

6CC546 – DATABASE FUNDAMENTALS

KHIZAR MUKHTIAR

ID – 100559119



Table of Contents

1.	Introduction.....	1
2.	Normalization.....	3
3.	Implementation.....	5
4.	Querying.....	14
5.	Optimization.....	20
6.	Securing Database	22
7.	Web Interface.....	25
8.	Conclusion.....	37
9.	References.....	38

1. Introduction

With ever-increasing active users on the Internet (Internet Live Stats, 2021), managing proper data storage is of utmost importance. Thus, paving the way for databases, which is a collection of data on a storage organized in a very specific fashion. The type of database that is most often used, and will be implemented in this project, is a relational database that consists of tables keeping similar information. The tables are further broken down in a distinct way, made up of columns and rows (EssentialSQL, 2014).

Prior to the development and deployment of any database, the most important prerequisite is to define a good data model that can be represented as being easily understandable; large data changes are scalable; the performance is predictable; and lastly changes in requirements are adaptable but independent of the previously mentioned points (Kejser, 2011). The first step to achieve this endeavor is to create an Entity Relationship Model (ERM), derived from a higher-level abstract view of the database known as the conceptual model that is viewed as containing the logical structure of the whole database, providing the complete overview of data requirements set forth by the specific domain of the organization and does not consider the data storage details (Connolly & Begg, 2014, Ch. 2). This report starts off with a similar pattern by first identifying and choosing entities for "*Stepping into History*", which are the real-world objects that can have information stored with regards to them. Followed by selecting attributes, that define the properties or characteristics of entities. Finally, after uniquely identifying each individual entity with a primary key, the relationship between and among entities are described (Biscobing, 2019). The following Entity Relation Diagram (ERD) demonstrates a simple visualization of entities and the relationship between them.

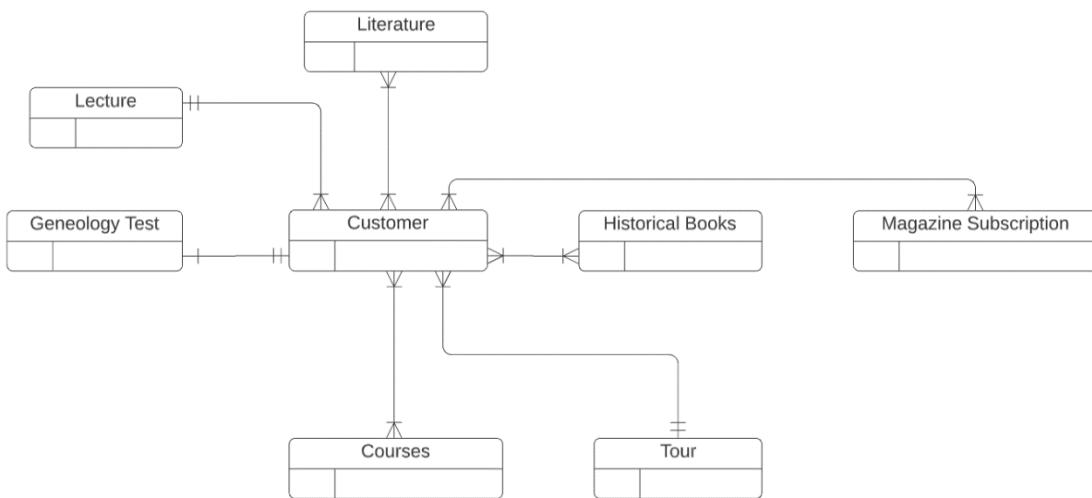


Figure 1.1 – Entity Relationship Diagram

The ERD above gives a simple abstract overview of all the possible entities from the domain set forth by the case study and the relationship between them, devoid of any attributes and primary/foreign keys, that will be covered subsequently in the complete Entity Relationship Model (ERM).

The ERM, based on the initial conceptual design of the ERD, can be seen in the figure 1.2 below as being much neater and elaborative, with all the possible attributes of the entities and their respective identities and relationships.

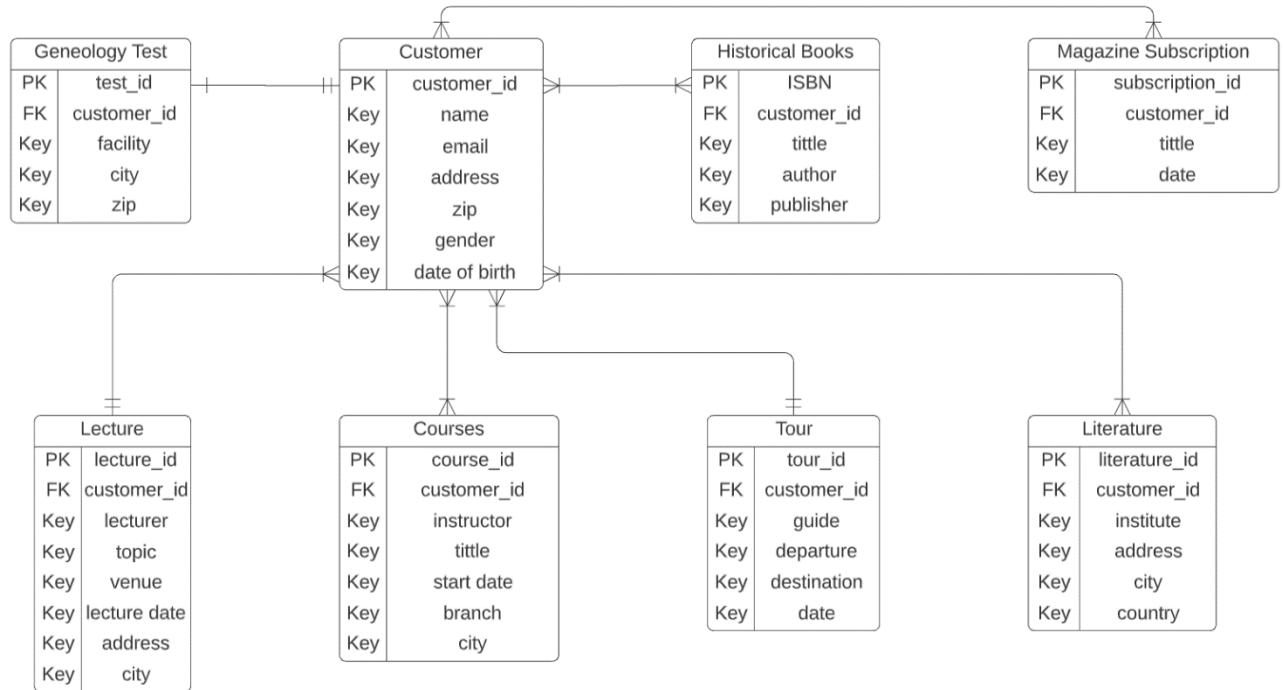


Figure 1.2 – Entity Relationship Model (ERM)

The ERM depicted in figure 1.2 clearly shows that all the relationships revolve around the *Customer* entity. The reason for this is the individuality of each entity and its unique relationship with the *Customer*. The case study is mainly aimed towards a particular niche of providing different services in the historiography domain. As seen in the model, a customer can have one genealogy test, but a particular test can have only one customer, this way any type of false positives may be avoided since there would not be any tests overlapping. Somewhat similarly, a customer can either be on one tour or at one lecture at any given time, but a tour or lecture can have numerous customers. Furthermore, a customer can buy one or many historical books, sign up/subscribe to one or multiple courses or magazines, and ask for the location of a single or several historical literatures and vice versa.

The next step in shaping a good database model, prior to development, is the normalization process which is a method used to help minimize data duplication (Isaac, 2021) and is covered in the next section.

2. Normalization

Upon designing a database, a set of formal criteria and rules are applied upon it. This method is known as database normalization and is divided into several different levels called the Normal Forms, the first four are, First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce Codd Normal Form (BCNF).

1NF is used to get rid of any repeating attributes and duplicate tuples. To further break it down, for a database design to satisfy this stage it must not have multiple values in the same cell. Moreover, the unique identification of each row is assured by having a primary key in the relation. Hence, to achieve 1NF in a database design that contains the issue addressed, the original design is broken down into separate relations under new names. The attributes with non-repeating tuples are placed in one relation while for the repeating ones a new relation is generated and are put there (Jayaram, 2018). Looking at the ERM in figure 1.2, there might be some repeating values in the relation *Tour*, for example:

<u>tour_id</u>	<u>customer_id</u>	guide	destination	date
001	01	Samuel Jackson	Taj Mahal	6/11/2021
	02			
	03			
002	10	Marylin Monroe	Machu Picchu	21/11/2021
	11			
	12			

Table 2.1

Looking at the dummy data in the example above, it is quite evident that there are multiple values in a single cell. To circumvent this issue, the parent relation is broken down into necessary child relations. Moreover, it is possible that two or more guides can have the same name, therefore, another new relation for guide is also required:

<u>guide_id</u>	name
01	Samuel Jackson
02	Marylin Monroe

Table 2.2 – Guide relation

<u>tour_id</u>	<u>guide_id</u>	destination	date
001	01	Taj Mahal	6/11/2021
002	02	Machu Picchu	24/11/2021

Table 2.3 – Tour 2 relation

Since a primary key is always unique to one row, hence all the tours will be designated to one particular ID. This ID will, in turn, link with the customers' foreign key in another table called *booked_tours*, as seen in figure 2.3. The same issue holds true for the relations, *Lecture*, *Courses*, and *Historical books* and can be thwarted in a similar way.

<u>booked_tour_id</u>	<u>tour_id</u>	<u>customer_id</u>
001	01	04
002	02	05
003	02	06

Table 2.3 – Booked Tour relation

Subsequently, 1NF becomes the first step for a relation to fulfil 2NF. Another most important step is to check if there are any partial dependencies of non-primary key attributes on the composite primary key attribute (if the primary key has a single attribute, then 2NF is satisfied automatically). If this condition is true, then the parent relation is broken down into child relations to conform that all non-key attributes in any relation are fully dependent on the whole composite primary key and not just part of it. The ERM demonstrated in figure 1.2, there are no composite primary keys in any relation, therefore satisfying the condition for 2NF.

Prior to 3NF, the relation must first satisfy 2NF. Furthermore, the next step in this stage is to identify and remove transitive dependency from a relation by decomposing the original relation into child relations and making sure that no data is lost during the process. The mathematical logic behind this concept means that if A implies B and B implies C, then A must imply C (Jayaram, 2018). Looking back at the database design for the case study, all the non-primary key attributes are fully dependent on one atomic primary key, thus, meeting the condition for 3NF.

Often termed as a stronger version of 3NF, BCNF requires the fulfilment of the previous stage and every determinant in the relation must be a candidate key (in simpler terms, a primary key). The determinant is defined as the attribute that determines the values of another attribute (Barnes, 2018). The primary key in all the relations of the database model in the case study determines the value for other attributes, hence, conforming to BCNF.

The database can now be developed based on the ERM and its normalization.

3. Implementation

MySQL Workbench will be used for the development of the database for this project since it integrates SQL (Structured Query Language). All the tables in SQL will be based on the ERM in figure 1.2, with only a few more additional tables for the attributes; *Lecturer*, *Instructor*, *Guide*, and other tables for booking of tours, books, lectures, and courses. The new tables will list all the relevant information of the mentioned attributes.

```
1 • CREATE TABLE customer(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     full_name VARCHAR(30) NOT NULL,
4     address VARCHAR(30) NOT NULL,
5     email VARCHAR(60) NOT NULL,
6     zip MEDIUMINT UNSIGNED NOT NULL,
7     gender ENUM('M', 'F'),
8     date_of_birth DATE NULL);
```



```
1 • INSERT INTO customer(id, full_name, address, email, zip, gender, date_of_birth) VALUES
2     (NULL, 'Christopher Jones', '347 Cedar St', 'christopherjones@bp.com', '30044', 'M', '1958-09-11'),
3     (NULL, 'Matthew Martinez', '602 Main Place', 'matthewmartinez@ge.com', '92336', 'M', '1961-09-04'),
4     (NULL, 'Melissa Moore', '463 Park Rd', 'melissamoore@aramark.com', '08701', 'M', '1967-08-27'),
5     (NULL, 'Melissa Brown', '712 View Ave', 'melissabrown@verizon.com', '77084', 'F', '1948-06-14'),
6     (NULL, 'Jennifer Thomas', '231 Elm St', 'jenniferthomas@aramark.com', '78572', 'F', '1998-03-14'),
7     (NULL, 'Stephanie Martinez', '386 Second St', 'stephaniemartinez@albertsons.com', '08701', 'M', '1998-01-24'),
8     (NULL, 'Daniel Williams', '107 Pine St', 'danielwilliams@tjx.com', '77449', 'F', '1985-07-20'),
9     (NULL, 'Lauren Anderson', '13 Maple Ave', 'laurenanderson@pepsi.com', '92503', 'F', '1973-09-09'),
10    (NULL, 'Michael Jackson', '818 Pine Ave', 'michaeljackson@disney.com', '78572', 'F', '1951-03-03'),
11    (NULL, 'Ashley Johnson', '874 Oak Ave', 'ashleyjohnson@boeing.com', '91331', 'F', '1977-05-18'),
12    (NULL, 'Brittany Thomas', '187 Maple Ave', 'brittanythomas@walmart.com', '78521', 'F', '1986-10-22'),
13    (NULL, 'Tiffany Smith', '123 Lake St', 'tiffanysmith@ups.com', '78521', 'F', '1950-06-16'),
14    (NULL, 'Lauren Wilson', '942 Fifth Ave', 'laurenwilson@target.com', '78572', 'M', '1965-12-26'),
15    (NULL, 'Justin Smith', '844 Lake Ave', 'justinsmith@boeing.com', '30044', 'F', '1956-03-16'),
16    (NULL, 'Jessica Garcia', '123 Pine Place', 'jessicagarcia@toyota.com', '92336', 'F', '1996-08-05'),
17    (NULL, 'Matthew Jackson', '538 Cedar Ave', 'matthewjackson@bp.com', '77449', 'M', '1996-02-26'),
18    (NULL, 'Stephanie Thomas', '804 Fourth Place', 'stephaniethomas@apple.com', '78521', 'F', '1988-08-26'),
19    (NULL, 'Jessica Jackson', '235 Pine Place', 'jessicajackson@aramark.com', '60629', 'F', '1991-07-22'),
20    (NULL, 'James Martinez', '831 Oak St', 'jamesmartinez@kroger.com', '78521', 'F', '1987-12-22'),
21    (NULL, 'Christopher Robinson', '754 Cedar St', 'christopherrobinson@ibm.com', '78577', 'F', '1972-06-25');
```

