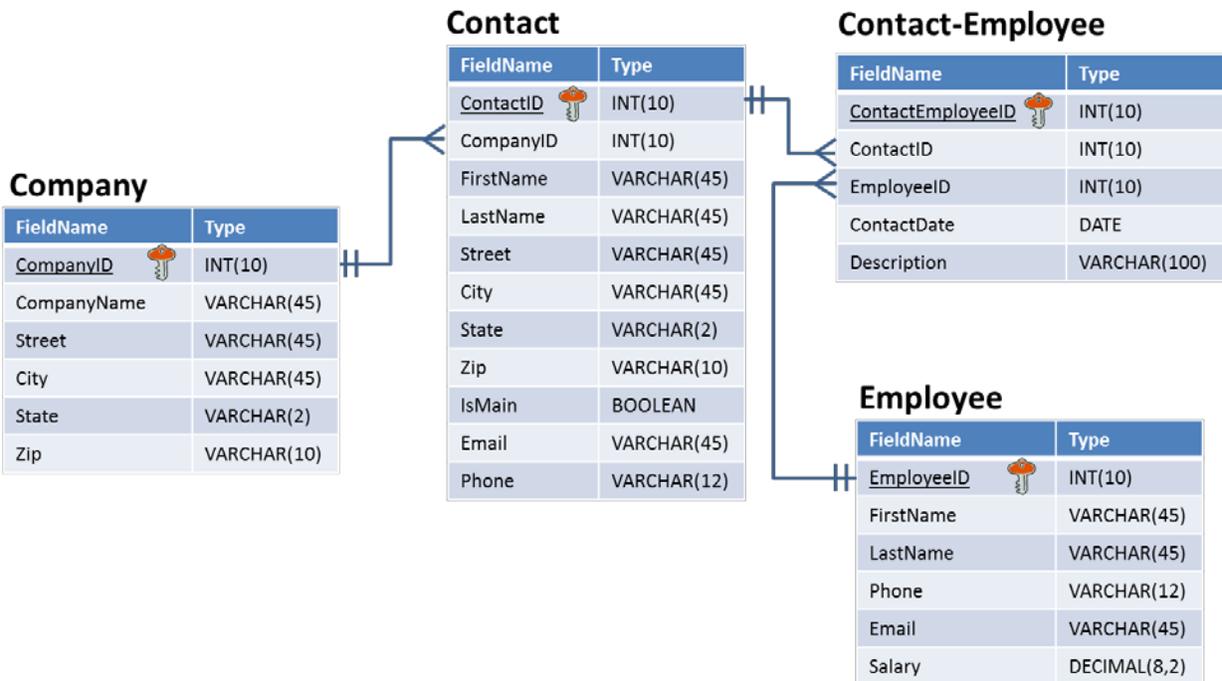


In-Class Exercise: SQL #2

Putting Information into a Database

In this exercise, you will begin to build a database for a simple contact management system for a marketing organization called MarketCo. You'll start with a blank schema (no tables, no data); you will be creating the tables and adding data. The exercise will walk you through creating the Company table. You'll finish the database in the next homework assignment.

Here is the schema for the database (the schema is called "m4ws"):



Remember, you will need to log in using your own account. You have your own copy of ws, named specifically according to your username.

So if you are logging in as user m45, the schema name for you will be **m45ws**.

Essentially, you have companies, contacts, and employees. The database will track key data about their sales contacts, the companies for which those employees work, and when an employee gets in touch with a contact. The database will also store basic information about an employee.

The Contact-Employee table implements the many-to-many relationship between employee and contact (an employee can contact many contacts, and a contact can be contacted by many employees). It represents a "contact event" – when an employee communicates with one of its customers (contact).

Here is the rest of the metadata, describing what each field means:

Contact

FieldName	Description
<u>ContactID</u>	Unique identifier for a contact
CompanyID	Foreign key identifier for the contact's company (from the Company table)
FirstName	Contact's first name
LastName	Contact's last name
Street	Street address for contact
City	Contact's city
State	Contact's state (abbreviation)
Zip	Contact's Zip code (9-digit format)
IsMain	1 if this is the main contact for the company; 0 if not
Email	Contact's email address
Phone	Contact's phone number (10-digit format)

Company

FieldName	Description
<u>CompanyID</u>	Unique identifier for a company
CompanyName	The name of the company
Street	Street address of the company
City	Company's city
State	Company's state (abbreviation)
Zip	Contact's Zip code (9-digit format)

Contact-Employee

FieldName	Description
<u>ContactEmployeeID</u>	Unique identifier for the table
ContactID	Foreign key identifier for the contact (from the Contact table)
EmployeeID	Foreign key identifier for the employee (from the Employee table)
ContactDate	Date of contact
Description	Brief explanation of the contact

Employee

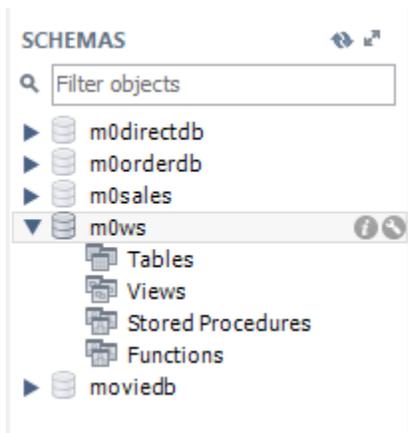
FieldName	Description
<u>EmployeeID</u>	Unique identifier for an employee
FirstName	Employee's first name
LastName	Employee's last name
Phone	Employee's phone number (10-digit format)
Email	Employee's email address

Part 1: Creating the Company Table

We need to build the database from your empty mxws schema on our MySQL server. This means there are no tables in the schema yet to put your data. But you can use the diagram on the previous page as a guide – that's where we're going to end up by the time you're finished with this exercise and the next homework assignment.

