece of equipment deleted
UPDATE Equipment
SET eq_status = 'DELETED'
WHERE eq_serial_num = '21398-UUP-991'
LIMIT 1;

-- Move a monitor out of storage and assign it to a computer
UPDATE Equipment
SET eq_status = 'ACTIVE'
WHERE eq_serial_num = '21398-UUP-990'
LIMIT 1;

UPDATE Computer
SET monitor_serial_num = '21398-UUP-990'
WHERE comp_name = 'STAFF-MANAGER'
LIMIT 1;
```

Project Evaluation

During the initial design phase, it was difficult to visualize how best to handle attributes that would necessarily have duplicates: the ARLS purchase codes and locations names shared by multiple libraries. Creating the ERD (using Dia diagramming software) clarified things. It became apparent that I needed to add a table to record purchases, and then I realized a linking table needed to bridge purchases and equipment (or so I thought at the time). As for the problem

of similarly named locations, I considered alternative solutions but stuck with the design you see above.

While going through steps of normalization process, I realized that giving each purchase line a unique ID (and including it with the purchase ID and equipment's serial number as a composite primary key) was unnecessary—a composite key of just those two other attributes is enough to uniquely identify each purchase line. After fixing that, I realized I had forgotten to include the ability to store the price of a purchase. After some thought, I added it to the purchase line table, so as to be able to easily capture prices of equipment on a per unit basis.

One complicating factor that turned up again and again was maintaining historical data for deleted items. It wasn't necessary to track every computer a desktop or monitor was associated with over the course of its lifetime, but a record of serials numbers and purchase info for deleted items would be beneficial. My simple solution was to add "deleted" to the list of possible equipment statuses (the others being "active", in "storage", and "ordered" but not yet received.)

With the initial design completed, I began to insert data into the tables. I soon realized I hadn't been clear enough with my business rules—ARLS purchase codes are unique, so the purchase table didn't need a surrogate key. Then it became apparent that the table linking purchases and equipment was unnecessary—the relationship between the tables isn't M:N, it's 1:M. I deleted the linking table and moved the price and alrs code attributes to the equipment table, with arls code as a foreign key.

Since WAT and BOG can share computer names (e.g., they each could have a computer named "CIRC-01", I thought I needed to add the owning library's ID to the computer table as part of a composite primary key. However, I then found that deleted computers could easily

have duplicate computer names and owning libraries, so I didn't have a good primary key candidate between the two of them. I weighed using the desktop's serial number as the primary key but instead decided that a surrogate key was better solution in this case.

I ended up revising the design of the database many times, both during the initial design phase and then over and over again as I inserted data and wrote queries. Knowing that I'd have to include the create and insert statements that would build a working copy of my table for the purposes of this paper made the development process frustratingly slow—in a "normal" development environment, I would have been freed to alter tables and data on the fly (and using a GUI), which would have made testing and refining simpler and quicker. I also encountered a moment when writing a select query where I got "stuck" and began to doubt my SQL skills. After a couple hours of slogging I realized that I'd actually made a series of data entry errors to one table—my SQL code was correct all along. The first select query I wrote was by far the hardest to get working—once I grasped that the normalized design I'd come up with facilitated a series of natural joins to link almost any of the tables to each other, I proceeded without much difficulty. My SQL skills improved greatly with this applied practice.

The one area of my design that I'm least happy with is the computer table, which links computer names to serial numbers for the unit's desktop and monitor (or, for a laptop, just a single serial number). The solution I came up with works, as my queries demonstrate, but I consider it inelegant and it leads to very complex queries for anything related to computers' monitors (as my queries also demonstrate.) As I continue to develop this database for my workplace, this is definitely one area I will look closely at. That weakness aside, I'm pleased with the database—it's a fully functional system to track exactly what my library needs to track.

Appendix: Data Dictionary

Table Name	Attribute Name	Type	Optional	Unique	Contents	Format	PK/FK	FK Referenced Table
Computer	comp_id	int(11)	No	Yes	Computer ID code		PK	
	comp_name	varchar(20)	No	No	Computer name	Prefix-99		
	comp_serial_num	varchar(50)	No	Yes	Computer serial number		FK	Equipment(eq_serial_num)
	monitor_serial_num	varchar(50)	Yes	No	Monitor serial number		FK	Equipment(eq_serial_num)
Equipment	eq_serial_num	varchar(50)	No	Yes	Equipment serial number		PK	
	eq_status	enum('ACTIVE', 'STORAGE', 'ORDERED', 'DELETED')	No	No	Equipment status			
	eq_price	decimal(8,2)	Yes	No	Equipment price	99.99		
	mod_id	int(11)	No	No	Equipment model	9	FK	Model
	lib_id	int(11)	No	No	Equipment owning library	9	FK	Library
	loc_id	int(11)	No	No	Equipment location	9	FK	Location
	pur_arls_code	varchar(8)	No	No	Equipment ARLS code	X-99-99	FK	Purchase
Library	lib_id	tinyint(4)	No	Yes	Library ID code	9	PK	
	lib_name	varchar(255)	Yes	Yes	Library name			
Location	loc_id	tinyint(4)	No	Yes	Location ID code	9	PK	
	loc_name	varchar(255)	Yes	Yes	Location name	9		
Model	mod_id	int(11)	No	Yes	Model ID code	9	PK	
	mod_name	varchar(255)	No	Yes	Model name			
	mod_manufacturer	varchar(255)	Yes	No	Model manufacturer			
	type_id	tinyint(4)	Yes	No	Model Type	9	FK	Type
Purchase	pur_arls_code	varchar(8)	No	Yes	Purchase ARLS code	X-99-99	PK	
	pur_date	datetime	Yes	No	Purchase date	YYYY-MM-DD		
	ven_id	int(11)	Yes	No	Purchase vendor	9	FK	Vendor
Type	type_id	tinyint(4)	No	Yes	Type code	9	PK	
	type_name	varchar(255)	Yes	Yes	Type name			
Vendor	ven_id	int(11)	No	Yes	Vendor code	9	PK	
	ven_name	varchar(255)	Yes	No	Vendor name	Xxxxx		
	ven_url	varchar(255)	Yes	No	Vendor web address	www.xxx.com		
	ven_phone	char(10)	Yes	No	Vendor phone number	9999999999		
	ven_sales_rep	varchar(255)	Yes	No	Vendor sales rep name	First M. Last		
	ven_rep_email	varchar(255)	Yes	No	Vendor sales rep email address	xxxxx@x.xxx		
	ven_rep_phone	char(10)	Yes	No	Vendor sales rep phone number	9999999999		