**LAB 5**

**Chapter 10 exercises**

**Exercise 1**:

Use MySQL Workbench to create an EER diagram for a database that stores information about products. Each product must have a product name, description, and price. Each product must belong to one category. Each category must have a category name and description. Each category can include multiple products.

A diagram of a product

Description automatically generated

**Exercise 2**:

Use MySQL Workbench to create an EER diagram for a database that stores information about customers.

Each customer must have an email address, first name, and last name. Each customer can have two or more addresses. Each customer can have a default billing address and a default shipping address. Each address must have a street address, city, state, postal code, and country. Each country name should be stored in one place only. In other words, you shouldn’t store the name of the country, which may be many characters, in the address.

A screenshot of a computer

Description automatically generated

**Exercise 3**:

Use MySQL Workbench to create an EER diagram for a database that tracks the memberships for an association and for the groups within the association. Each member must have an email address, first name, and last name. Each member can belong to any number of groups. Each group must have a name. Each group can have any number of members.

A close-up of a group

Description automatically generated

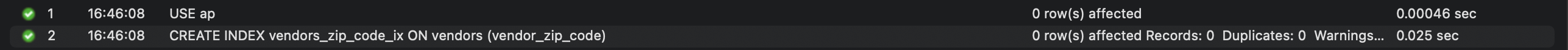
**Chapter 11 exercises**

**Exercise 1**:

Write a script that adds an index to the AP database for the zip code field in the Vendors table.

USE ap;

CREATE INDEX vendors\_zip\_code\_ix ON vendors (vendor\_zip\_code);



**Exercise 2**:

Write a script that contains the CREATE TABLE statements needed to imple-ment the following design in the EX database:

A diagram of a company

Description automatically generated

These tables provide for members of an association, and each member can be registered in one or more committees within the association. The member\_id and committee\_id columns are the primary keys of the Members and Committees tables, and these columns are foreign keys in the Members\_Committees table. Include any constraints or default values that you think are necessary. Include statements to drop the tables if they already exist.

USE ex;

DROP TABLE IF EXISTS members\_commitees;

DROP TABLE IF EXISTS members;

DROP TABLE IF EXISTS committees;

CREATE TABLE members

(

member\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

address VARCHAR(50) NOT NULL,

city VARCHAR(25) NOT NULL,

state CHAR(2),

phone VARCHAR(20)

);

CREATE TABLE committees

(

committee\_id INT PRIMARY KEY AUTO\_INCREMENT,

committee\_name VARCHAR(50) NOT NULL

);

CREATE TABLE members\_committees

(

member\_id INT NOT NULL,

committee\_id INT NOT NULL,

CONSTRAINT members\_committees\_fk\_members FOREIGN KEY (member\_id)

REFERENCES members (member\_id),

CONSTRAINT members\_committes\_fk\_committees FOREIGN KEY (committee\_id)

REFERENCES committees (committee\_id)

);

A screen shot of a computer

Description automatically generated

**Exercise 3**:

Write INSERT statements that add rows to the tables that are created in exercise 2.

Add two rows to the Members table for the first two member IDs.

Add two rows to the Committees table for the first two committee IDs.

Add three rows to the Members\_Committees table: one row for member 1 and committee 2; one for member 2 and committee 1; and one for member 2 and committee 2.

Write a SELECT statement that joins the three tables and retrieves the committee name, member last name, and member first name.

Sort the results by the committee name, member last name, and member first name.

Add one column for the payment date.

USE ex;

INSERT INTO members

VALUES (DEFAULT, 'John', 'Smith', '334 Valencia St.', 'San Francisco', 'CA', '415-942-1901');

INSERT INTO members

VALUES (DEFAULT, 'Jane', 'Doe', '872 Chetwood St.', 'Oakland', 'CA', '510-123-4567');

INSERT INTO committees

VALUES (DEFAULT, 'Book Drive');

INSERT INTO committees

VALUES (DEFAULT, 'Bicycle Coalition');

INSERT INTO members\_committees

VALUES (1, 2);

INSERT INTO members\_committees

VALUES (2, 1);

INSERT INTO members\_committees

VALUES (2, 2);

SELECT c.committee\_name, m.last\_name, m.first\_name

FROM committees c

JOIN members\_committees mc

ON c.committee\_id = mc.committee\_id

JOIN members m

ON mc.member\_id = m.member\_id

ORDER BY c.committee\_name, m.last\_name, m.first\_name;

A screenshot of a computer

Description automatically generated

**A black screen with white text

Description automatically generated**

**Exercise 4**:

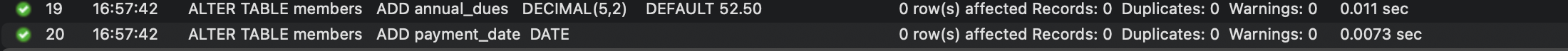
Write an ALTER TABLE statement that adds two new columns to the Members table created in exercise 2. Add one column for annual dues that provides for three digits to the left of the decimal point and two to the right. This column should have a default value of 52.50. Add one column for the payment date.

ALTER TABLE members

ADD annual\_dues DECIMAL(5,2) DEFAULT 52.50;

ALTER TABLE members

ADD payment\_date DATE;



**Exercise 5**:

Write an ALTER TABLE statement that modifies the Committees table created in exercise 2 so the committee name in each row has to be unique. Then, use an INSERT statement to attempt to insert a duplicate name. This statement should fail due to the unique constraint.

ALTER TABLE committees

MODIFY committee\_name VARCHAR(50) NOT NULL UNIQUE;

INSERT INTO committees (committee\_name)

VALUES ('Book Drive');

