



University of Westminster & Informatics Institute of Technology

Degree: BEng(Hons) in Software Engineering

Unit Code and Description:
(2022) 5COSC020W Database Systems

Module Leader: Mr.Sivaraman Ragu.

Assignment: Course Work(Part A)

Assignment Type: Individual Course Work

Issue Date: 27th Septemper 2022.

Deadline: 12th December 2022 at 23.59 pm.

Tutorial Group : B

Student Name: S.Mohanaranjan.

Student ID

IIT : 20200607

UOW : 18705841 / W1870584

Part A Project Brief: Tourmato

Tourmato is a touristic company that offers exciting “off-the-beaten track” visiting tours around several cities across Europe. Essentially, Tourmato takes groups of customers around European cities to get them to see them and experience their unique atmospheres, under the direction of a local experienced tour guide.

Each city covered by Tourmato contains multiple touristic attractions considered worth visiting. These attractions fall under two categories: touristic landmarks and restaurants. Landmarks are simply relevant locations in the city that can be viewed, such as monuments, buildings, statues, squares, streets, parks, places of worship and so many more. Restaurants, on the other hand, offer interesting typical foods (dishes and/or drinks) to be sampled so that to allow people to experience the local culinary delicacies.

To provide a range of experiences, every city covered by Tourmato offers many visiting tours. Importantly, the same tour can be scheduled and run multiple times and slightly customised every time. When a tour is actually scheduled on a particular day, it is referred to as a “tour session”. Essentially, a tour session is a visiting tour that has been assigned a specific start date and time, a start address, an end date and time and an end address. Every tour session is also given a tailored price and a maximum number of customers that it can accommodate. A given tour session is naturally associated to a selected list of attractions to be visited (a minimum of 8 for every tour session) and is allocated to a specific tour guide to lead the tour session.

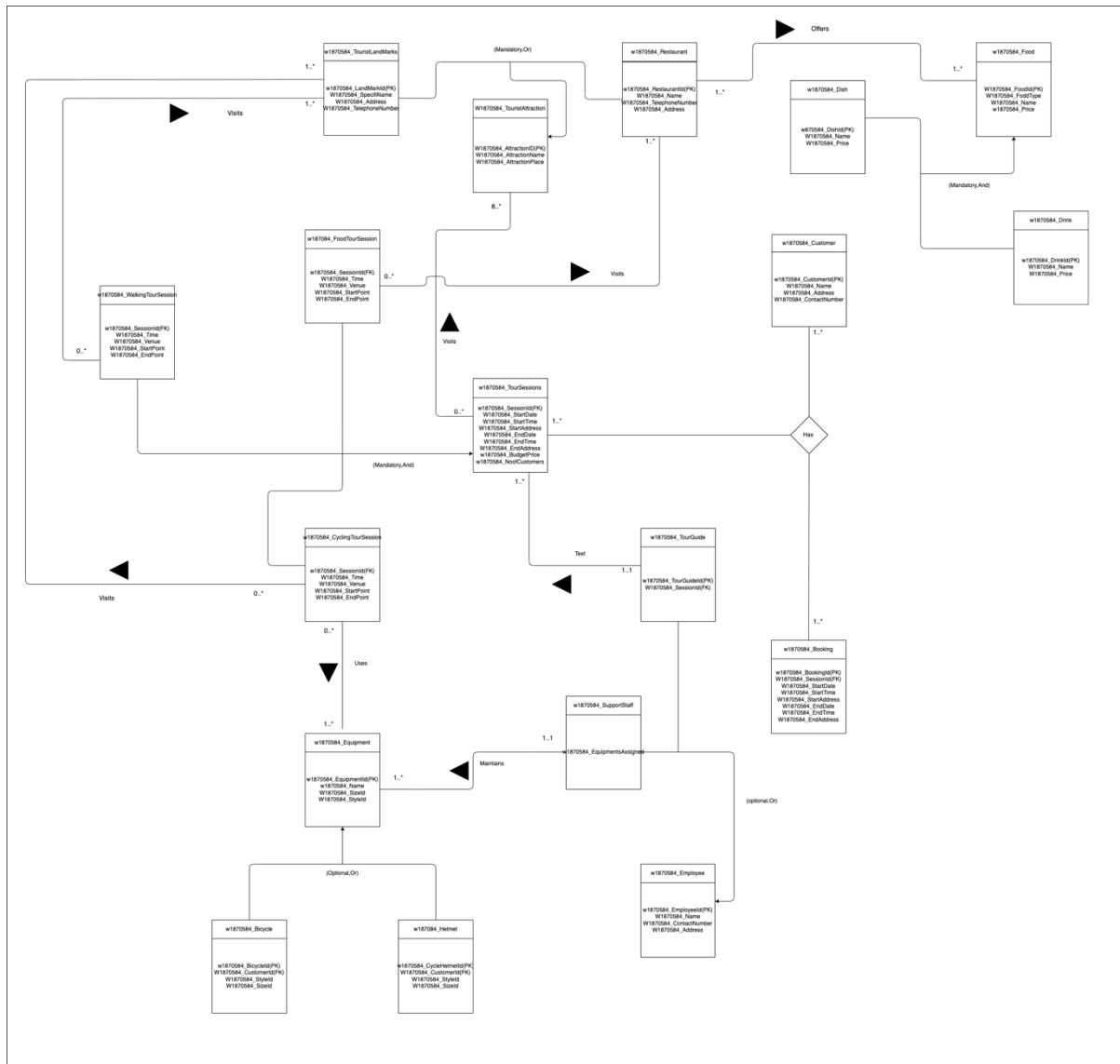
Tourmato customers can place a booking for a variety of exciting tour sessions, depending on what they like. In fact, there are three different types of tour sessions (all led by a tour guide): walking tour sessions, cycling tour sessions and food tour sessions. Walking tour sessions are conducted on foot: customers visit the city by walking from one attraction to another. On cycling tour sessions, customers ride from one attraction to another on a bicycle. Finally, food tour sessions allow customers to stop at different restaurants and sample selected foods i.e., several dishes and/or drinks. It is possible for a walking tour session or a cycling tour session to also be a food tour session: in this case, customers will walk or ride between different attractions, see selected landmarks and stop at different restaurants to taste some of the nice foods on offer. Naturally, a cycling session requires the use of equipment that need to be provided by Tourmato. Tourmato lends two main types of equipment for cycling tour sessions: bicycles and cycle helmets. Bicycles come in several styles and sizes, while helmets also have different sizes.

In terms of staffing, Tourmato relies on highly-trained employees with specialised roles. The job of tour guides is naturally to lead the tour sessions: they take the customers around the city, stop at every attraction, and enthusiastically narrate key facts on the history, geography, architecture, ecology, or gastronomy for each visited attraction. Support staff also play a key role by ensuring the strict maintenance of all equipment used by Tourmato. Each member of the support staff is assigned the responsibility of several pieces of equipment to ensure that they are always kept in great condition. This is obviously a top priority for the company so that to guarantee the best experience and the highest safety of all paying customers placing bookings for exciting tour sessions with Tourmato.

Question 01

Produce a complete **CONCEPTUAL EERD** for Tourmato.

This **CONCEPTUAL EERD** needs to include all the **entities, specialisations, relationships, multiplicities, attributes, and primary keys** that you have identified. It should be easy to read and needs to fit on one page of the report.



Assumption :-

- Only one tour guide is allocated per a tour session
- An equipment is only being maintained by one staff member.

Question 02

Create a **Data Dictionary** to document how you identified the **entities** and **specialisations** for Tourmato. To achieve this, fill in the 2 tables below to summarise and briefly explain the meaning of each entity.

Entity name	Brief Description
W1870584_Customer	A general term describes the people getting the services from the touristic company.
W1870584_Employee	A general term describes the people who work at the company.
W1870584_Equipment	A general term describes the objects which are used by the company to provide the services.
W1870584_TourSession	A general term that describes the details of the organized tour sessions.
W1870584_TouristAttraction	A general term that describes the places which are visited by the customers during the tour.
W1870584_SupportStaff	A general term that describes the employee who is responsible for the maintenance of equipment.
W1870584_TourGuide	A general term that describes the employee who is lead the tour session.
W1870584_Bicycle	A general term describes equipment used to travel during bicycle tours.
W1870584_Helmet	A general term describing equipment that is used during bicycle travel.
W1870584_TouristLandmark	A general term describes the locations in the city which are viewed by the customers.
W1870584_WalkingTourSession	A general term describes the tour session where the customer travels on foot.
W1870584_CyclingTourSession	A general term describes the tour session where the customer travels by bicycle.
W1870584_FoodTourSession	A general term describes the tour session where customers can eat sample foods at different restaurants.
W1870584_Restaurant	A general term that describes the place where customers can eat sample foods.
W1870584_Food	A general term that describes the thing offered by the restaurant.
W1870584_Drink	A general term that describes the thing served by the restaurant to drink.
W1870584_Dish	A general term that describes the variety of things served as food.

General Entity	Specialised Entity	Brief Explanation
W1870584_Employee	W1870584_TourGuide	An employee who is lead the tour session.
W1870584_Employee	W1870584_SupportStaff	An employee who is responsible for the maintenance of equipment.
W1870584_TourSession	W1870584_WalkingTourSession	Tour session where the customer travels on foot.
W1870584_TourSession	W1870584_CyclingTourSession	Tour session where the customer travels by bicycle.
W1870584_TourSessions	W1870584_FoodTourSession	Tour session where customers can eat sample foods at different restaurants.
W1870584_Equipment	W1870584_Bicycle	Equipment used to travel during bicycle tours.
W1870584_Equipment	W1870584_Helmet	Equipment that is used during bicycle travel.
W1870584_Food	W1870584_Drink	Food served by the restaurant to drink.
W1870584_Food	W1870584_Dish	The variety of things served as food.
W1870584_ToursitAttraction	W1870584_Restaurant	The place where customers can eat sample foods.
W1870584_TouristAttraction	W1870584_TouristLandmark	The locations in the city are viewed by the customers.

Question 03

Create a **Data Dictionary** to document how you identified the **relationships** and **multiplicities** for Tourmato. To achieve this, fill in the table below to summarise and justify the multiplicities for each relationship.

Entity name	Multiplicity	Relationship	Multiplicity	Entity name	Brief justifications for the multiplicity (4 statements for each relationship)
W1870584_Customer	1..*	has	1..*	W1870584_TourSessions	A customer must have at least one tour session.
					A customer may have many tour sessions.
W1870584_Customer	1..*	has	1..*	W1870584_Booking	A customer must have at least one booking.
					A customer may have many bookings.
W1870584_TourSession	1..*	has	1..*	W1870584_Customer	A Tour session must have at least one customer.
					A Tour session may have many customers.
W1870584_TourSession	1..*	has	1..*	W1870584_Booking	A Tour session must have at least one booking.
					A Tour session may have many bookings.
W1870584_Booking	1..*	has	1..*	W1870584_Customer	A booking must have at least one customer.
					A booking may have many customers.
W1870584_Booking	1..*	has	1..*	W1870584_TourSessions	A booking must have at least one tour session.
					A booking may have many tour sessions.
W1870584_TourGuide	1..1	conducts	0..*	W1870584_TourSession	A tour guide may not conduct any tour session.
					A tour guide may conduct many tour sessions.

Entity name	Multiplicity	Relationship	Multiplicity	Entity name	Brief justifications for the multiplicity (4 statements for each relationship)
W1870584_TourGuide	1..1	conducts	0..*	W1870584_TourSession	A tour session must be conducted by at least one tour guide.
					A tour session can be conducted by only one tour guide.
W1870584_TourSession	0..*	visits	8..*	W1870584_Attraction	A Tour session must visit at least eight attractions.
					A Tour session may visit multiple attractions.
					An attraction may not be visited by any tour session.
					An attraction may be visited by multiple tour sessions.
W1870584_SupportStaff	1..1	maintains	1..*	W1870584_Equipment	A support staff member must maintain at least one equipment.
					A support staff member may maintain many equipment.
					An equipment must be maintained by at least one support staff member.
					An equipment may only be maintained by one support staff member.
W1870584_CyclingTourSession	0..*	uses	1..*	W1870584_Equipment	A Cycling tour must use at least one equipment.
					A Cycling tour may use many equipment.
					An equipment may not be used by any cycling tour.
					An equipment may be used by many cycling tours.

Entity name	Multiplicity	Relationship	Multiplicity	Entity name	Brief justifications for the multiplicity (4 statements for each relationship)
W1870584_Restaurant	1..*	Offers	1..*	W1870584_Food	<p>A restaurant must offer at least one type of food.</p> <p>A restaurant may offer many types of food.</p> <p>A type of food must be offered by at least one restaurant.</p> <p>A type of food may be offered by many restaurants.</p>
W1870584_FoodTourSession	0..*	Visits	1..*	W1870584_Restaurant	<p>A food tour must visit at least one restaurant.</p> <p>A food tour may visit many restaurants.</p> <p>A restaurant may not be visited by any food tour.</p> <p>A restaurant may be visited by many food tours.</p>
W1870584_WalkingTourSession	0..*	Visits	1..*	W1870584_TouristLandmark	<p>A walking tour must visit at least one landmark.</p> <p>A walking tour may visit many landmarks.</p> <p>A landmark may not be visited by any walking tour.</p> <p>A landmark may be visited by many walking tours.</p>
W1870584_CyclingTourSession	0..*	Visits	1..*	W1870584_TouristLandmarks	<p>A cycling tour must visit at least one landmark.</p> <p>A cycling tour may visit many landmarks.</p> <p>A landmark may not be visited by any cycling tour.</p> <p>A landmark may be visited by many cycling tour</p>

Question 04

Create a **Data Dictionary** to document how you identified the **attributes** and **primary keys** for each entity for Tourmato. To achieve this, fill in the table below to summarise and explain the meaning of each attribute and primary key.

Entity name	Attributes for this entity	Justification
W1870584_Customer	W1870584_CustomerId {PK} W1870584_Name W1870584_ContactNumber W1870584_Address	The code used to uniquely identify the customer. Name of the customer. Contact number of the customer. Address of the customer.
W1870584_Employee	W1870584_EmployeeID {PK} W1870584_Name W1870584_ContactNumber W1870584_Address	The code used to uniquely identify the Employee. Name of the employee. Contact number of the employee. Address of the employee.
W1870584_Equipment	W1870584_EquipmentID {PK} W1870584_Name W1870584_StyleId W1870584_Sizeid	The code used to uniquely identify the Equipment. Name of the equipment. Type of Style with id Type of size with id
W1870584_TouristAttraction	W1870584_AttractionID {PK} W1870584_AttractionName W1870584_AttractionPlace	The code used to uniquely identify the attraction Name of the attraction. City/Place in which the attraction is located.
W1870584_TourSessions	W1870584_TourID {PK} W1870584_StartDate W1870584_StartTime W1870584_StartAddress W1870584_EndDate W1870584_EndTime W1870584_EndAddress W1870584_BudgetPrice W1870584_NoofCustomers	The code used to uniquely identify the booked tour session. Start date of the tour. Start time of the tour. Start address of the tour. End date of the tour. End time of the tour. End address of the tour. Average price for the tour session. Maximum no of customers that can be participate to the tour

Entity name	Attributes for this entity	Justification
W1870584_Bicycle	w1870584_BicycleId(PK) W1870584_CustomerId(FK) W1870584_StyleId W1870584_SizeId	Bicycle code to identify Customer from customer table Style of the bicycle with id Size of the bicycle with id
W1870584_Helmet	w1870584_BicycleId(PK) W1870584_CustomerId(FK) W1870584_StyleId W1870584_SizeId	Helmet code to identify Customer from customer table Style of the helmet with id Size of the helmet with id
W1870584_TourGuide	w1870584_TourGuideId(PK) W1870584_SessionId(FK) W1870584_GuideSpecialization W1870584_YearsofExperiace	Tourguide code Tour guide session involved Specialization of the tour guide relevant to attractions. Years of experience of the tour guide.
W1870584_SupportStaff	W1870584_AssignedEquipments	Equipment assigned to the support staff employee to be maintained.
W1870584_TouristLandmark	w1870584_LandMarkId(PK) W1870584_SpecifiName W1870584_Address W1870584_TelephoneNumber	Specific code to identify Specific name Address of that place Contact number of hat place
W1870584_Restaurant	w1870584_RestaurantId(PK) W1870584_Name W1870584_TelephoneNumber W1870584_Address	The code used to uniquely identify the restaurant. Name of the Restaurant. Contact number Address of that place
W1870584_WalkingTourSession	w1870584_SessionId(FK) W1870584_Time W1870584_Venue W1870584_StartPoint W1870584_EndPoint	Session id from TourSession Time to start Place of gather Start point of session End point of session
W1870584_CyclingTourSession	w1870584_SessionId(FK) W1870584_Time W1870584_Venue W1870584_StartPoint W1870584_EndPoint	Session id from TourSession Time to start Place of gather Start point of session End point of session
W1870584_FoodTourSession	w1870584_SessionId(FK) W1870584_Time W1870584_Venue W1870584_StartPoint W1870584_EndPoint	Session id from TourSession Time to start Place of gather Start point of session End point of session
W1870584_Food	W1870584_FoodId(PK) W1870584_FoodType W1870584_Name w1870584_Price	Code for specific food Type of food Name of food Price of food
W1870584_Dish	W1870584_DishID	the type of the dish.
W1870584_Drink	W1870584_DrinkID	The type of the drink.

Part B Project Brief: BoilHeater

BoilHeater is a home services provider based in the UK. It is specialised in the ongoing maintenance, regular servicing and repairing of central heating systems in properties around the country. Customers can subscribe to a BoilHeater contract to cover their properties. If a maintenance contract is signed, then BoilHeater guarantees the servicing of the central heating installation at regular intervals, as well as repairs in case of breakdowns. Additional costs might be incurred for the purchasing of spare parts for repairs.

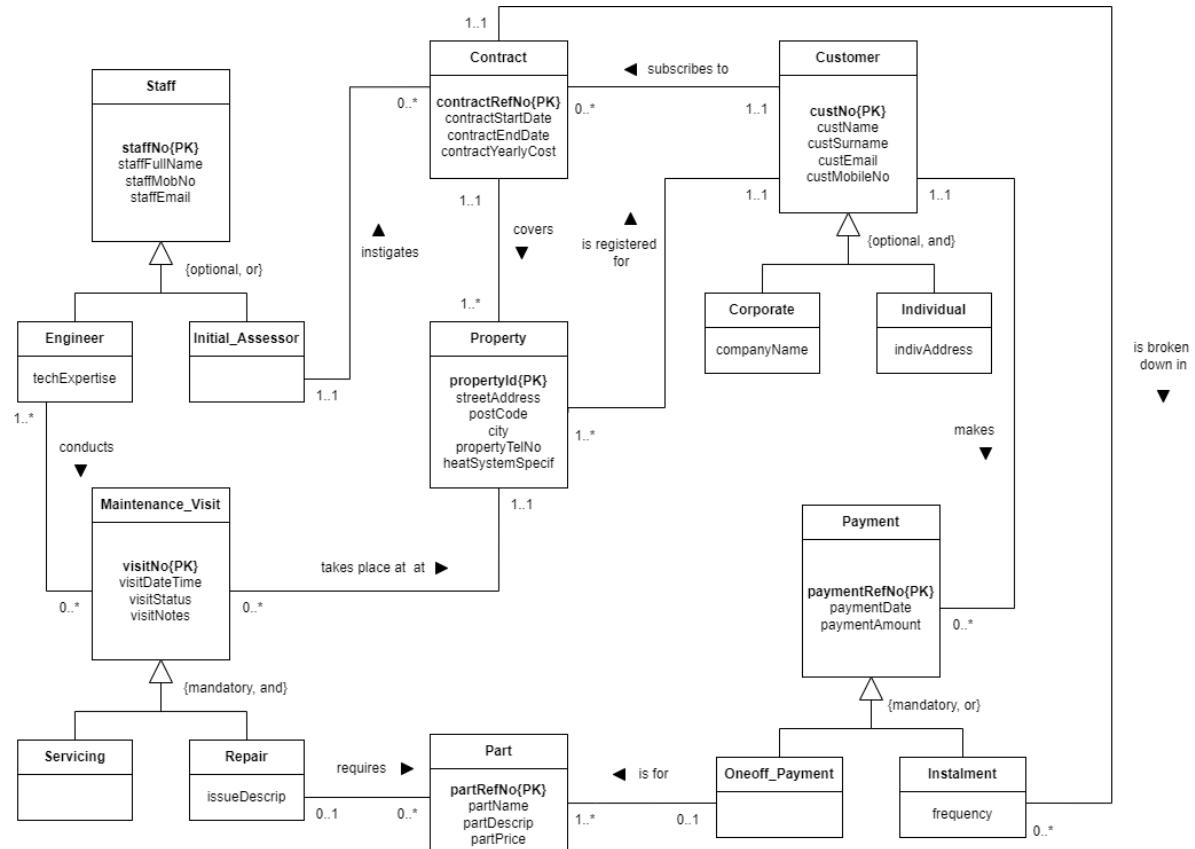
Part B Questions

You have been hired by BoilHeater as a **Database Consultant** to undertake a database project to support the data needs of the firm. In this second part, you are given a conceptual data model for BoilHeater (figure 1), and your first goal is to **map it** onto a high-quality **LOGICAL ENTITY RELATIONALHIP DIAGRAM (ERD)** to logically represent how the key business data needs can be organised as a set of interrelated tables that can then be implemented. These tables need to be interconnected according to the strict rules of the relational model to be implementable. You also have to write a key **SQL query** to retrieve specific data. Finally, you need to present a brief comparative analysis of **a specific relational database vs. a specific NoSQL database** with the aim to provide BoilHeater with informed guidance on which one may be the best option for them.

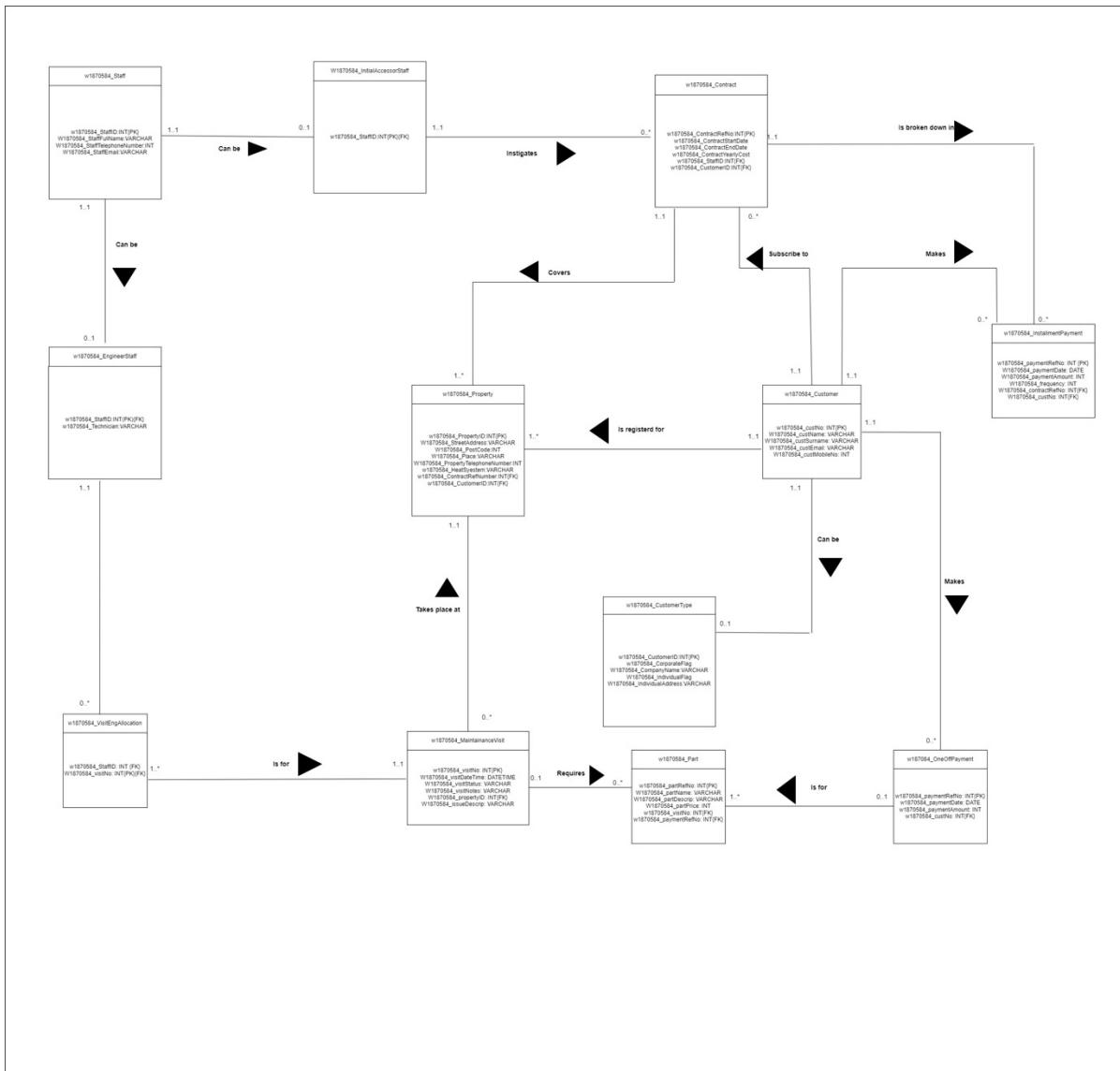
QUESTION 5

Map the Conceptual EERD given on **figure 1** to produce a complete **LOGICAL ERD** for BoilHeater.

This **LOGICAL ERD** needs to include all the **correct tables, relationships, multiplicity constraints, attributes, primary keys and foreign keys**. It should be easy to read and needs to fit on one page of the report.



Answer



QUESTION 6

Based on your logical ERD for BoilHeater, write an **SQL query** to retrieve a list of customer surnames and emails along with the reference numbers, dates and amounts of the one-off payments they made. The list should be restricted as follows: it should only show those customers whose surnames starts with the 3 letters 'Pat' and only the one-off payments for which the paid amount is £80 and under.

Creating Database for BoilHeater

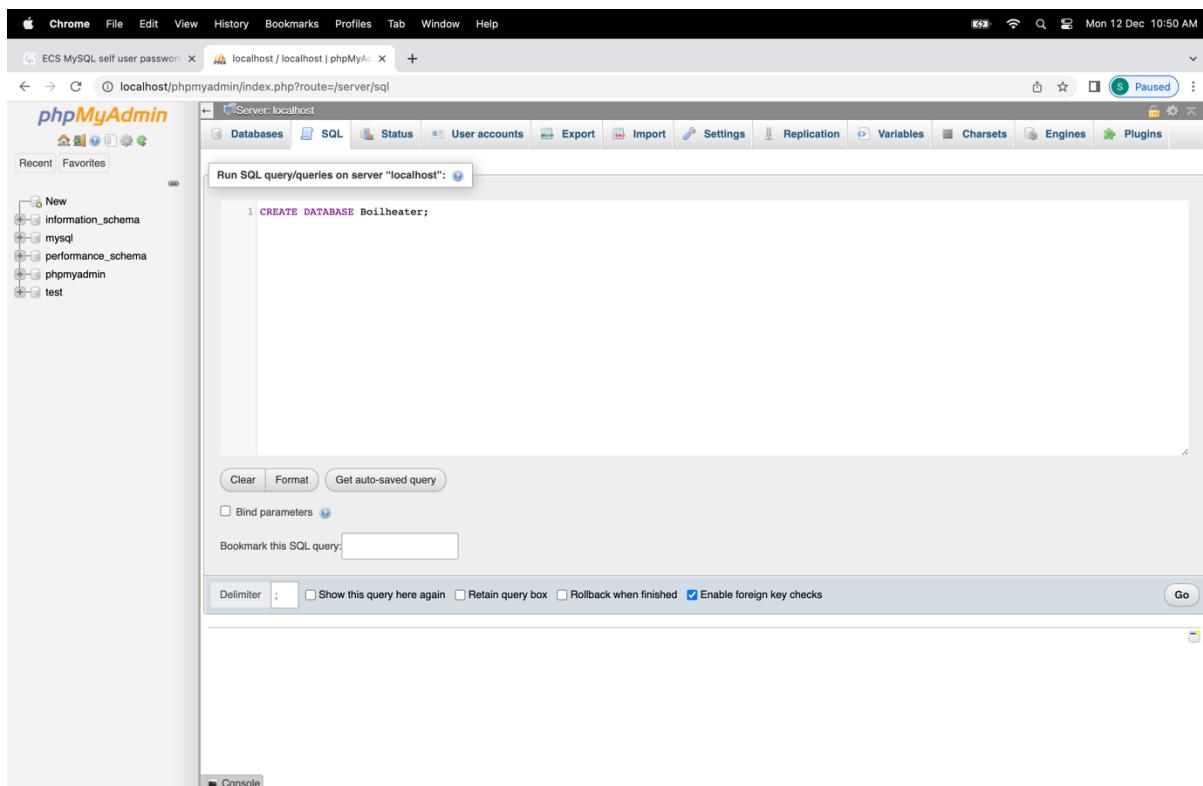
Before Creating Database

The screenshot shows the phpMyAdmin interface on a Mac OS X system. The title bar indicates it's running on localhost. The main window is titled 'Databases' and shows a list of existing databases: information_schema, mysql, performance_schema, phpmyadmin, and test. On the left, there's a sidebar with 'Recent' and 'Favorites' sections, and a 'New' button. In the center, a 'Create database' form is open, with 'Database name' set to 'utf8mb4_general_ci' and a 'Create' button highlighted. Below this, a table lists the current databases with their collations and actions (Check privileges). A note at the bottom says: 'Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.' There's also a link to 'Enable statistics'. At the bottom of the page, there's a 'Console' tab.

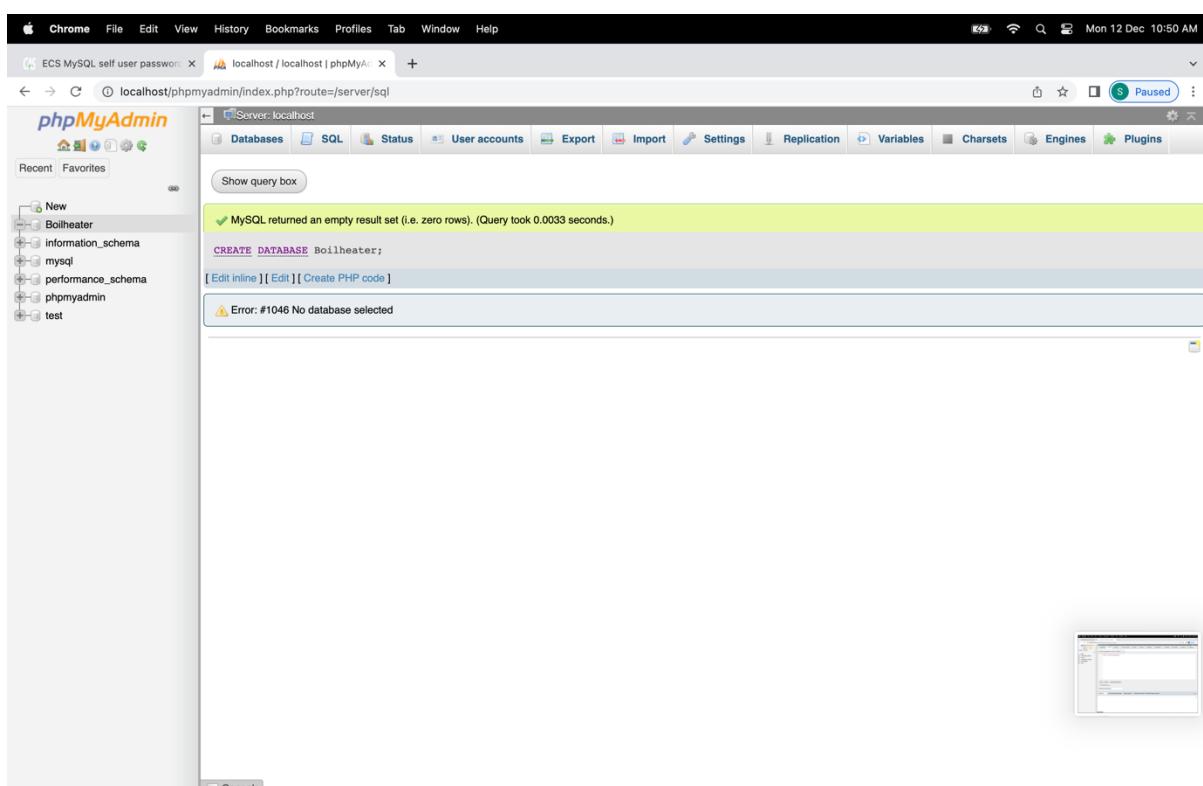
Code:

```
CREATE DATABASE Boilheater ;
```