Follow these steps:

- 1) Log in to the MySQL server through MySQL Workbench using your MySQL ID and password.
- 2) Double-click on the mxws schema in the SCHEMAS pane on the left side of the screen. You'll see the categories of database components:



- 3) Now double-click on Tables. You'll see that although the arrow disappears, nothing else happens. This means there's nothing to show. There are no tables yet in this schema.
- 4) Let's create the Company table. Type this CREATE TABLE statement in the SQL File window:

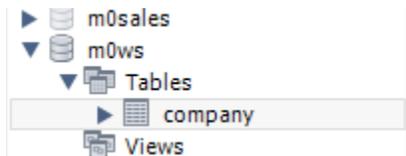
```
CREATE TABLE mxws.Company (
  CompanyID INT(10) NOT NULL,
  CompanyName VARCHAR(45) NULL,
  Street VARCHAR(45) NULL,
  City VARCHAR(45) NULL,
  State VARCHAR(2) NULL,
  PRIMARY KEY (CompanyID));
```

(Make sure that the **BOLDED** words appear in blue. This means that MySQL Workbench recognizes them as SQL keywords and that you didn't make a typo! Also make sure you're substituting the "x" in mxws for your own My SQL ID number!)

- 5) Execute the statement by clicking on the lightning bolt icon (the same thing you've been doing to execute SQL queries in the previous exercises). It doesn't look like much has happened, but at the bottom of the screen you'll see:

Output			
Action Output			
	Time	Action	Message
✓	1 16:27:24	CREATE TABLE m0ws.Company (CompanyID INT(10...	0 row(s) affected Duration / Fetch: 0.078 sec

This means the statement executed successfully. Now click on refresh symbol () at the top of the Object Browser panel. Then double-click on "Tables" and you'll see the company table:



- 6) If you expand the company table and then the “Columns” category you’ll see all the fields. Except zip code – we forgot the “zip” field. So let’s fix that. Type the following statement:

```
ALTER TABLE mxws.Company  
ADD COLUMN Zip VARCHAR(10);
```

Execute the statement. Now refresh () the Object Browser again and Zip will appear.

- 7) Now we think about it a little more and we decide that 45 characters for City is not necessary. Let’s change that:

```
ALTER TABLE mxws.Company  
CHANGE COLUMN City  
City VARCHAR(15);
```

Execute the statement. Now refresh () the Object Browser and click once on Columns under Tables/company. Table metadata is displayed in the Information window below the Object Browser. Note that City is now data type VARCHAR(15).

Part 2: On Your Own

- 1) You just found out that there is a city in Pennsylvania called Kleinfeltersville. So now you’ve reconsidered and want City to allow 45 characters. Modify the Company table and record the appropriate SQL statement below:
- 2) Add another column to the Company table called “Email” with data type VARCHAR(30). Write the SQL statement you used below:
- 3) Now write (and execute) the SQL statement to remove that column from the table:

Part 3: Adding, deleting, and modifying records

- 1) Let's add two companies to the table:

CompanyID	CompanyName	Street	City	State	Zip
101	Comcast	1701 JFK Blvd.	Philadelphia	PA	19103
102	Verizon	140 West St.	New York	NY	10007

Execute the following statements:

```
INSERT INTO mxws.Company
```

```
(CompanyID, CompanyName, Street, City, State, Zip)
```

```
VALUES (101,'Comcast','1701 JFK Blvd.','Philadelphia','PA','19103');
```

```
INSERT INTO mxws.Company
```

```
(CompanyID, CompanyName, Street, City, State, Zip)
```

```
VALUES (102,'Verizon','140 West St.','New York','NY','10007');
```

And then check to make sure they've been added with a SELECT query:

```
SELECT * FROM mxws.Company;
```

- 2) Now let's change the CompanyName of Verizon to Verizon Communications and the Zip code to 10007-1111:

```
UPDATE mxws.Company
```

```
SET CompanyName='Verizon Communications', Zip='10007-1111'
```

```
WHERE CompanyID=102;
```

Note that we use the primary key to identify the record!

Use that SELECT query to make sure your changes were successful.

- 3) Now say we don't want Comcast in there at all. Delete the record:

```
DELETE FROM mxws.Company WHERE CompanyID=101;
```

And use that SELECT query to verify the record is gone.

Part 4: On Your Own

1) Write and execute the statement(s) to add two more companies to the Company table:

CompanyID	CompanyName	Street	City	State	Zip
103	Independence Blue Cr.	1901 Market St.	Philadelphia	PA	19103
104	Aramark	1101 Chestnut St.	Philadelphia	PA	19107

2) Write and execute the statement(s) to change the Street for Aramark to 1101 Market St.

3) Write and execute the statement(s) to change the CompanyName for the record with CompanyID 103 to "Independence Blue Cross" and its Street to "1905 Market St."

4) Write and execute the statement(s) to delete Aramark's record from the table: