



INFORMATICS
INSTITUTE OF
TECHNOLOGY

UNIVERSITY OF
WESTMINSTER™

Data Base System

Course Work Report

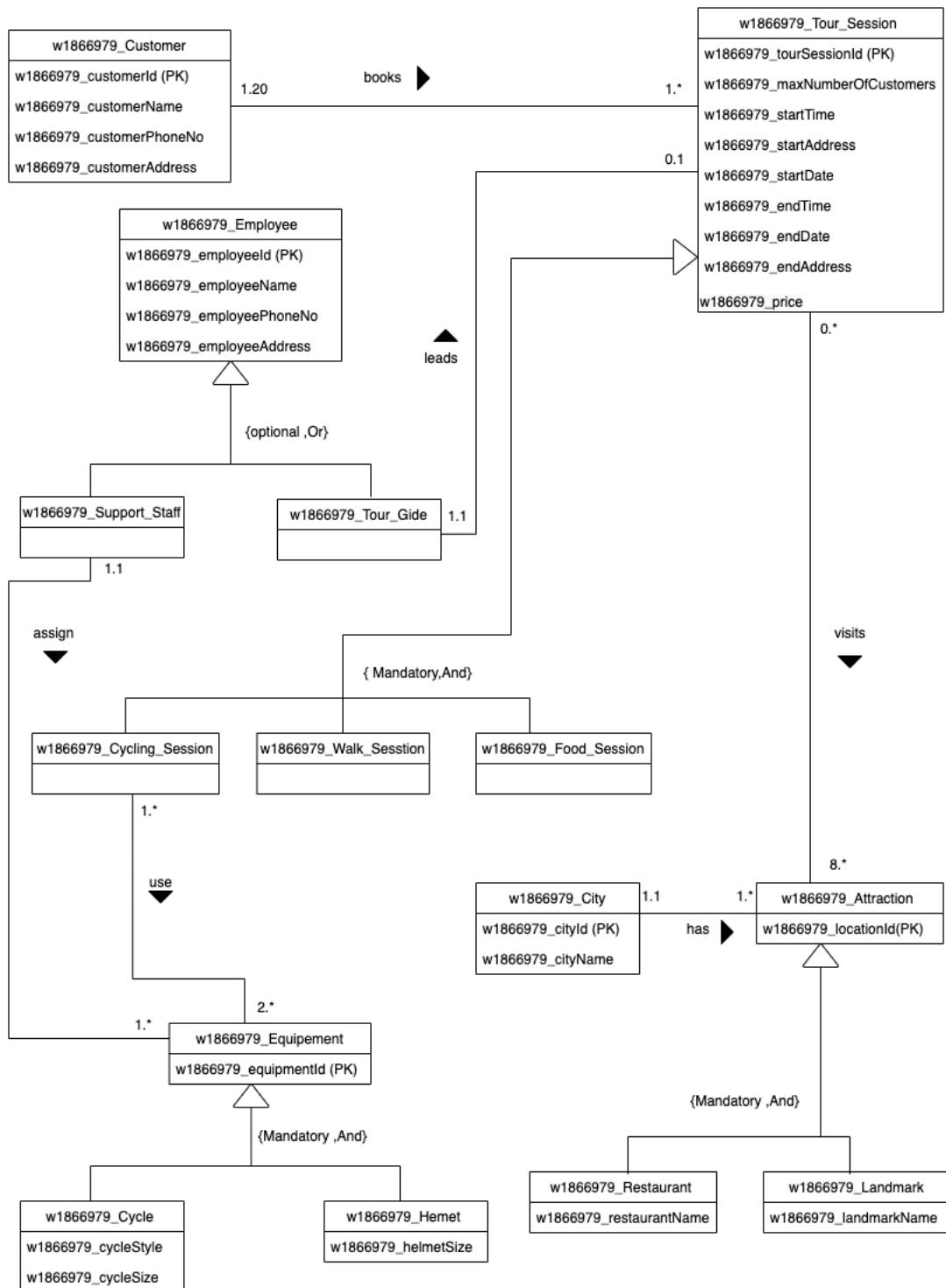
Parts A and B

Name: R.M.S.J. Bandara
UOW Number: w1866979
IIT Number: 20200649
Group B

Table of Contents

<i>Entity Relationship Diagram</i>	2
<i>Data Dictionaries</i>	3
2.1 Entities	3
2.2 General Entity	4
2.3 Relationships & Multiplicities	5
2.4 Attributes & Primary Keys	6
3. <i>BoilHeater Database part B</i>	8
3.1 Logical Entity Relationship Diagram	8
3.2 MySQL Screenshots	9
3.2.1 Create Database	9
3.2.2 Create customer Table	10
3.2.3 Update primary key	11
3.2.4 Create oneoff_payment Table	12
3.2.5 Populating customer Table	13
3.2.6 Populating oneoff_Payment Table	15
3.2.7 Update foreign key	17
3.2.8 Query	18
4 <i>MySQL and MongoDB</i>	19
Reference	20

Entity Relationship Diagram



Data Dictionaries

2.1 Entities

Entity Name	Brief Description
w1866979_Customer	Describing customers of tourmato
w1866979_City	Describing cities provide in tourmato service
w1866979_Attraction	Describing places in tour session
w1866979_Tour_Session	Describing Tour session information
w1866979_Employee	Describing employees in tourmato
w1866979_Support_Staff	Describing employees to maintain the equipment
w1866979_Tour_Guide	Describing tour guide tour session to attraction in tourmato
w1866979_Cycling_Session	Describing tour session by cycling to attraction places
w1866979_Walk_Session	Describing tour session by walking to attraction places
w1866979_Food_Session	Describe the tour session visiting different restaurants
w1866979_Restaurant	Describing restaurants in tour session
w1866979_Landmark	Describing the land marks in the city
w1866979_Equipement	Describing equipment in the tour session
w1866979_Cycle	Describing transport equipment in tour session
w1866979_Hemet	Describing the equipment while using the cycle

2.2 General Entity

General Entity	Specialised Entity	Brief Explanation
w1866979_Tour_Session	w1866979_Cycling_Session	Cycle used move to attractions in the tour section
	w1866979_Walk_Sesstion	Walking to attraction places in the tour session
	w1866979_Food_Session	Visit a different restaurant and taste different foods in the tour session
w1866979_Employee	w1866979_Support_Staff	Crew members maintain equipment
	w1866979_Tour_Gide	Crew members guide the customers in tour session
w1866979_Attraction	w1866979_Restaurant	Restaurants in tour session
	w1866979_Landmark	Landmarks in tour session
w1866979_Equipement	w1866979_Cycle	Cycle use to cycle session
	w1866979_Hemet	Equipment use for protection while riding cycle

2.3 Relationships & Multiplicities

Entity Name	Multiplicity	Relationship	multiplicity	Entity name	Brief Justifications for the Multiplicity
w1866979_Customer	1.*	books	1.*	w1866979_Tour Session	One Customer need to book at least one tour session
					One Customer can book many tour Sessions
					A Tour session has at least one customer
					A Tour session may have 20 maximum customers
w1866979_Tour Session	0.*	visits	8.*	w1866979_Attraction	A Tour session visits a minimum of 8 attraction
					A Tour session can visit many attraction
					One attraction may not visit tour session
					One attraction has many tour session
w1866979_Tour Guide	1.1	leads	0.1	w1866979_Tour Session	A Tour guide may not leads a tour session
					A Tour guide may lead a one tour session
					One Tour session must lead one tour Guide
					One Tour session can lead by only one tour guide
w1866979_Support Staff	1.1	assign	1.*	w1866979_Equipment	One Support staff may not be assign equipment
					One Support staff can be assigned to many equipment
					One Equipment must be assigned to one person
					One Equipment assign to at least one person
w1866979_Cycling Session	1.*	use	2.*	w1866979_Equipment	One Cycle session must use at least two equipment
					One Cycle session can use many equipment
					One Equipment may use to one cycle session
					One Equipment can be use many cycling session
w1866979_City	1.1	has	1.*	w1866979_Attraction	One City has at least one attraction
					One City has many attractions
					One Attraction must have located one city
					One Attraction may have one city

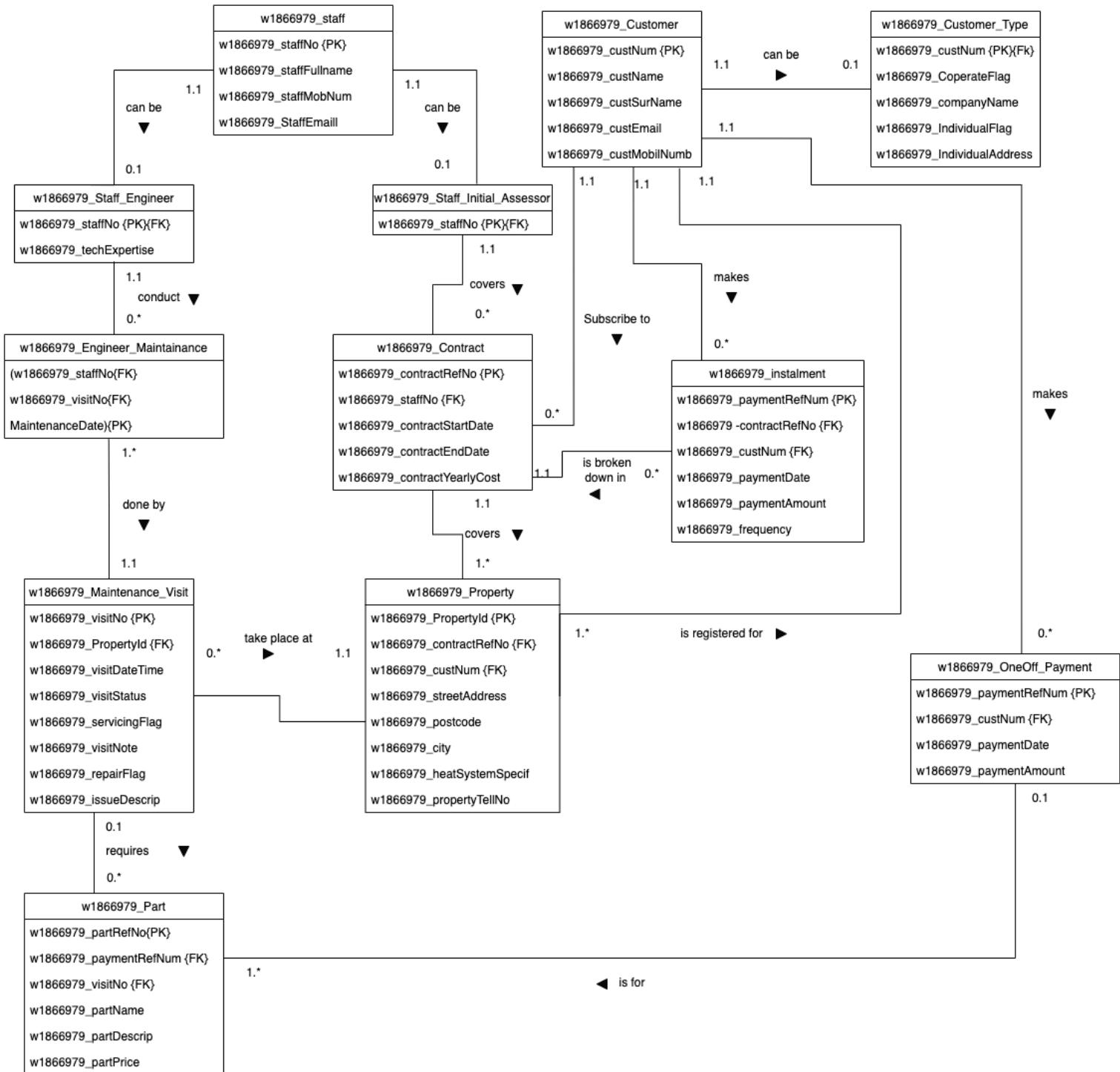
2.4 Attributes & Primary Keys

Entity Name	Attribute of this entity	Brief explanation
w1866979_Customer	w1866979_customerId (PK)	Uniquely identify customer
	w1866979_customerName	Customers full name
	w1866979_customerPhoneNo	Customer phone number
	w1866979_customerAddress	Customer address
w1866979_Employee	w1866979_employeeId (PK)	Uniquely identify employee
	w1866979_employeeName	Employee name
	w1866979_employeePhoneNo	Employee phone number
	w1866979_employeeAddress	Employee address
w1866979_Tour_Session	w1866979_tourSessionId (PK)	Uniquely identify tour session id
	w1866979_maxNumberOfCustomers	Maximum number of customers for tour session
	w1866979_startTime	Tour session start time
	w1866979_startAddress	Tour session start address
	w1866979_startDate	Tour session start date
	w1866979_endTime	Tour session end time
	w1866979_endDate	Tour session end date
	w1866979_endAddress	Tour session end Address
	w1866979_price	Tour session price
w1866979_City	w1866979_cityId (PK)	Uniquely identify city
	w1866979_cityName	City name
w1866979_Attraction	w1866979_locationId (PK)	Uniquely identify attraction
w1866979_Restaurant	w1866979_restaurantName	Name of the restaurant
w1866979_Landmark	w1866979_landmarkName	Name of the landmark

w1866979_Equipement	w1866979_equipmentId (PK)	Uniquely identify equipment
w1866979_Cycle	w1866979_cycleStyle	Cycle style
	w1866979_cycleSize	Cycle size
w1866979_Hemet	w1866979_helmetSize	Helmet size
w1866979_Cycling_Session		
w1866979_Walk_Sesstion		
w1866979_Food_Session		
w1866979_Support_Staff		
w1866979_Tour_Gide		

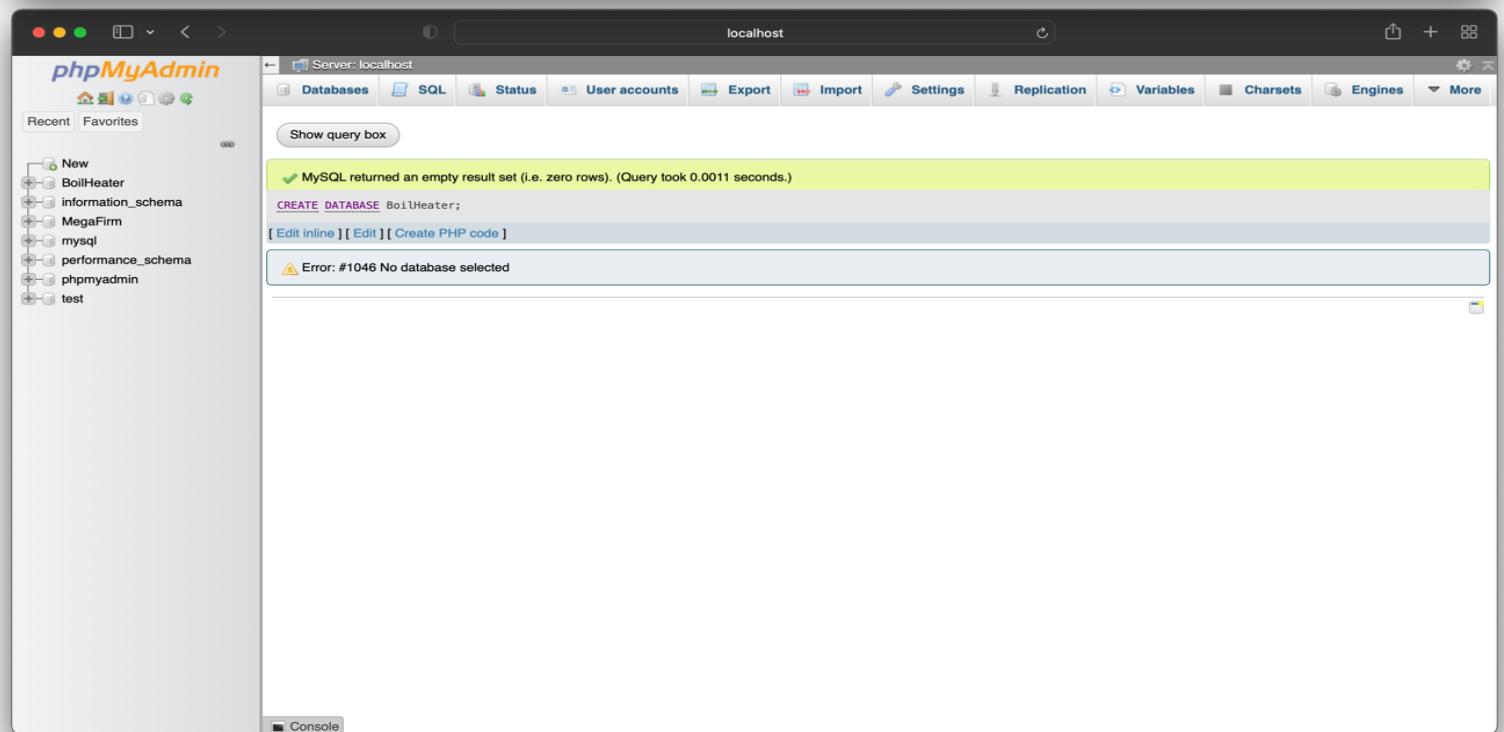
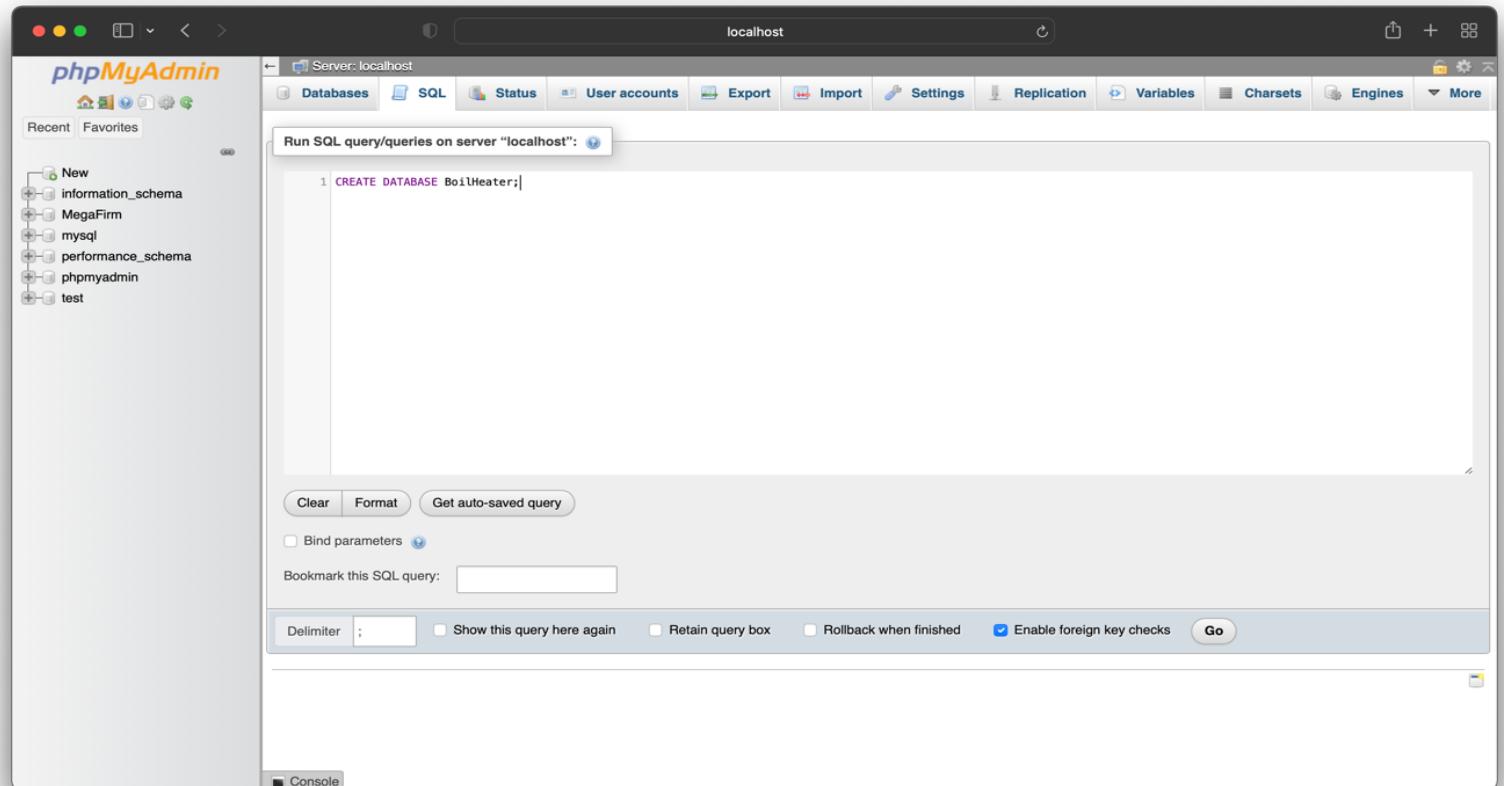
3. BoilHeater Database part B

3.1 Logical Entity Relationship Diagram

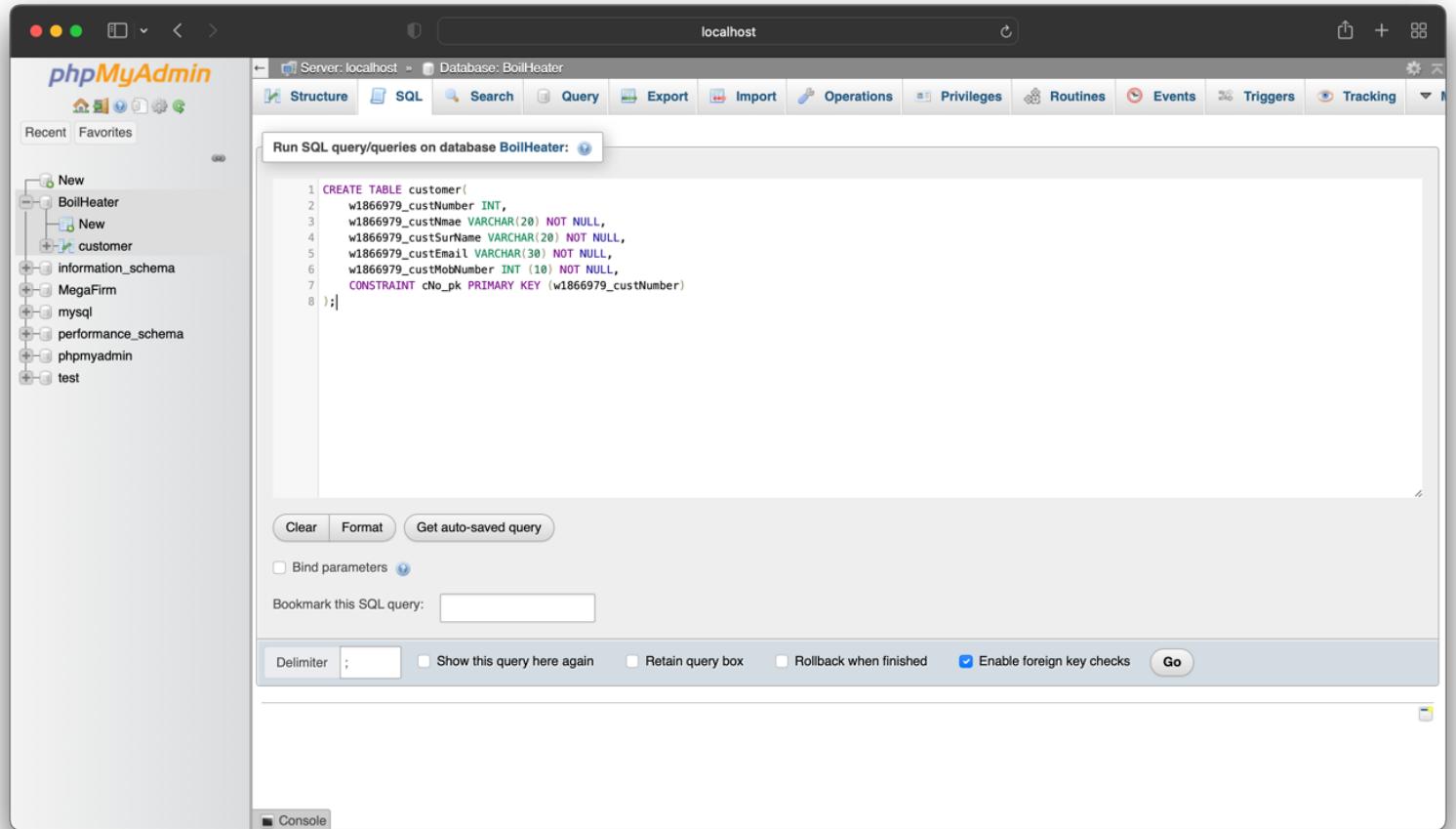


3.2 MySQL Screenshots

3.2.1 Create Database



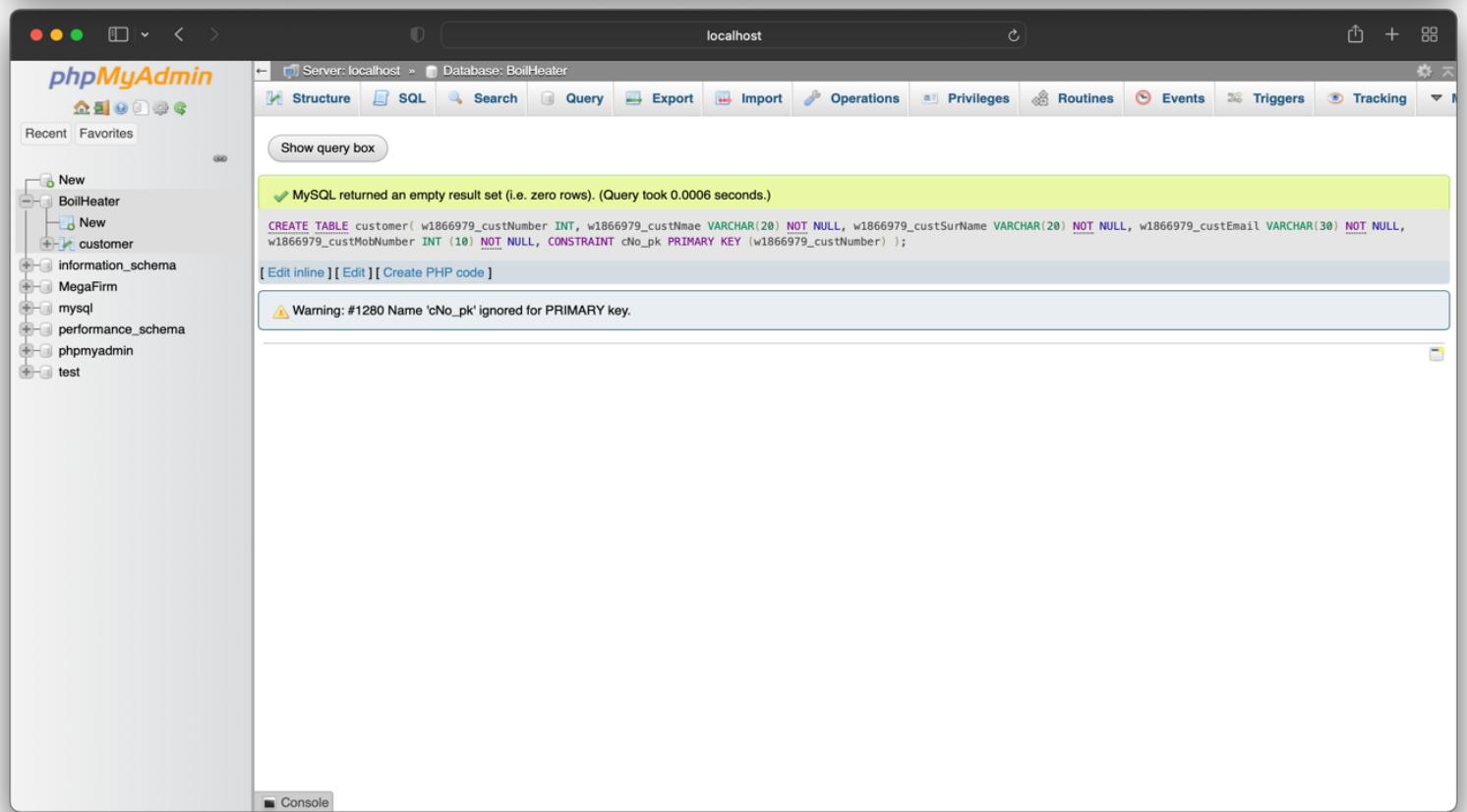
3.2.2 Create customer Table



The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The left sidebar shows the database structure with a 'customer' table selected. The main area is the 'SQL' tab, which contains the following SQL code:

```
1 CREATE TABLE customer(
2     w1866979_custNumber INT,
3     w1866979_custName VARCHAR(20) NOT NULL,
4     w1866979_custSurName VARCHAR(20) NOT NULL,
5     w1866979_custEmail VARCHAR(30) NOT NULL,
6     w1866979_custMobNumber INT (10) NOT NULL,
7     CONSTRAINT cNo_pk PRIMARY KEY (w1866979_custNumber)
8 );
```

Below the code, there are buttons for 'Clear', 'Format', and 'Get auto-saved query'. There is also a checkbox for 'Bind parameters' and a text input for 'Bookmark this SQL query'. At the bottom, there are options for 'Delimiter' (set to ':'), checkboxes for 'Show this query here again', 'Retain query box', and 'Rollback when finished', and a checked checkbox for 'Enable foreign key checks'. A 'Go' button is also present.

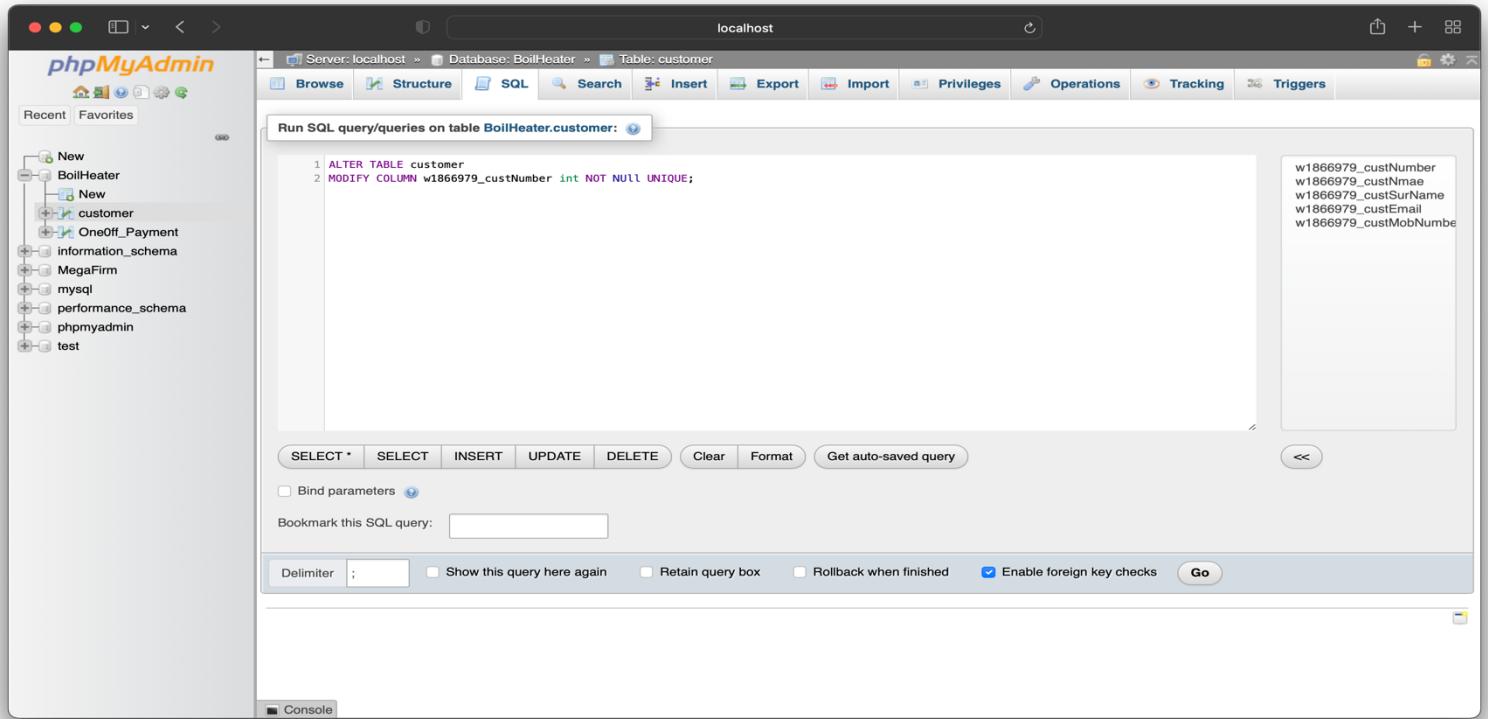


The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The left sidebar shows the database structure with a 'customer' table selected. The main area is the 'SQL' tab, which displays the results of the previously run SQL query. A green message bar at the top states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)'. Below this, the executed SQL code is shown:

```
CREATE TABLE customer( w1866979_custNumber INT, w1866979_custName VARCHAR(20) NOT NULL, w1866979_custSurName VARCHAR(20) NOT NULL, w1866979_custEmail VARCHAR(30) NOT NULL, w1866979_custMobNumber INT (10) NOT NULL, CONSTRAINT cNo_pk PRIMARY KEY (w1866979_custNumber) );
```

Below the code, there are buttons for '[Edit inline]', '[Edit]', and '[Create PHP code]'. A warning message is displayed in a yellow box: 'Warning: #1280 Name 'cNo_pk' ignored for PRIMARY key.' At the bottom, there is a 'Console' button.

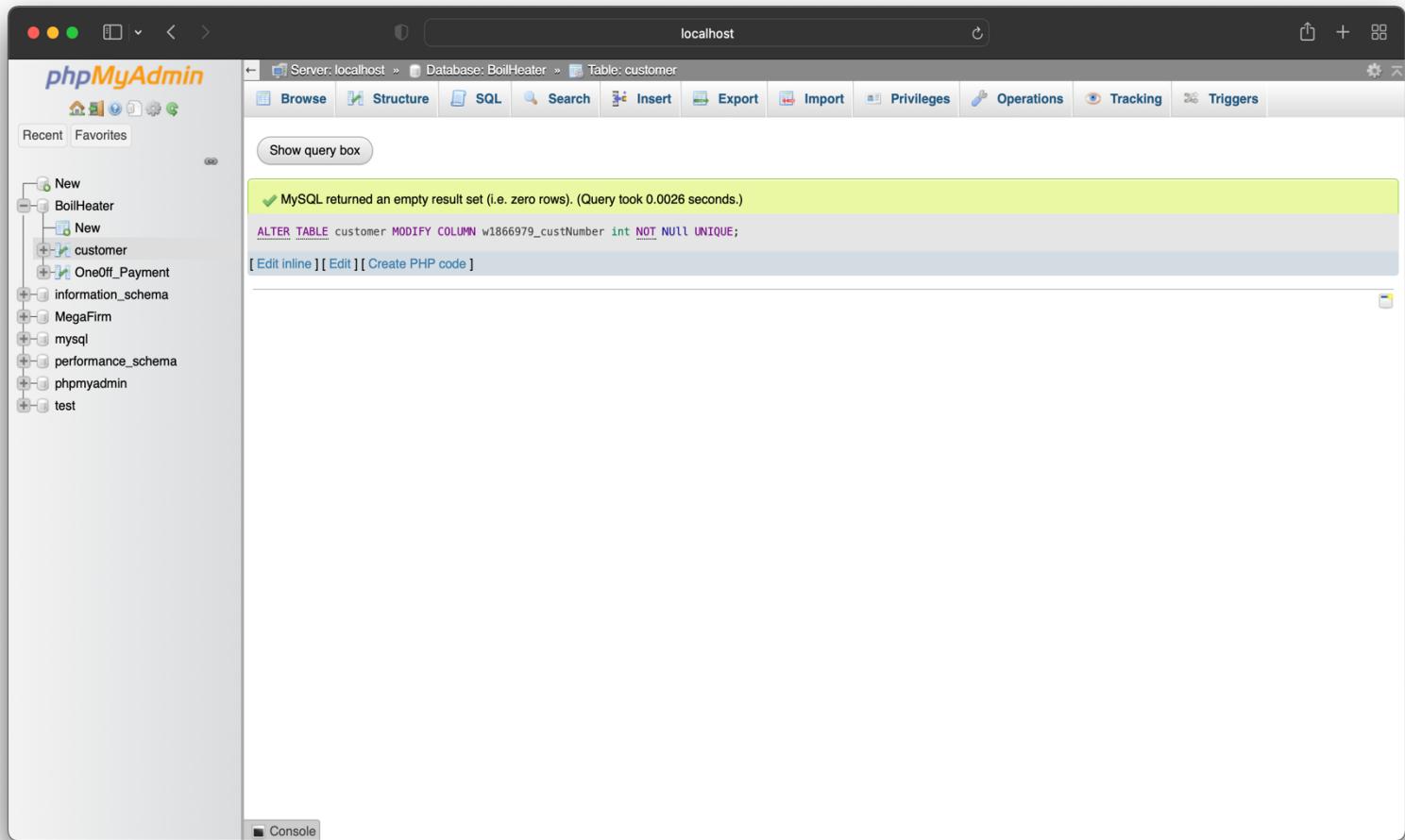
3.2.3 Update primary key



The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar shows the database structure with a 'customer' table selected. The main area is titled 'Run SQL query/queries on table BoilHeater.customer:'. It contains the following SQL code:

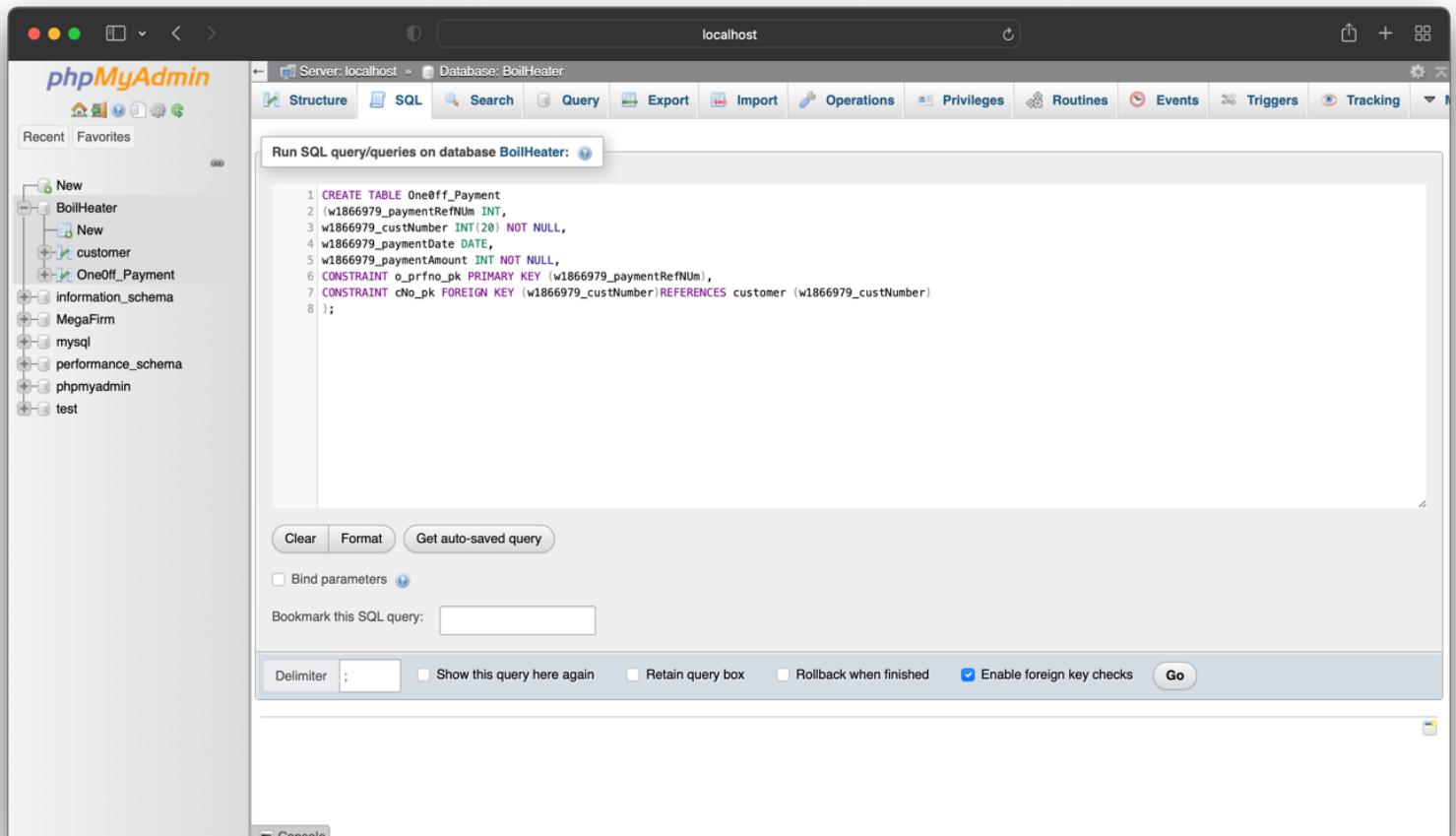
```
1 ALTER TABLE customer
2 MODIFY COLUMN w1866979_custNumber int NOT NULL UNIQUE;
```

On the right, a tooltip displays the column definition: `w1866979_custNumber int NOT NULL UNIQUE;`. Below the SQL input, there are several buttons: 'SELECT *', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', 'Format', 'Get auto-saved query', 'Bind parameters', 'Bookmark this SQL query', 'Delimiter ;', 'Show this query here again', 'Retain query box', 'Rollback when finished', 'Enable foreign key checks', and a 'Go' button.



The screenshot shows the phpMyAdmin interface after the SQL query has been executed. The main area displays a message: `MySQL returned an empty result set (i.e. zero rows). (Query took 0.0026 seconds.)`. Below this message, the same SQL code is shown again: `ALTER TABLE customer MODIFY COLUMN w1866979_custNumber int NOT NULL UNIQUE;`. There are three buttons below the message: '[Edit inline]', '[Edit]', and '[Create PHP code]'.

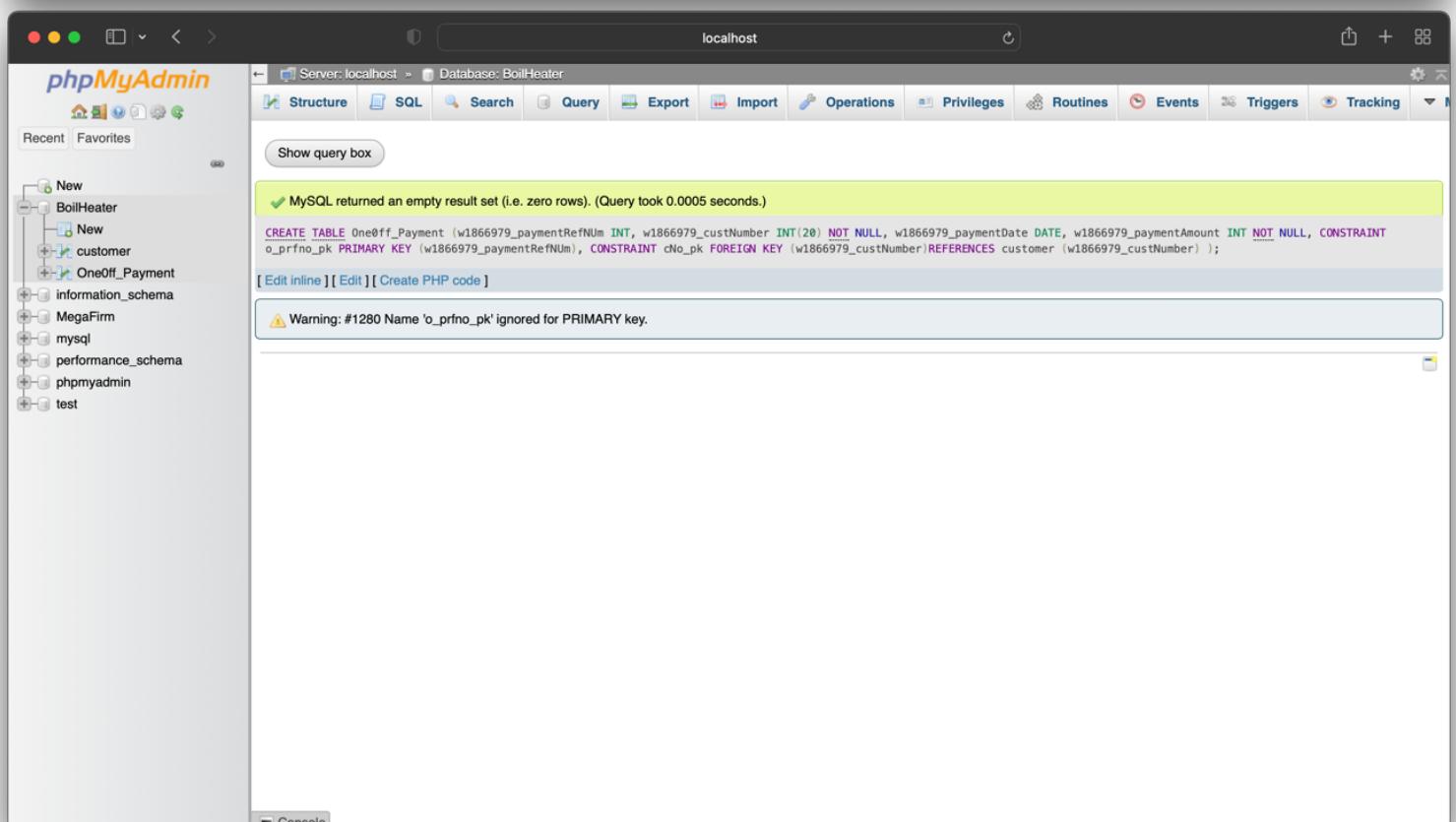
3.2.4 Create oneoff_payment Table



The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The 'SQL' tab is selected. The SQL query editor contains the following code:

```
1 CREATE TABLE OneOff_Payment
2 (w1866979_paymentRefNum INT,
3 w1866979_custNumber INT(20) NOT NULL,
4 w1866979_paymentDate DATE,
5 w1866979_paymentAmount INT NOT NULL,
6 CONSTRAINT o_prfno_pk PRIMARY KEY (w1866979_paymentRefNum),
7 CONSTRAINT cNo_pk FOREIGN KEY (w1866979_custNumber)REFERENCES customer (w1866979_custNumber)
8 );
```

Below the query editor, the 'Go' button is highlighted. The 'Console' tab is visible at the bottom.



The screenshot shows the phpMyAdmin interface after the table creation query has been run. The 'SQL' tab is selected. The results pane displays the following message:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

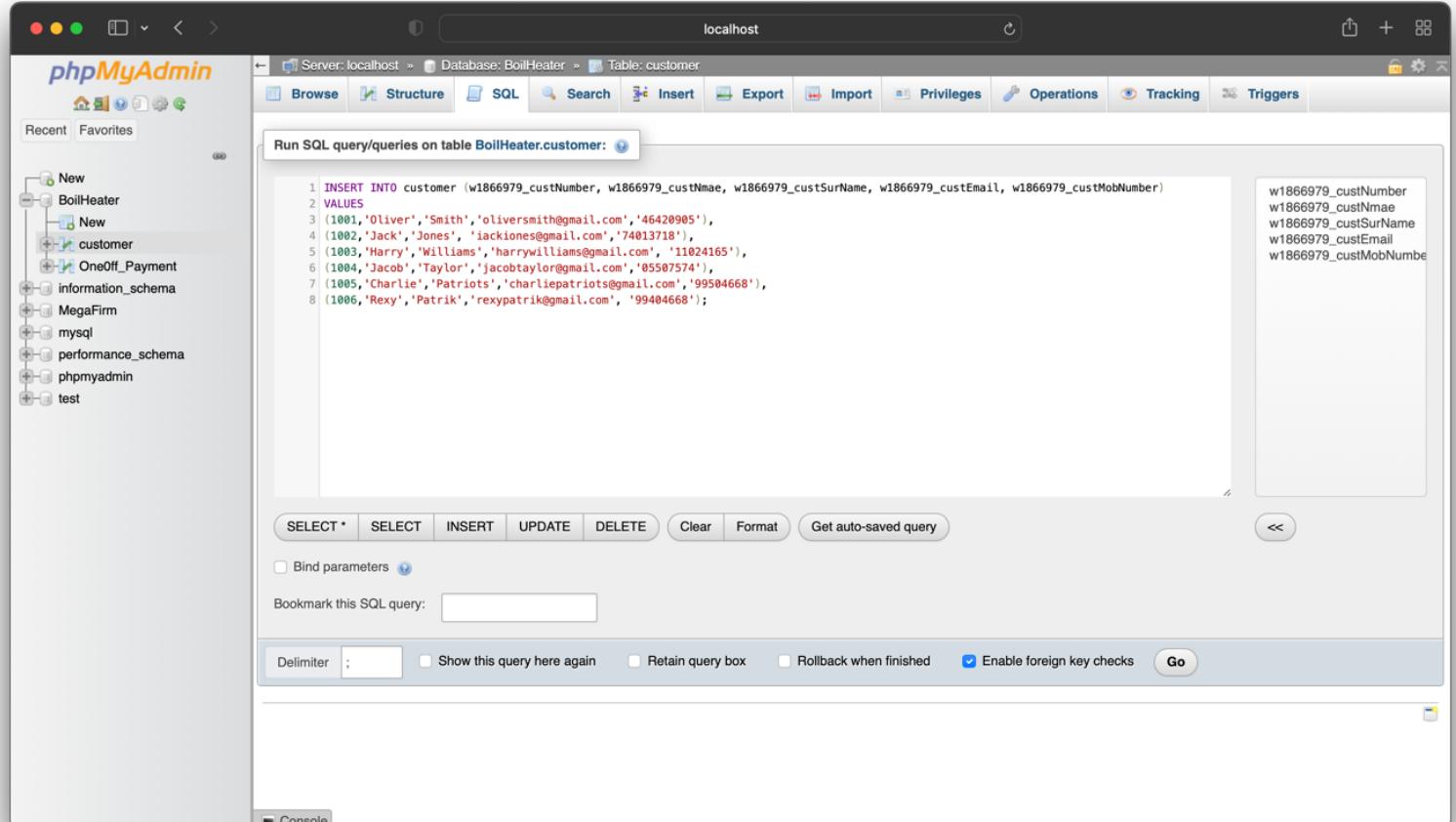
```
CREATE TABLE OneOff_Payment (w1866979_paymentRefNum INT, w1866979_custNumber INT(20) NOT NULL, w1866979_paymentDate DATE, w1866979_paymentAmount INT NOT NULL, CONSTRAINT o_prfno_pk PRIMARY KEY (w1866979_paymentRefNum), CONSTRAINT cNo_pk FOREIGN KEY (w1866979_custNumber)REFERENCES customer (w1866979_custNumber) );
```

Below the message, a warning is shown:

Warning: #1280 Name 'o_prfno_pk' ignored for PRIMARY key.

The 'Console' tab is visible at the bottom.

3.2.5 Populating customer Table



The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The 'customer' table is selected. The SQL query entered is:

```

1 INSERT INTO customer (w1866979_custNumber, w1866979_custName, w1866979_custSurName, w1866979_custEmail, w1866979_custMobNumber)
2 VALUES
3 (1001,'Oliver','Smith','oliversmith@gmail.com','46420905'),
4 (1002,'Jack','Jones','jacksones@gmail.com','74013718'),
5 (1003,'Harry','Williams','harrywilliams@gmail.com','11024165'),
6 (1004,'Jacob','Taylor','jacbtaylor@gmail.com','05507574'),
7 (1005,'Charlie','Patriots','charliepatriots@gmail.com','99504668'),
8 (1006,'Rexy','Patrik','rexpatrik@gmail.com','99404668');

```

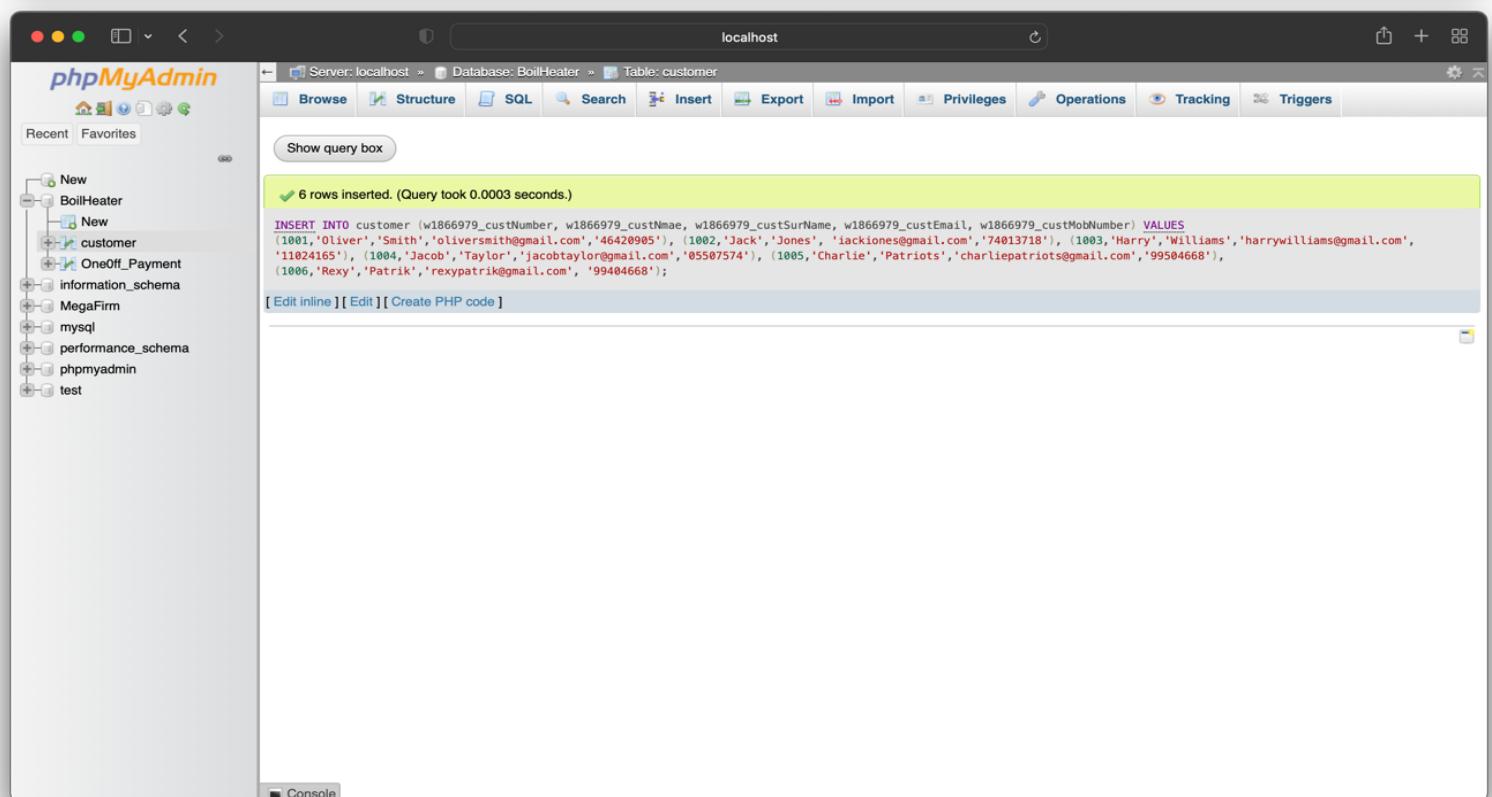
The results pane on the right shows the inserted data:

```

w1866979_custNumber
w1866979_custName
w1866979_custSurName
w1866979_custEmail
w1866979_custMobNumber

```

Below the query editor, the 'Go' button is highlighted.



The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The 'customer' table is selected. The message '6 rows inserted. (Query took 0.0003 seconds.)' is displayed in a green box. The SQL query is the same as in the previous screenshot:

```

INSERT INTO customer (w1866979_custNumber, w1866979_custName, w1866979_custSurName, w1866979_custEmail, w1866979_custMobNumber) VALUES
(1001,'Oliver','Smith','oliversmith@gmail.com','46420905'), (1002,'Jack','Jones','jacksones@gmail.com','74013718'), (1003,'Harry','Williams','harrywilliams@gmail.com','11024165'), (1004,'Jacob','Taylor','jacbtaylor@gmail.com','05507574'), (1005,'Charlie','Patriots','charliepatriots@gmail.com','99504668'), (1006,'Rexy','Patrik','rexpatrik@gmail.com','99404668');

```

Below the message, there are links for 'Edit inline', 'Edit', and 'Create PHP code'.

localhost

phpMyAdmin

Recent Favorites

New

BoilHeater

New

customer

OneOff_Payment

information_schema

MegaFirm

mysql

performance_schema

phpmyadmin

test

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

SELECT * FROM `customer`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	w1866979_custNumber	w1866979_custName	w1866979_custSurName	w1866979_custEmail	w1866979_custMobNumber
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1001	Oliver	Smith	oliversmith@gmail.com	46420905
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1002	Jack	Jones	iackiones@gmail.com	74013718
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1003	Harry	Williams	harrywilliams@gmail.com	11024165
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1004	Jacob	Taylor	jacobtaylor@gmail.com	5507574
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1005	Charlie	Patriots	charliepatriots@gmail.com	99504668
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1006	Rexy	Patrik	rexpatrik@gmail.com	99404668

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

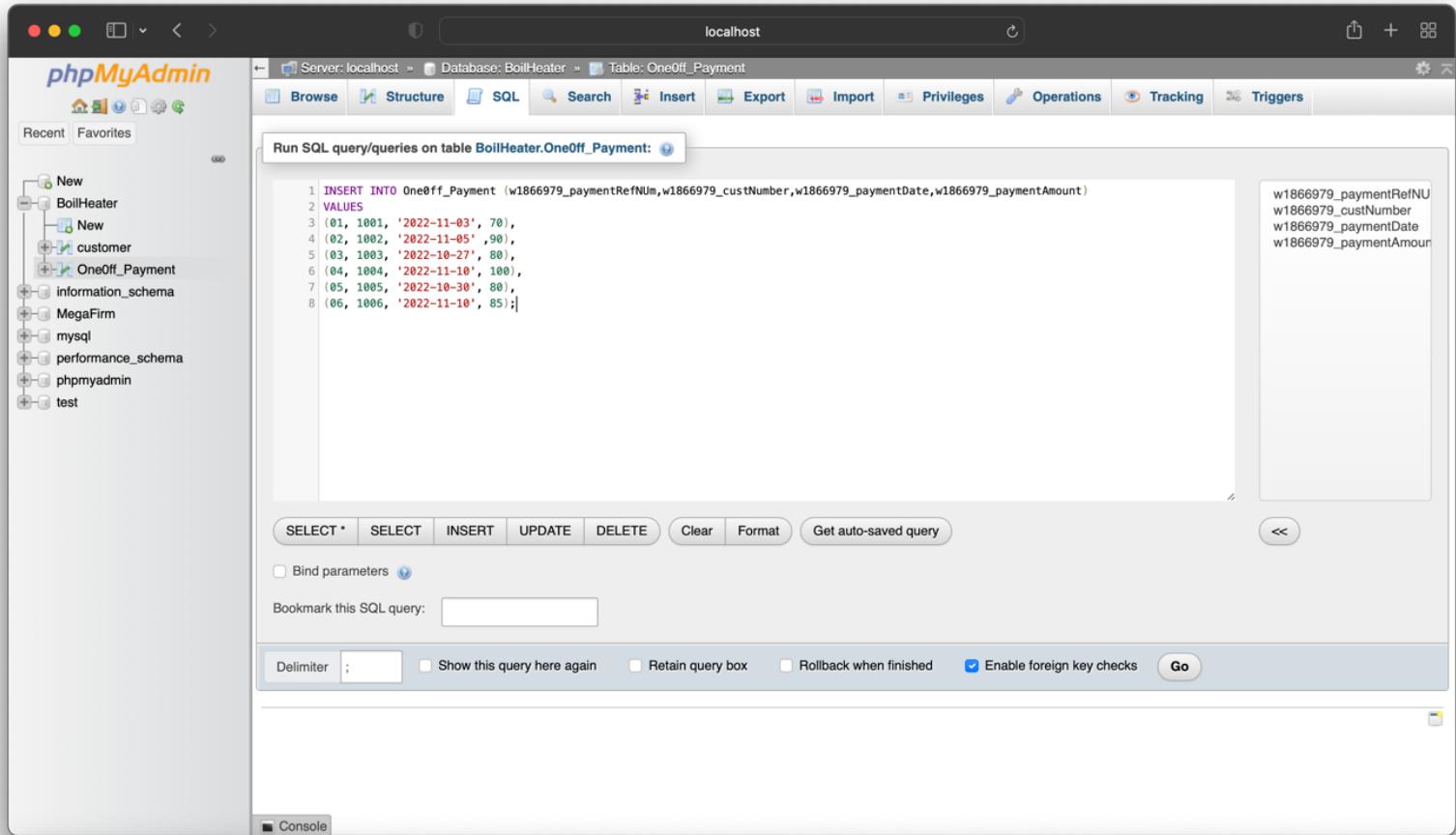
Bookmark this SQL query

Console Let every user access this bookmark

The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The 'customer' table is selected. The table data is as follows:

	w1866979_custNumber	w1866979_custName	w1866979_custSurName	w1866979_custEmail	w1866979_custMobNumber
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1001	Oliver	Smith	oliversmith@gmail.com	46420905
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1002	Jack	Jones	iackiones@gmail.com	74013718
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1003	Harry	Williams	harrywilliams@gmail.com	11024165
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1004	Jacob	Taylor	jacobtaylor@gmail.com	5507574
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1005	Charlie	Patriots	charliepatriots@gmail.com	99504668
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1006	Rexy	Patrik	rexpatrik@gmail.com	99404668

3.2.6 Populating oneoff_Payment Table

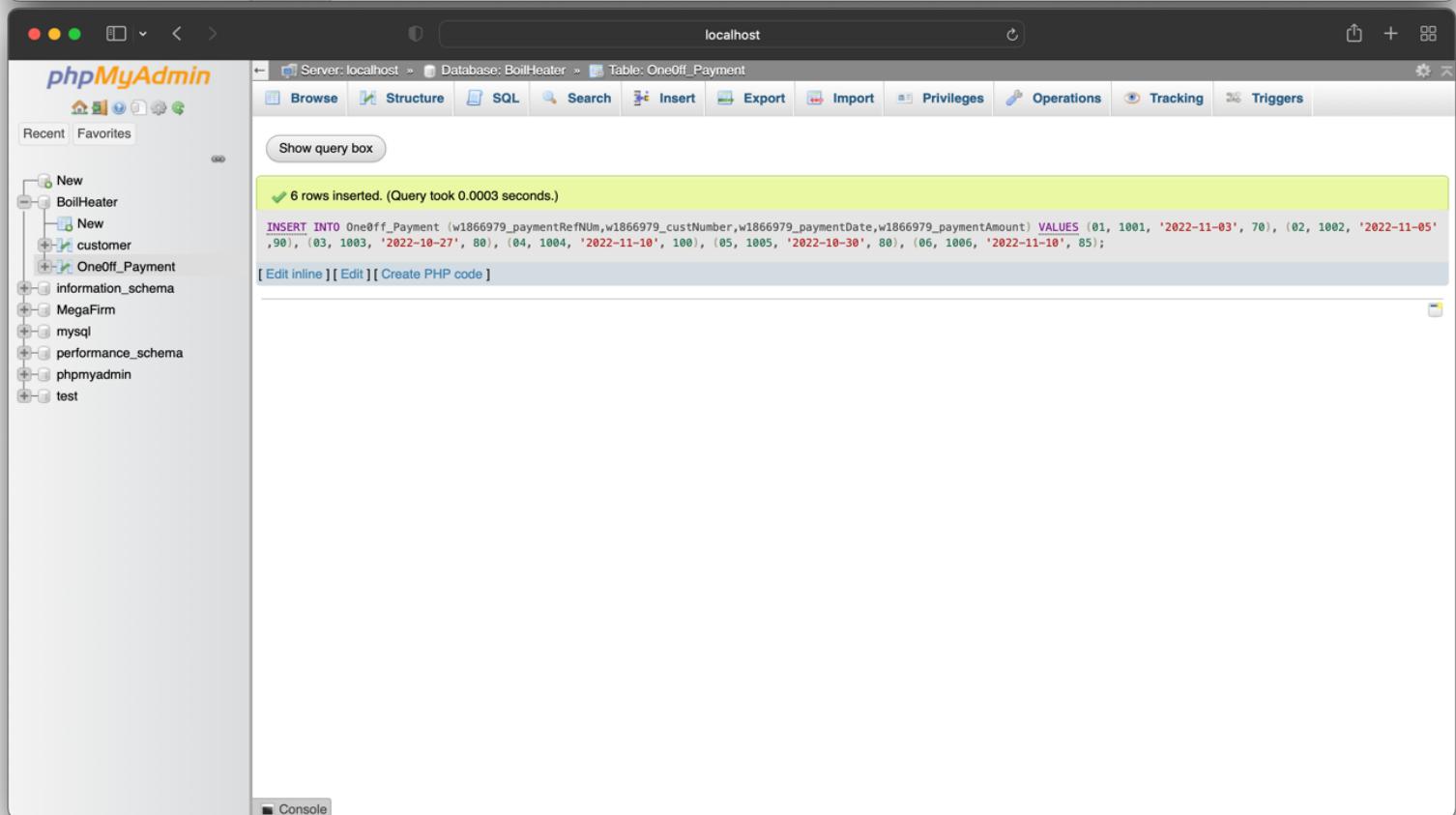


The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The 'OneOff_Payment' table is selected. The SQL tab contains the following query:

```
1 INSERT INTO OneOff_Payment (w1866979_paymentRefNum, w1866979_custNumber, w1866979_paymentDate, w1866979_paymentAmount)
2 VALUES
3 (01, 1001, '2022-11-03', 70),
4 (02, 1002, '2022-11-05', 90),
5 (03, 1003, '2022-10-27', 80),
6 (04, 1004, '2022-11-10', 100),
7 (05, 1005, '2022-10-30', 80),
8 (06, 1006, '2022-11-10', 85);
```

The results pane on the right shows the inserted rows:

w1866979_paymentRefNU	w1866979_custNumber	w1866979_paymentDate	w1866979_paymentAmour
01	1001	2022-11-03	70
02	1002	2022-11-05	90
03	1003	2022-10-27	80
04	1004	2022-11-10	100
05	1005	2022-10-30	80
06	1006	2022-11-10	85



The screenshot shows the phpMyAdmin interface for the 'BoilHeater' database. The 'OneOff_Payment' table is selected. The results pane displays the message: "6 rows inserted. (Query took 0.0003 seconds.)" Below the message is the same SQL query as in the previous screenshot.

localhost

phpMyAdmin

Recent Favorites

New

BoilHeater

- New
- customer
- Oneoff_Payment

information_schema

MegaFirm

mysql

performance_schema

phpmyadmin

test

Server: localhost > Database: BoilHeater > Table: Oneoff_Payment

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

SELECT * FROM `Oneoff_Payment`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	w1866979_paymentRefNum	w1866979_custNumber	w1866979_paymentDate	w1866979_paymentAmount
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1001	2022-11-03	70
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1002	2022-11-05	90
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	1003	2022-10-27	80
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	1004	2022-11-10	100
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	1005	2022-10-30	80
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	1006	2022-11-10	85

Check all With selected: Edit Copy Delete Export

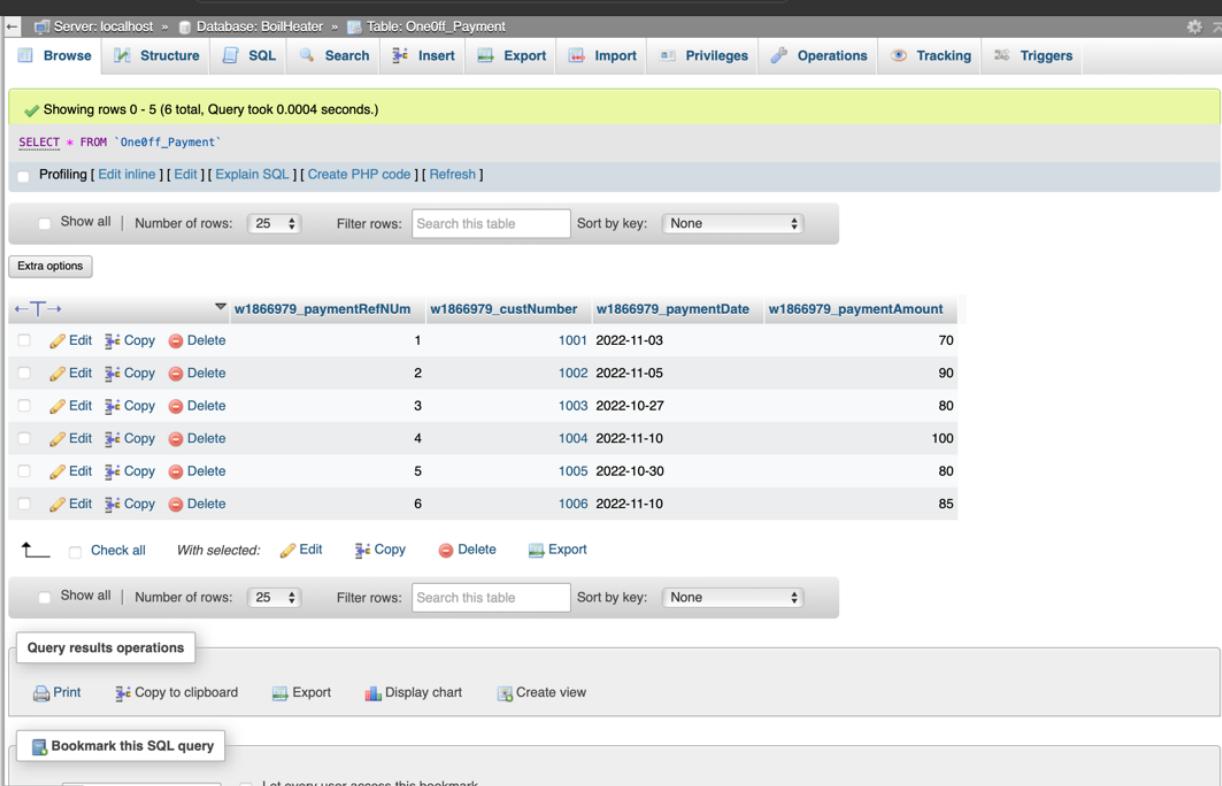
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

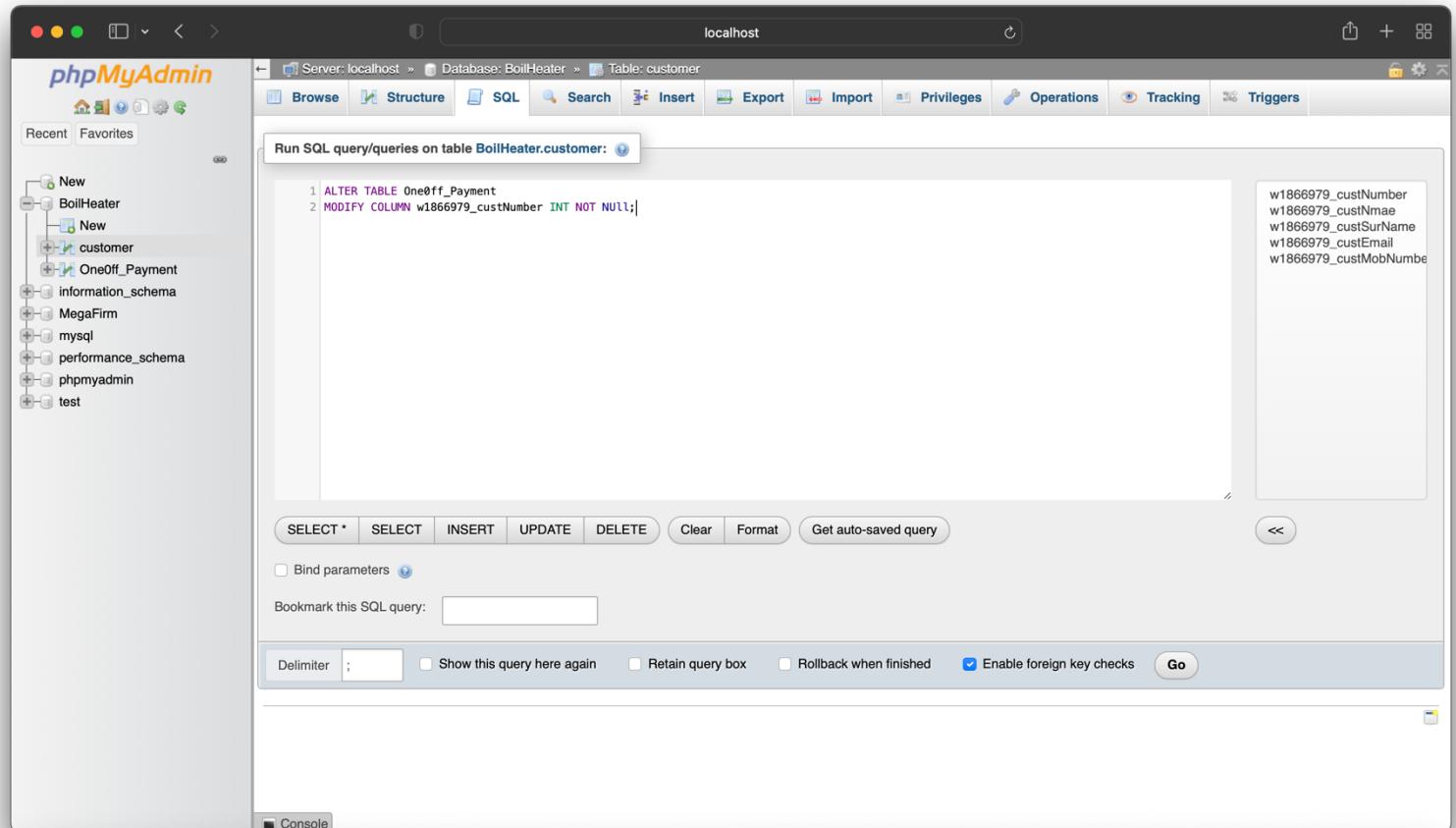
Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Console Let every user access this bookmark