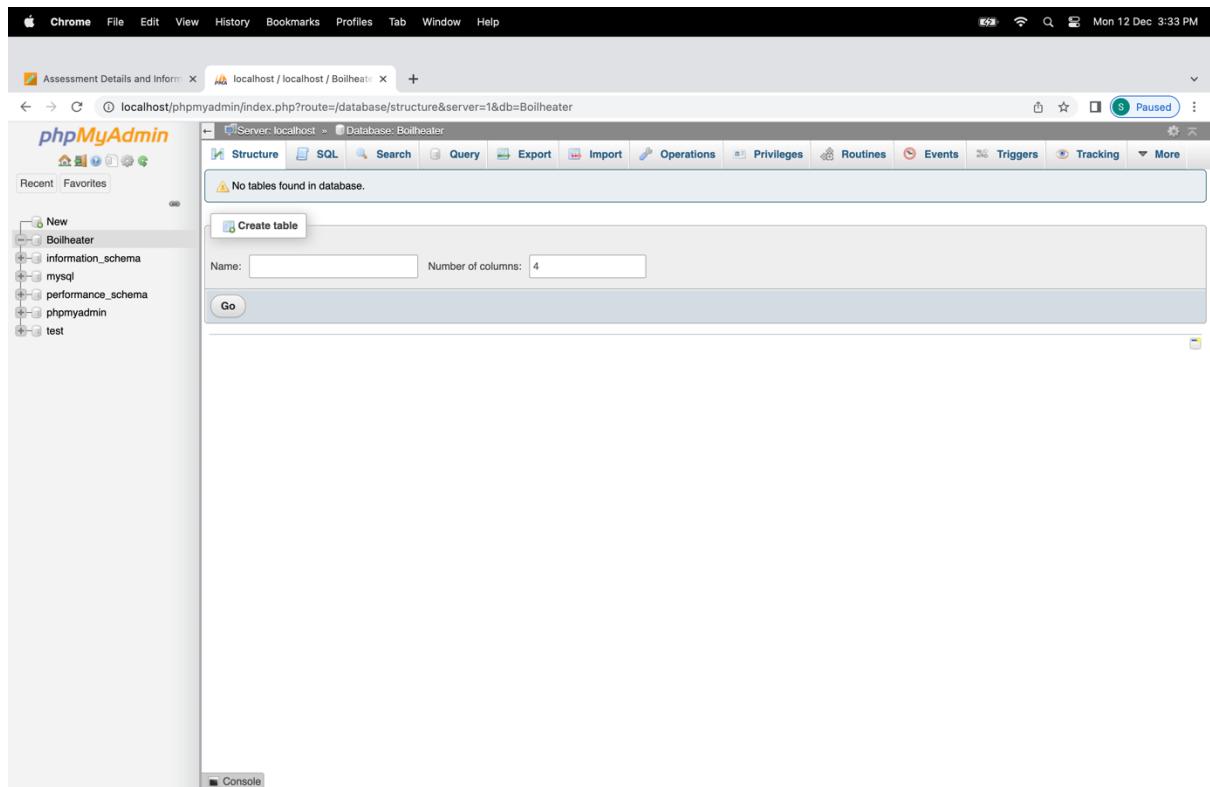
While code



After Execution



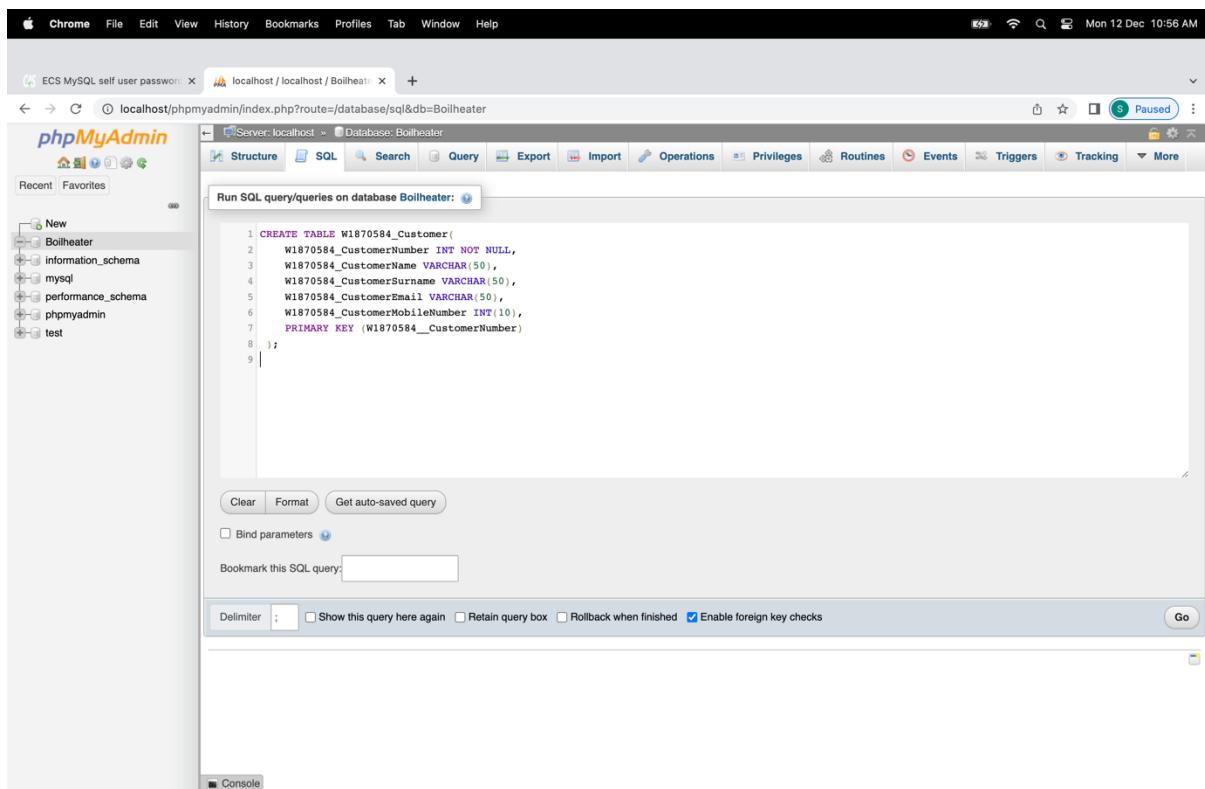
Creating Table for Customer



Code:

```
CREATE TABLE W1870584_Customer(
    W1870584_CustomerNumber INT NOT NULL,
    W1870584_CustomerName VARCHAR(50),
    W1870584_CustomerSurname VARCHAR(50),
    W1870584_CustomerEmail VARCHAR(50),
    W1870584_CustomerMobileNumber INT(10),
    PRIMARY KEY (W1870584_CustomerNumber)
);
```

While Code

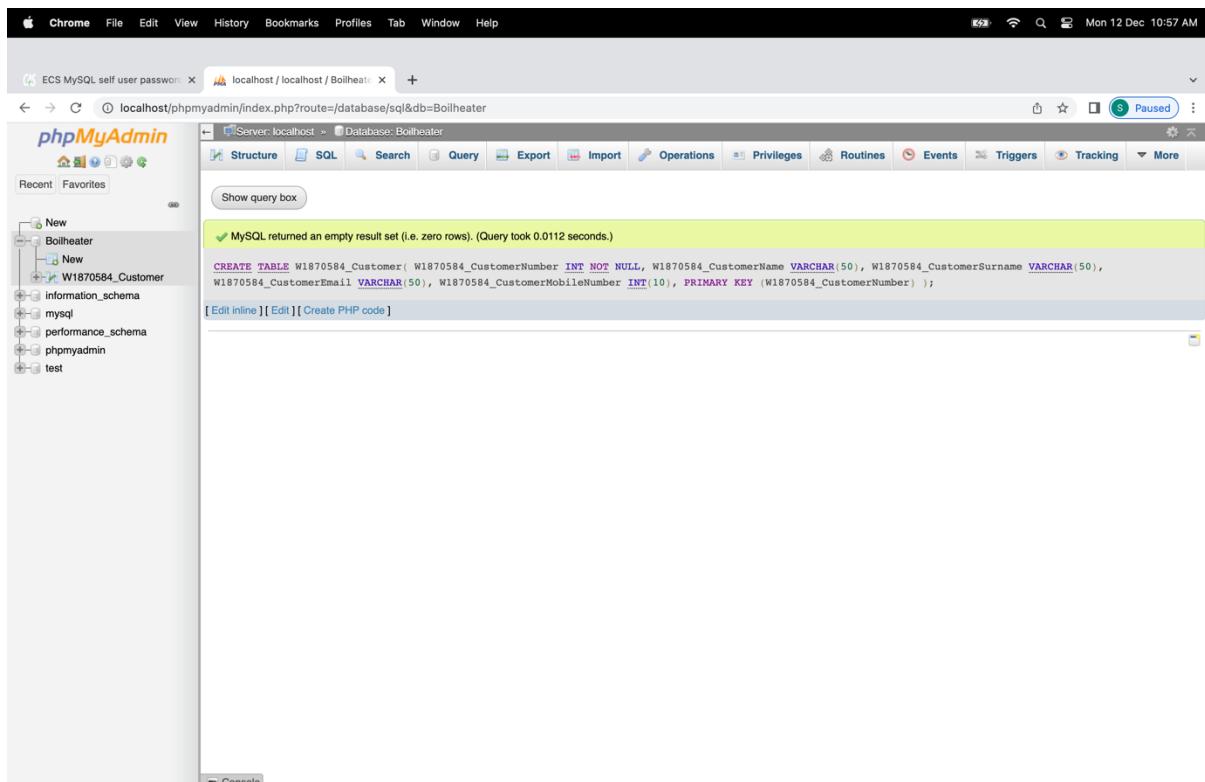


The screenshot shows the phpMyAdmin interface in a web browser. The left sidebar lists databases: New, Bolilheater, information_schema, mysql, performance_schema, phpmyadmin, and test. The main area shows a SQL query window with the following code:

```
1 CREATE TABLE W1870584_Customer(
2     W1870584_CustomerNumber INT NOT NULL,
3     W1870584_CustomerName VARCHAR(50),
4     W1870584_CustomerSurname VARCHAR(50),
5     W1870584_CustomerEmail VARCHAR(50),
6     W1870584_CustomerMobileNumber INT(10),
7     PRIMARY KEY (W1870584_CustomerNumber)
8 );
9 
```

Below the code are buttons for Clear, Format, Get auto-saved query, Bind parameters, and Bookmark this SQL query. At the bottom are Delimiter, Show this query here again, Retain query box, Rollback when finished, Enable foreign key checks, and a Go button.

After Execution



The screenshot shows the phpMyAdmin interface after the SQL query has been executed. The left sidebar remains the same. The main area displays a message: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0112 seconds.)" Below this message is the executed SQL code:

```
CREATE TABLE W1870584_Customer( W1870584_CustomerNumber INT NOT NULL, W1870584_CustomerName VARCHAR(50), W1870584_CustomerSurname VARCHAR(50), W1870584_CustomerEmail VARCHAR(50), W1870584_CustomerMobileNumber INT(10), PRIMARY KEY (W1870584_CustomerNumber) );
```

At the bottom of the message area are links for Edit inline, Edit, and Create PHP code.

Final Output

The screenshot shows the phpMyAdmin interface for the 'Boilheater' database. The left sidebar shows the database structure with 'W1870584_Customer' selected. The main area displays the 'Table structure' for the 'W1870584_Customer' table. The table has five columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	W1870584_CustomerNumber	int(11)	utf8mb4_general_ci	No	None				Change Drop More
2	W1870584_CustomerName	varchar(50)	utf8mb4_general_ci	Yes	NULL				Change Drop More
3	W1870584_CustomerSurname	varchar(50)	utf8mb4_general_ci	Yes	NULL				Change Drop More
4	W1870584_CustomerEmail	varchar(50)	utf8mb4_general_ci	Yes	NULL				Change Drop More
5	W1870584_CustomerMobileNumber	int(10)		Yes	NULL				Change Drop More

Below the table structure, there is a section for 'Indexes' with one primary key index defined:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	W1870584_CustomerNumber	0	A	No	

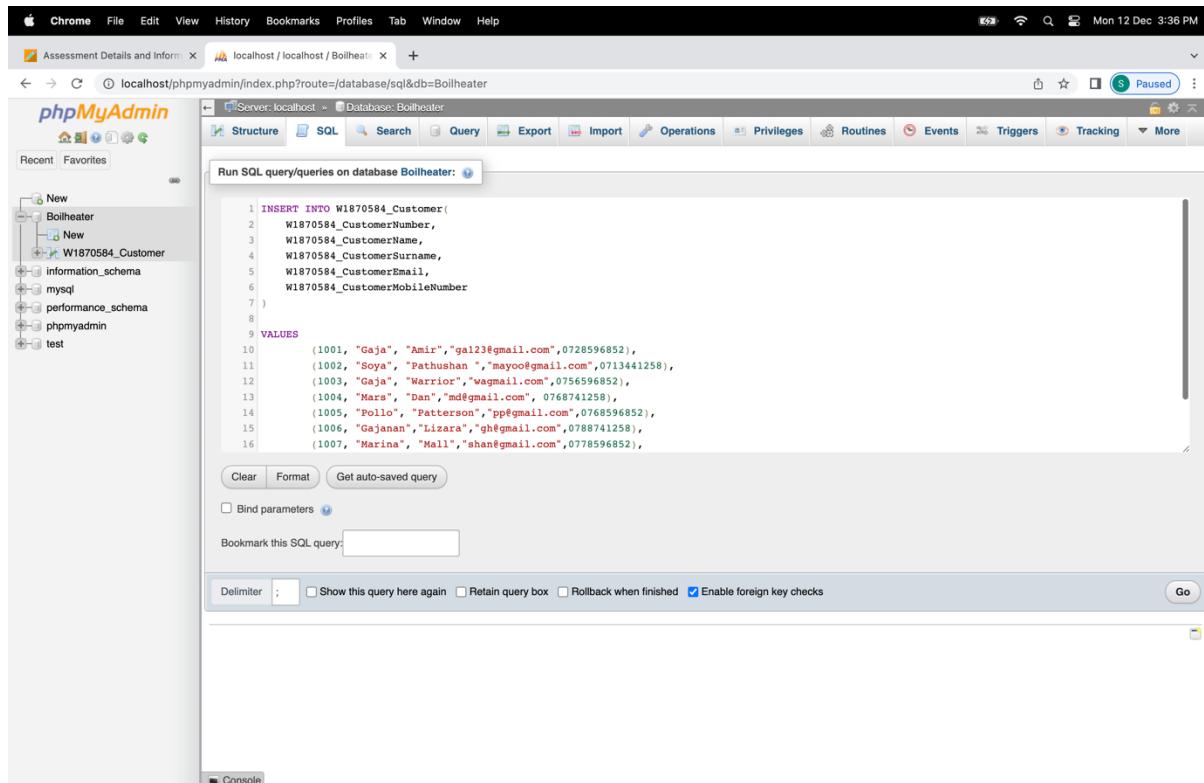
At the bottom, there are buttons for 'Add', 'Print', 'Propose table structure', 'Track table', 'Move columns', 'Normalize', and 'Go'.

Input Values for Customer

Code:

```
INSERT INTO W1870584_Customer(
    W1870584_CustomerNumber,
    W1870584_CustomerName,
    W1870584_CustomerSurname,
    W1870584_CustomerEmail,
    W1870584_CustomerMobileNumber
)
VALUES
    (1001, "Gaja", "Amir", "ga123@gmail.com", 0728596852),
    (1002, "Soya", "Pathushan", "mayoo@gmail.com", 0713441258),
    (1003, "Gaja", "Warrior", "wagmail.com", 0756596852),
    (1004, "Mars", "Dan", "md@gmail.com", 0768741258),
    (1005, "Pollo", "Patterson", "pp@gmail.com", 0768596852),
    (1006, "Gajanan", "Lizara", "gh@gmail.com", 0788741258),
    (1007, "Marina", "Mall", "shan@gmail.com", 0778596852),
    (1008, "Gajavarshan", "Abi", "mohan@gmail.com", 0777741258),
    (1009, "Kiri", "Patler", "ranjan@gmail.com", 0778596852),
    (1010, "Dialog", "Pattern", "raju@gmail.com", 0768741258);
```

While Code

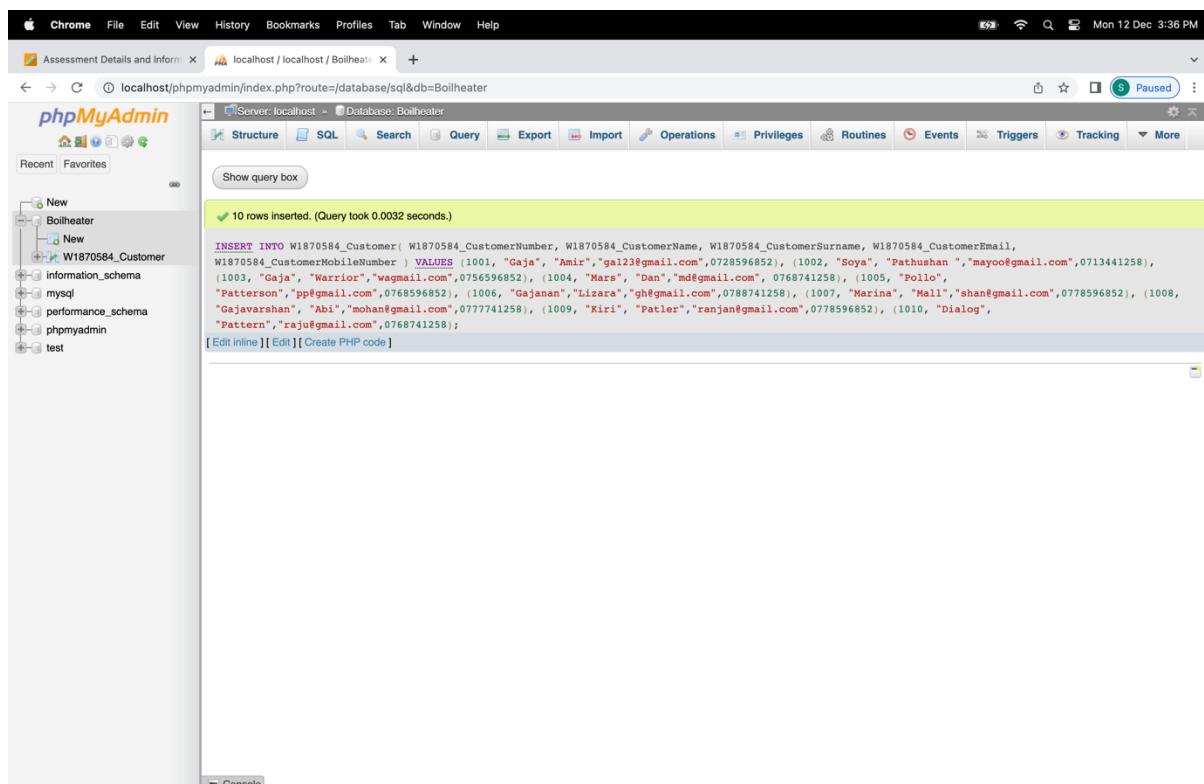


The screenshot shows the phpMyAdmin interface in a web browser. The left sidebar shows the database structure with a 'Boilheater' database selected, containing tables like 'W1870584_Customer'. The main area is titled 'Run SQL query/queries on database Boilheater:' and contains the following SQL code:

```
1 INSERT INTO W1870584_Customer(
2     W1870584_CustomerNumber,
3     W1870584_CustomerName,
4     W1870584_CustomerSurname,
5     W1870584_CustomerEmail,
6     W1870584_CustomerMobileNumber
7 )
8
9 VALUES
10      (1001, "Gaja", "Amir", "gal23@gmail.com", 0728596852),
11      (1002, "Soya", "Pathushan ", "mayoo@gmail.com", 0713441258),
12      (1003, "Gaja", "Warrior", "wagmail.com", 0756596852),
13      (1004, "Mars", "Dan", "md@gmail.com", 0768741258),
14      (1005, "Pollo", "Patterson", "pp@gmail.com", 0768596852),
15      (1006, "Gajanan", "Lizara", "gh@gmail.com", 0788741258),
16      (1007, "Marina", "Mall", "shan@gmail.com", 0778596852),
```

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', 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks'. A 'Go' button is located at the far right.

After Execution



The screenshot shows the same phpMyAdmin interface after the query has been executed. The message '10 rows inserted. (Query took 0.0032 seconds.)' is displayed in a green box at the top. Below it, the same SQL insert statement is shown again, indicating it was successful. The left sidebar remains the same, showing the 'Boilheater' database structure.

Final Output

The screenshot shows the phpMyAdmin interface for the 'Boilheater' database. The left sidebar lists databases like 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The 'W1870584_Customer' table is selected. The main area displays 10 rows of customer data:

	W1870584_CustomerNumber	W1870584_CustomerName	W1870584_CustomerSurname	W1870584_CustomerEmail	W1870584_CustomerMobileNumber
<input type="checkbox"/>	1001	Gaja	Amir	ga123@gmail.com	728596852
<input type="checkbox"/>	1002	Soya	Pathushan	mayoo@gmail.com	713441258
<input type="checkbox"/>	1003	Gaja	Warrior	wagmail.com	756596852
<input type="checkbox"/>	1004	Mars	Dan	md@gmail.com	768741258
<input type="checkbox"/>	1005	Pollo	Patterson	pp@gmail.com	768596852
<input type="checkbox"/>	1006	Gajanan	Lizara	gh@gmail.com	788741258
<input type="checkbox"/>	1007	Marina	Mall	shan@gmail.com	778596852
<input type="checkbox"/>	1008	Gajavarshan	Abi	mohan@gmail.com	777741258
<input type="checkbox"/>	1009	Kiri	Paller	ranjan@gmail.com	778596852
<input type="checkbox"/>	1010	Dialog	Pattern	raju@gmail.com	768741258

Before Create Table for Payment

The screenshot shows the phpMyAdmin interface for the 'Boilheater' database. The left sidebar lists databases like 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The 'W1870584_Customer' table is selected. The main area shows the table structure with 1 row and 1 column.

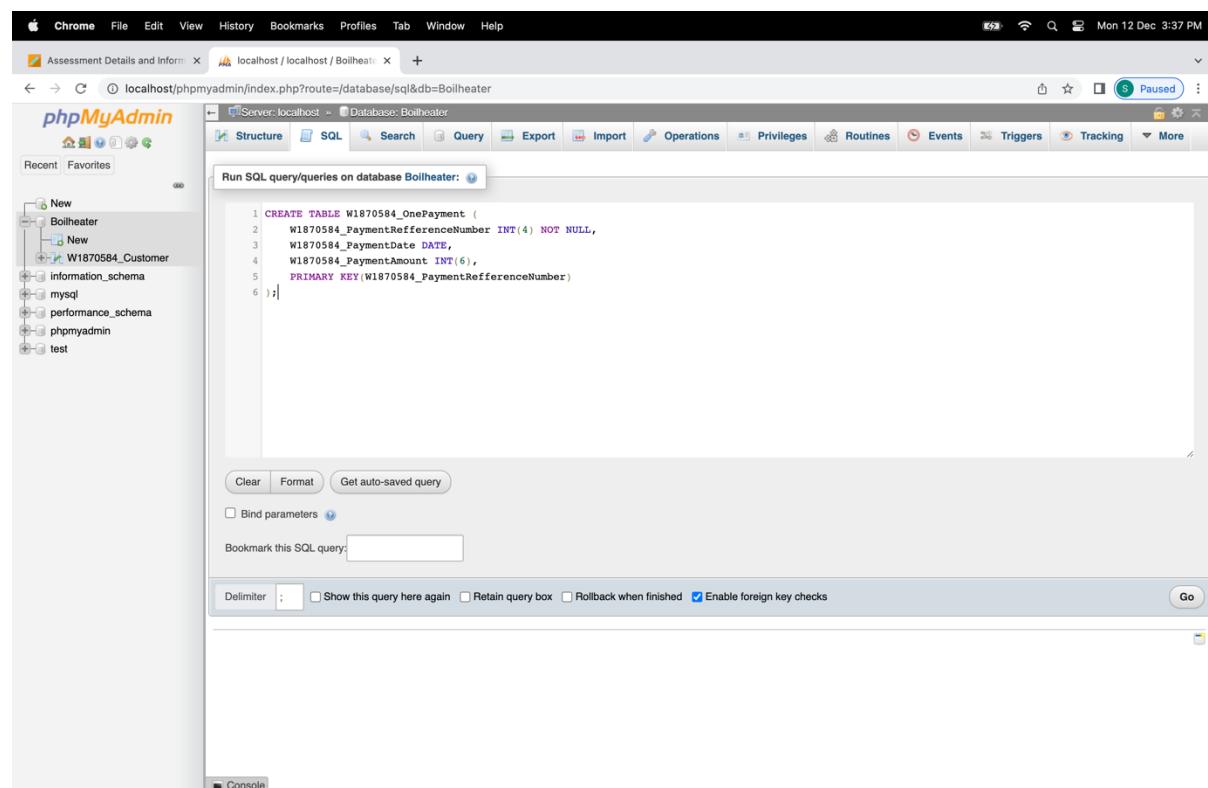
Table	Action	Rows	Type	Collation	Size	Overhead
W1870584_Customer	Sum	10	InnoDB	utf8mb4_general_ci	16.0 Kib	-

Creating Table for Payment

Code:

```
CREATE TABLE W1870584_OnePayment (
    W1870584_PaymentRefferenceNumber INT(4) NOT NULL,
    W1870584_PaymentDate DATE,
    W1870584_PaymentAmount INT(6),
    PRIMARY KEY(W1870584_PaymentRefferenceNumber)
);
```

While Code



After Execution

The screenshot shows the phpMyAdmin interface after executing a SQL query. The left sidebar shows the database structure with a new database named 'Boilheater' containing tables 'W1870584_Customer' and 'W1870584_OnePayment'. The right panel displays the results of a query to create a table:

```
CREATE TABLE W1870584_OnePayment ( W1870584_PaymentReferenceNumber INT(4) NOT NULL, W1870584_PaymentDate DATE, W1870584_PaymentAmount INT(6), PRIMARY KEY(W1870584_PaymentReferenceNumber) );
```

The status bar at the bottom indicates the query took 0.0115 seconds.

Final Output

The screenshot shows the phpMyAdmin interface displaying the structure of the 'W1870584_OnePayment' table. The table has three columns: 'W1870584_PaymentReferenceNumber' (int(4)), 'W1870584_PaymentDate' (date), and 'W1870584_PaymentAmount' (int(6)). The 'W1870584_PaymentReferenceNumber' column is defined as the primary key. The 'Indexes' section shows a single index named 'PRIMARY' on the 'W1870584_PaymentReferenceNumber' column. The 'Information' section provides details about the table's storage: Data size is 16.0 KIB, Format is dynamic, and Collation is utf8mb4_general_ci. The table contains 0 rows.

Insert Values for Payment

Code:

```
INSERT into W1870584_OnePayment(
W1870584_PaymentReferenceNumber,
W1870584_PaymentDate,
W1870584_PaymentAmount
)
```

Values

```
(1001, '2022-10-21', '45'),
(1002, '2022-11-19', '90'),
(1003, '2022-05-18', '35'),
(1004, '2022-06-07', '45'),
(1005, '2021-12-29', '55'),
(1006, '2022-09-30', '150'),
(1007, '2022-10-22', '25'),
(1008, '2022-12-05', '65'),
(1009, '2022-10-17', '75'),
(1010, '2022-01-05', '98');
```

While Code

The screenshot shows the phpMyAdmin interface in a web browser. The left sidebar displays the database structure for 'Boilheater', including tables like 'Customer' and 'OnePayment'. The main area shows the SQL query:

```
1 INSERT into W1870584_OnePayment(
2 W1870584_PaymentReferenceNumber,
3 W1870584_PaymentDate,
4 W1870584_PaymentAmount
5 )
6
7 Values
8 (1001, '2022-10-21', '45'),
9 (1002, '2022-11-19', '90'),
10 (1003, '2022-05-18', '35'),
11 (1004, '2022-06-07', '45'),
12 (1005, '2021-12-29', '55'),
13 (1006, '2022-09-30', '150'),
14 (1007, '2022-10-22', '25'),
15 (1008, '2022-12-05', '65'),
16 (1009, '2022-10-17', '75'),
```

Below the query, there are several 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'.

After Execution

The screenshot shows the phpMyAdmin interface after executing an SQL query. The database selected is 'Boilheater'. The query executed was:

```
INSERT into W1870584_OnePayment( W1870584_PaymentReferenceNumber, W1870584_PaymentDate, W1870584_PaymentAmount ) Values (1001, '2022-10-21', '45'), (1002, '2022-11-19', '90'), (1003, '2022-05-18', '35'), (1004, '2022-06-07', '45'), (1005, '2021-12-29', '55'), (1006, '2022-09-30', '150'), (1007, '2022-10-22', '25'), (1008, '2022-12-05', '65'), (1009, '2022-10-17', '75'), (1010, '2022-01-05', '98');
```

The message at the top indicates "10 rows inserted. (Query took 0.0030 seconds.)".

Final Output

The screenshot shows the phpMyAdmin interface displaying the contents of the 'W1870584_OnePayment' table. The table has three columns: 'W1870584_PaymentReferenceNumber', 'W1870584_PaymentDate', and 'W1870584_PaymentAmount'. The data is as follows:

	W1870584_PaymentReferenceNumber	W1870584_PaymentDate	W1870584_PaymentAmount
<input type="checkbox"/>	1001	2022-10-21	45
<input type="checkbox"/>	1002	2022-11-19	90
<input type="checkbox"/>	1003	2022-05-18	35
<input type="checkbox"/>	1004	2022-06-07	45
<input type="checkbox"/>	1005	2021-12-29	55
<input type="checkbox"/>	1006	2022-09-30	150
<input type="checkbox"/>	1007	2022-10-22	25
<input type="checkbox"/>	1008	2022-12-05	65
<input type="checkbox"/>	1009	2022-10-17	75
<input type="checkbox"/>	1010	2022-01-05	98

The list should be restricted as follows: it should only show those customers whose surnames starts with the 3 letters 'Pat' and only the one-off payments for which the paid amount is £80 and under.

SQL Query to select the records with the given conditions.

Before Code

	W1870584_CustomerNumber	W1870584_CustomerName	W1870584_CustomerSurname	W1870584_CustomerEmail	W1870584_CustomerMobileNumber
<input type="checkbox"/>	1001	Gaja	Amir	ga123@gmail.com	728596852
<input type="checkbox"/>	1002	Soya	Pathushan	mayoo@gmail.com	713441258
<input type="checkbox"/>	1003	Gaja	Warrior	wagmail.com	756596852
<input type="checkbox"/>	1004	Mars	Dan	md@gmail.com	768741258
<input type="checkbox"/>	1005	Pollo	Patterson	pp@gmail.com	768596852
<input type="checkbox"/>	1006	Gajanan	Lizara	gh@gmail.com	788741258
<input type="checkbox"/>	1007	Marina	Mall	shan@gmail.com	778596852
<input type="checkbox"/>	1008	Gajavarshan	Abi	mohan@gmail.com	777741258
<input type="checkbox"/>	1009	Kiri	Patler	ranjan@gmail.com	778596852
<input type="checkbox"/>	1010	Dialog	Pattern	raju@gmail.com	768741258

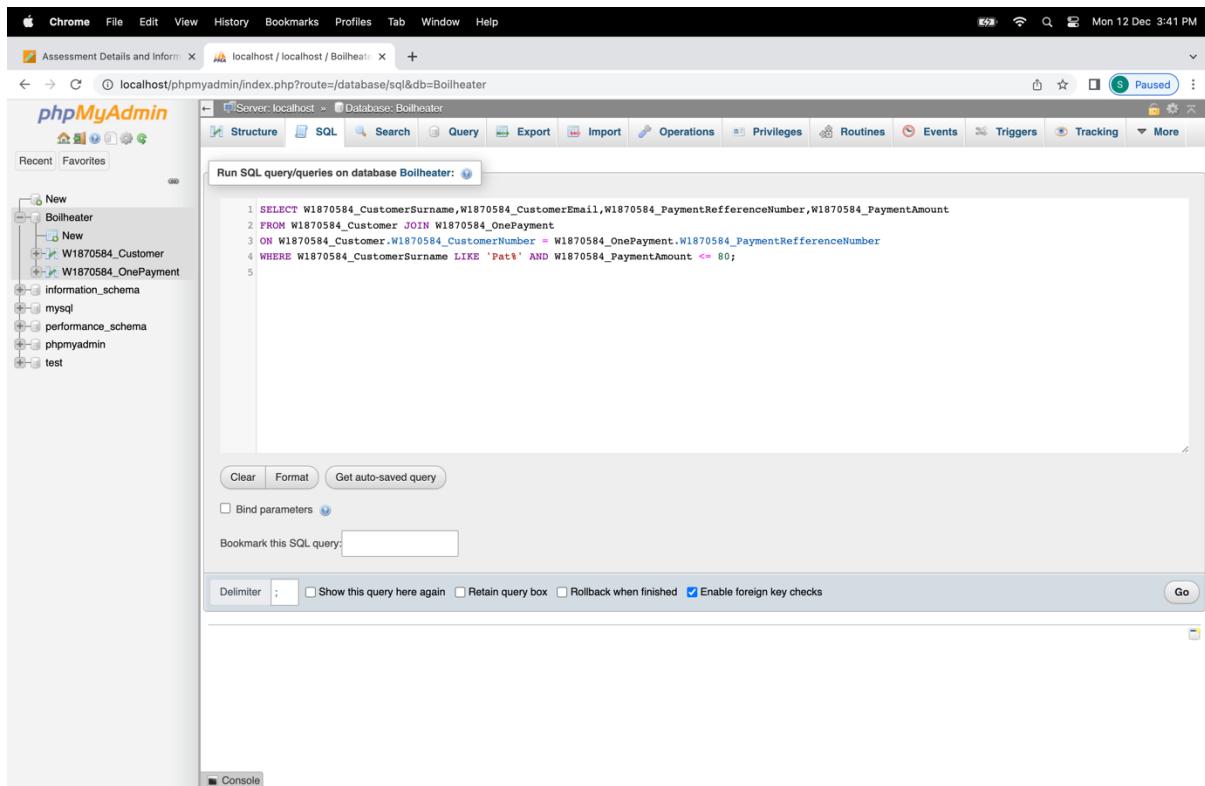
Code:

```

SELECT
W1870584_CustomerSurname,W1870584_CustomerEmail,W1870584_PaymentRefference
Number,W1870584_PaymentAmount
FROM W1870584_Customer JOIN W1870584_OnePayment
ON W1870584_Customer.W1870584_CustomerNumber =
W1870584_OnePayment.W1870584_PaymentRefferenceNumber
WHERE W1870584_CustomerSurname LIKE 'Pat%' AND W1870584_PaymentAmount <=
80;

```

While Code

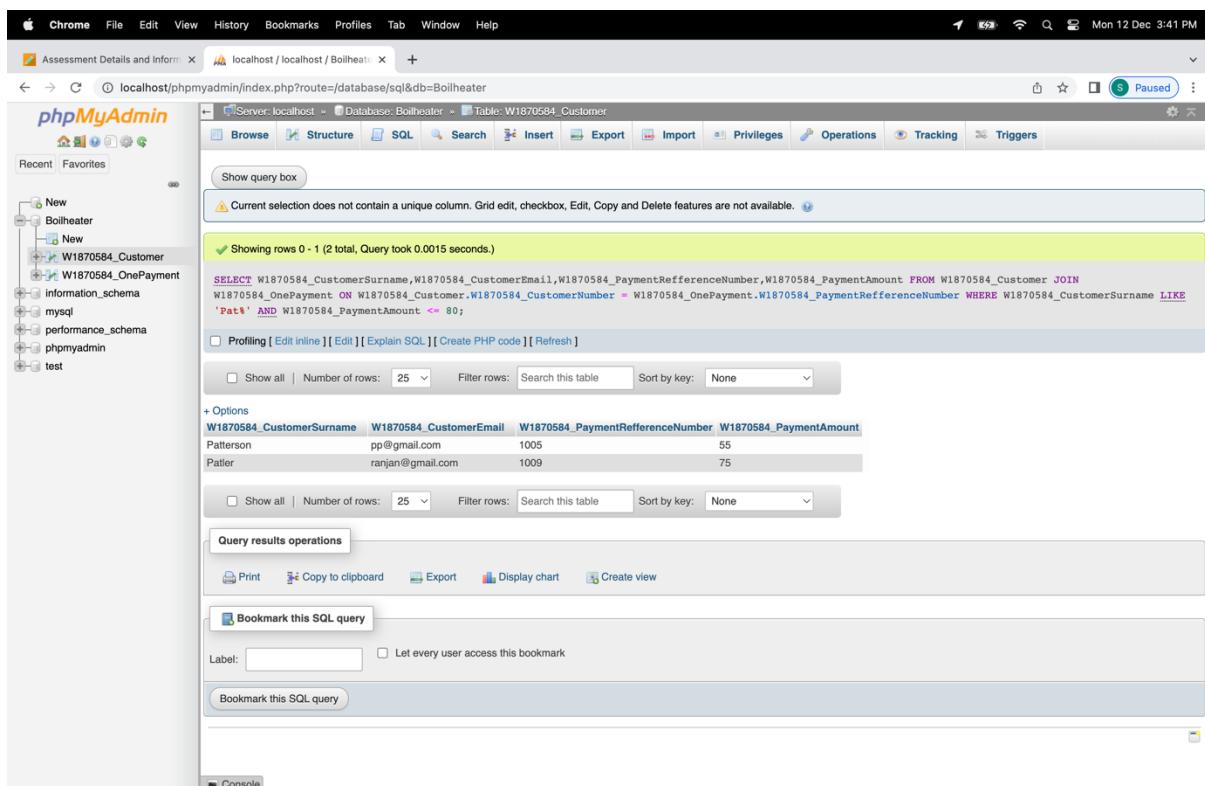


The screenshot shows the phpMyAdmin interface in a web browser. The left sidebar lists databases: New, Boilheater, New, W1870584_Customer, W1870584_OnePayment, information_schema, mysql, performance_schema, phpyadmin, and test. The main area shows the SQL tab with the following query:

```
1 SELECT W1870584_CustomerSurname,W1870584_CustomerEmail,W1870584_PaymentReferenceNumber,W1870584_PaymentAmount
2 FROM W1870584_Customer JOIN W1870584_OnePayment
3 ON W1870584_Customer.W1870584_CustomerNumber = W1870584_OnePayment.W1870584_PaymentReferenceNumber
4 WHERE W1870584_CustomerSurname LIKE 'Pat%' AND W1870584_PaymentAmount <= 80;
5
```

Below the query, there are buttons for Clear, Format, Get auto-saved query, Bind parameters, and Bookmark this SQL query. At the bottom, there are options for Delimiter, Show this query here again, Retain query box, Rollback when finished, and Enable foreign key checks. A 'Go' button is at the far right.

After Execution



The screenshot shows the phpMyAdmin interface after executing the SQL query. The left sidebar is the same as the previous screenshot. The main area shows the results of the query execution:

Showing rows 0 - 1 (2 total, Query took 0.0015 seconds.)

```
SELECT W1870584_CustomerSurname,W1870584_CustomerEmail,W1870584_PaymentReferenceNumber,W1870584_PaymentAmount FROM W1870584_Customer JOIN
W1870584_OnePayment ON W1870584_Customer.W1870584_CustomerNumber = W1870584_OnePayment.W1870584_PaymentReferenceNumber WHERE W1870584_CustomerSurname LIKE
'Pat%' AND W1870584_PaymentAmount <= 80;
```

Below the query results, there is a table with two rows:

W1870584_CustomerSurname	W1870584_CustomerEmail	W1870584_PaymentReferenceNumber	W1870584_PaymentAmount
Patterson	pp@gmail.com	1005	55
Patler	ranjan@gmail.com	1009	75

At the bottom, there are buttons for Print, Copy to clipboard, Export, Display chart, Create view, and a 'Bookmark this SQL query' section with a 'Label:' input field and a 'Let every user access this bookmark' checkbox.

QUESTION 7

Create a **Comparative Analysis Table** to compare and contrast the **MySQL** Relational Database Management System and the **MongoDB** document-oriented database program with a view to inform the decision-making of the management of a firm.

Your comparison table should present **five clear comparison criteria** (or informative decision factors) as rows and have a column for MySQL and a column for MongoDB so that you can compare them side by side.

You could consider areas such as schemas, data consistency, storage, performance, workload, infrastructures, security, etc.

You need to reference your findings by making accurate citations to reliable external sources, using either the “Cite Them Right Harvard” or “Westminster Harvard” referencing systems.

You also need to include a list of references right at the end of your report.

Criteria	MySQL	MongoDB
Data structure	Data is stored and represented in JSON (JavaScript Object Notation) like documents.	Data is stored as a table cell structure with rows and columns.
Schema	It requires a schema definition for the tables in the databases. So, the schema can't be altered. Only the inputs confirmed to the given schema are accepted.	It stores data in collections without enforced schema.
Security	It is safer to use MySQL since it has data structure with a higher level of consistency in comparison to other databases.	Because there is no fixed structure, there is the possibility of inconsistencies and data security concerns.
Architecture	Based on the client-server architecture, it provides optimized storage performance as well as multithreaded processing.	Based on the Nexus Architecture, it offers higher levels of flexibility and availability than its predecessor.
Performance	In MySQL, joins between multiple tables that have been indexed appropriately are optimized for high performance	The write performance of MongoDB is optimized for writing, which means that the speed performance of MongoDB is much higher than that of MySQL.
User Friendliness	In comparison to MongoDB, MySQL is a bit more complex due to the schema of its tables, foreign keys, normalization, etc.	As a data storage solution, MongoDB is attractive to developers since it offers ease of use and a clear understanding of the storage philosophy.

References

1. Scaler Academy (August 16, 2022)

Available at : <https://www.interviewbit.com/blog/mongodb-vs-mysql/#:~:text=In%20simpler%20words%2C%20MongoDB%20is,specific%20row%2Dtable%20based%20schema>.
[2022.12.10]

2. Kinsta, Durga Prasad Acharya , (November 25, 2022)

Available at : <https://kinsta.com/blog/mongodb-vs-mysql/#:~:text=But%20if%20you%20have%20a,MongoDB%20offers%20more%20schema%20flexibility>.
[2022.12.08]

3. Mongo DB

Available at: <https://www.mongodb.com/compare/mongodb-mysql>.
[2022.12.09]