3.2.7 Update foreign key



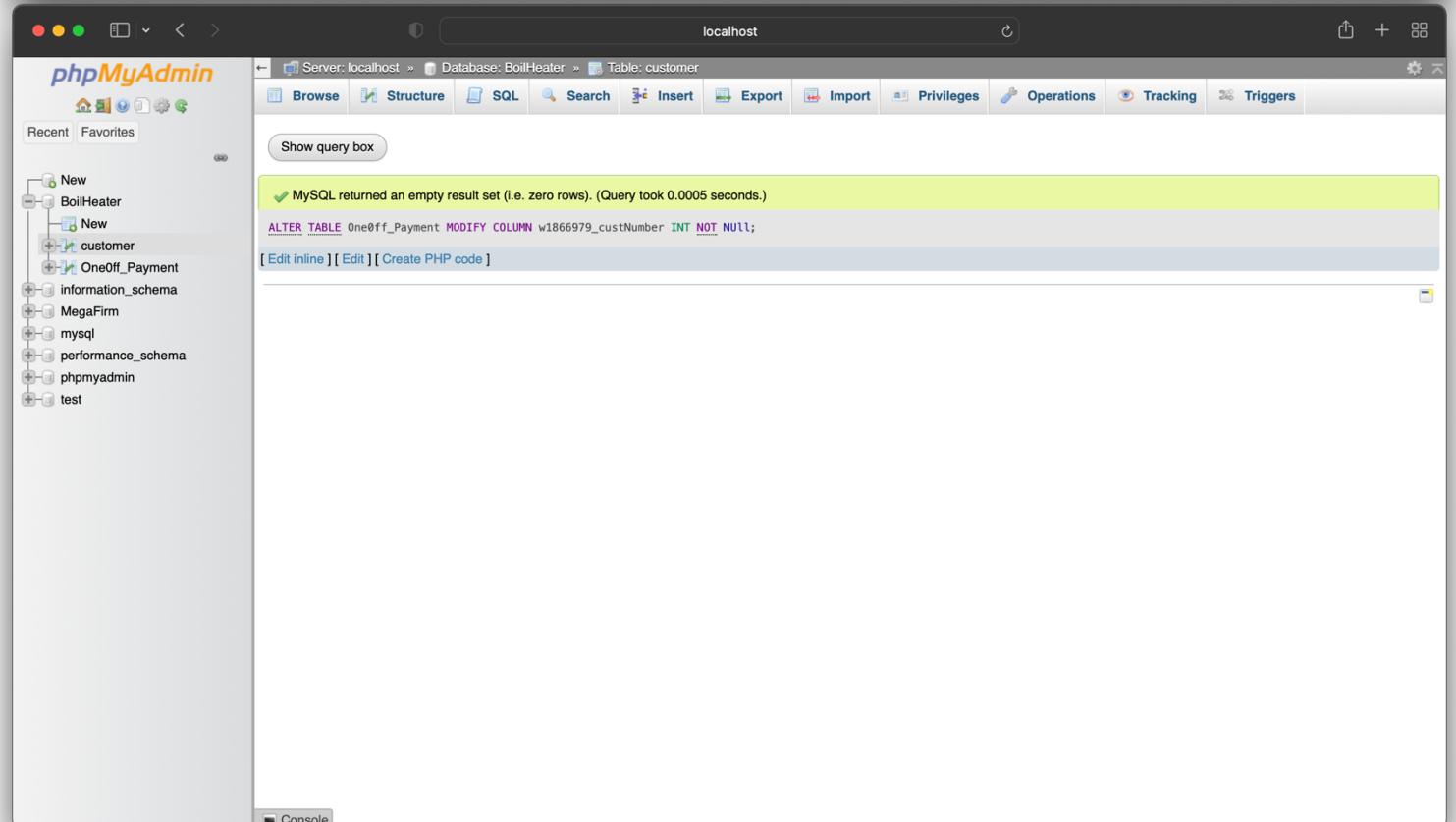
The screenshot shows the phpMyAdmin interface for the 'customer' table in the 'BoilHeater' database. The SQL query entered is:

```
1 ALTER TABLE OneOff_Payment
2 MODIFY COLUMN w1866979_custNumber INT NOT NULL;
```

The results pane on the right shows the following table structure:

w1866979_custNumber
w1866979_custName
w1866979_custSurName
w1866979_custEmail
w1866979_custMobileNumber

Below the query, there are several buttons: SELECT*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, Get auto-saved query, Bind parameters, and a bookmark input field. At the bottom, there are options for Delimiter, Show this query here again, Retain query box, Rollback when finished, and Enable foreign key checks (which is checked). A 'Go' button is also present.

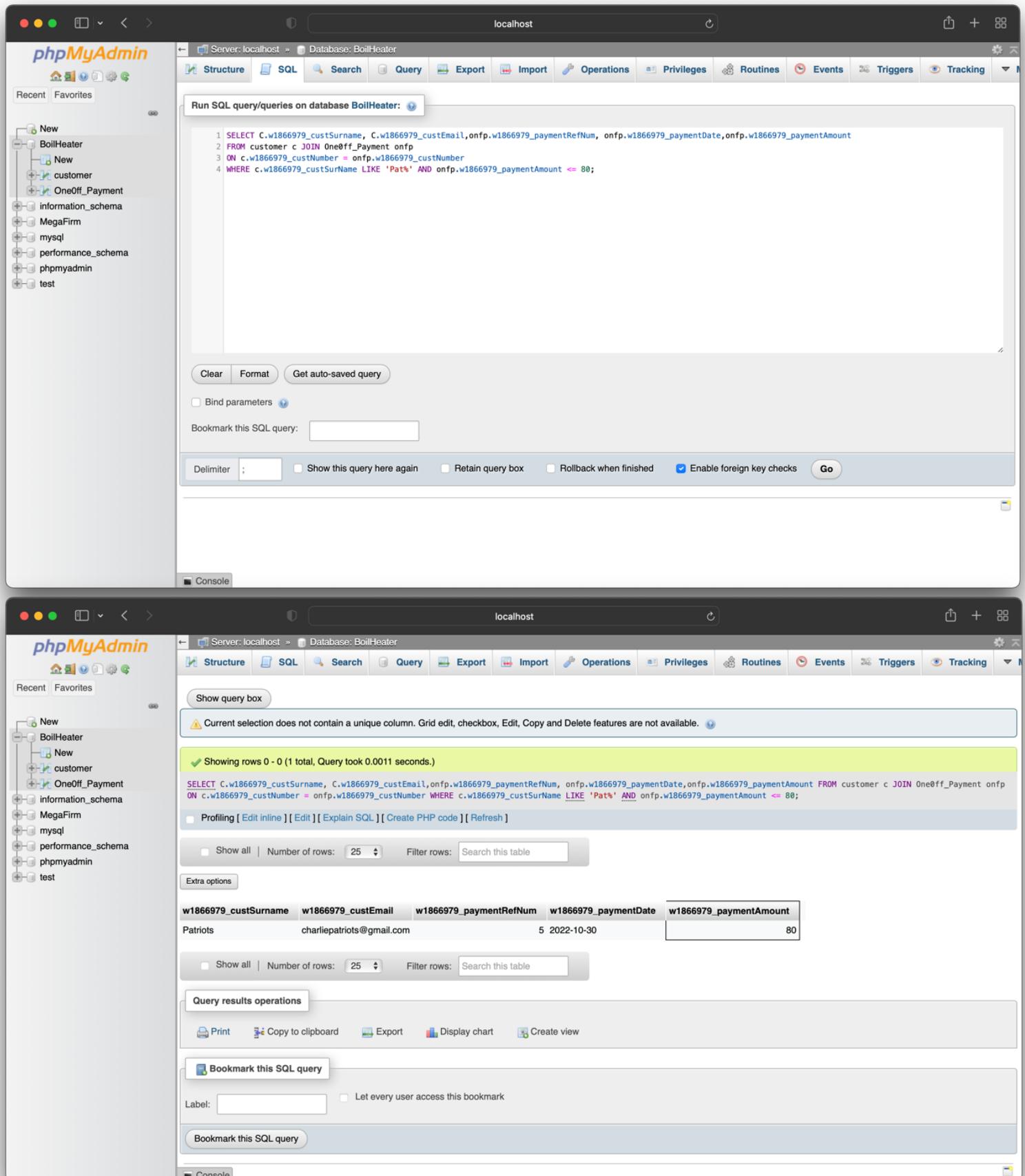


The screenshot shows the phpMyAdmin interface after the query execution. The results pane displays a message: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)" The query listed is:

```
ALTER TABLE OneOff_Payment MODIFY COLUMN w1866979_custNumber INT NOT NULL;
```

Below the message, there are links for Edit inline, Edit, and Create PHP code.

3.2.8 Query



The image shows two screenshots of the phpMyAdmin interface, one above the other, demonstrating a SQL query and its results.

Top Screenshot (Query Page):

- Left Panel:** Shows the database structure with the 'BoilHeater' database selected. Tables include 'New', 'BoilHeater', 'New', 'customer', 'Oneoff_Payment', 'information_schema', 'MegaFirm', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'.
- Top Bar:** Shows 'localhost' and the 'BoilHeater' database.
- Toolbar:** Includes 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', 'Privileges', 'Routines', 'Events', 'Triggers', and 'Tracking'.
- Query Editor:** Contains the following SQL code:

```
1 SELECT C.w1866979_custSurname, C.w1866979_custEmail, onfp.w1866979_paymentRefNum, onfp.w1866979_paymentDate, onfp.w1866979_paymentAmount
2 FROM customer c JOIN Oneoff_Payment onfp
3 ON c.w1866979_custNumber = onfp.w1866979_custNumber
4 WHERE c.w1866979_custSurName LIKE 'Pat%' AND onfp.w1866979_paymentAmount <= 80;
```
- Buttons:** 'Clear', 'Format', 'Get auto-saved query', 'Bind parameters', 'Bookmark this SQL query', 'Delimiter', 'Show this query here again', 'Retain query box', 'Rollback when finished', 'Enable foreign key checks', and 'Go'.
- Console:** A large empty area for command-line input.

Bottom Screenshot (Results Page):

- Left Panel:** Same as the top screenshot.
- Top Bar:** Shows 'localhost' and the 'BoilHeater' database.
- Toolbar:** Same as the top screenshot.
- Message:** 'Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)'
- Query Results:** Shows the following data:

```
SELECT C.w1866979_custSurname, C.w1866979_custEmail, onfp.w1866979_paymentRefNum, onfp.w1866979_paymentDate, onfp.w1866979_paymentAmount FROM customer c JOIN Oneoff_Payment onfp
ON c.w1866979_custNumber = onfp.w1866979_custNumber WHERE c.w1866979_custSurName LIKE 'Pat%' AND onfp.w1866979_paymentAmount <= 80;
```

w1866979_custSurname	w1866979_custEmail	w1866979_paymentRefNum	w1866979_paymentDate	w1866979_paymentAmount
Patriots	charliepatriots@gmail.com	5	2022-10-30	80
- Buttons:** 'Show all', 'Number of rows: 25', 'Filter rows: Search this table', 'Extra options', 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view), and 'Bookmark this SQL query' (Label: [input], Let every user access this bookmark).
- Console:** A large empty area for command-line input.

4 MySQL and MongoDB

MySQL	MongoDB
open-source relational database management system (RDBMS)	open-source non-relational database management system (NoSQL)
MySQL is quite slow in comparison to MongoDB while dealing with large databases.	It has the ability to handle large unstructured data
Constant development is done by the Oracle Corporation.	Ongoing development is done by MongoDB, Inc.
options for scalability are much more limited.	the database is extremely easy to scale
MySQL supports JOIN operations.	MongoDB doesn't support JOIN.
Supported languages are C++, C and JavaScript	Supported languages are C++, C
In MySQL, each individual records are stored as 'rows' in a table	In MongoDB, each individual records are stored as 'documents'.
A 'table' is used to store rows (records) of similar type.	Documents belonging to a particular class or group are stored in a 'collection'.
MySQL concept does not allow efficient replication and sharding but in MySQL, one can access associated data using joins which minimizes duplication.	MongoDB was designed with high availability and scalability in mind, and includes out-of-the-box replication and sharding.
Risk of SQL injection attacks	No schema definition required so lesser risk of attack due to design

Reference

- Comparing The Differences - MongoDB Vs MySQL. (no date). *MongoDB*. Available from <https://www.mongodb.com/compare/mongodb-mysql> [Accessed 7 December 2022].
- MongoDB vs MySQL. (2018). *GeeksforGeeks*. Available from <https://www.geeksforgeeks.org/mongodb-vs-mysql/> [Accessed 7 December 2022].
- Taylor, D. (2020). MongoDB vs MySQL – Difference Between Them. Available from <https://www.guru99.com/mongodb-vs-mysql.html> [Accessed 7 December 2022].