Figure 3.1 – Creating the customer table and inserting data into it.

As seen in figure 3.1, the customer table is created to take in the values for name, address, email, zip, gender, and date of birth with a specific type. Dummy data is then inserted respectively in these attributes.

```

1 • CREATE TABLE instructor(
2   id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3   full_name VARCHAR(30) NOT NULL,
4   email VARCHAR(60) NOT NULL,
5   gender ENUM('M','F'),
6   speciality VARCHAR(60) NOT NULL);
```

```

1 INSERT INTO instructor(id, full_name, email, gender, speciality) VALUES
2 (NULL, 'Samuel Jacson', 'samuel@gmail.com', 'M', 'Analysis and Use of Sources'),
3 (NULL, 'Miranda Kerr', 'kerr@hotmail.com', 'F', 'Chronology'),
4 (NULL, 'Mark Zuckerberg', 'zuckmark@gmail.com', 'M', 'Perspectives and Interpretations'),
5 (NULL, 'Monica Geller', 'geller@yahoo.com', 'M', 'Anthropology'),
6 (NULL, 'Leonardo DeCaprio', 'caprio@gmail.com', 'M', 'Sociology'),
7 (NULL, 'Warren Buffet', 'w.buffet@yahoo.com', 'M', 'Prehistoric Archaeology');
```

Figure 3.2 – Creating the instructor table and inserting data into it.

```

1 • CREATE TABLE guide(
2   id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3   full_name VARCHAR(30) NOT NULL,
4   email VARCHAR(60) NOT NULL,
5   gender ENUM('M','F'));
```

```

1 INSERT INTO guide(id, full_name, email, gender) VALUES
2 (NULL, 'Christopher Wallace', 'christopher@bp.com', 'M'),
3 (NULL, 'Thomas Shelby', 'shelby@peaky.com', 'M'),
4 (NULL, 'Micheal Scott', 'micheal@mifflin.com', 'M'),
5 (NULL, 'David Wallace', 'david@dunder.com', 'M'),
6 (NULL, 'Lena Grey', 'lena@gmail.com', 'F'),
7 (NULL, 'Stephanie Marvel', 'stephanie@comicon.com', 'F'),
8 (NULL, 'Bruce Wayne', 'wayne@billionaire.com', 'M'),
9 (NULL, 'Lauren Potter', 'laurengett@hogwarts.com', 'M'),
10 (NULL, 'Richard Nixon', 'nixon@usa.com', 'M'),
11 (NULL, 'Pheobe Buffet', 'pheobe@centralperk.com', 'F'),
12 (NULL, 'Brittany Amber', 'brittany@derby.com', 'F'),
13 (NULL, 'Elaine Benes', 'elaine@amazon.com', 'F'),
14 (NULL, 'Lauren James', 'tiffanyj@netflix.com', 'F');
```

Figure 3.3 – Creating the guide table and inserting data into it.

```

1 • CREATE TABLE lecturer(
2   id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3   full_name VARCHAR(30) NOT NULL,
4   email VARCHAR(60) NOT NULL,
5   gender ENUM('M','F'),
6   speciality VARCHAR(60) NOT NULL);
```



```

1 INSERT INTO lecturer(id, full_name, email, gender, speciality) VALUES
2 (NULL, 'Nial Ferguson', 'nial@gmail.com', 'M', 'International History'),
3 (NULL, 'Timothy D. Snyder', 'timothysyder@hotmail.com', 'M', 'Central and Eastern Europe History'),
4 (NULL, 'Simon Schama', 'simon@gmail.com', 'M', 'Art, Dutch, and Jewish History'),
5 (NULL, 'Eamon Duffy', 'eamon@yahoo.com', 'M', 'Religious History of Britain'),
6 (NULL, 'Anthony Beevor', 'anthony@gmail.com', 'M', 'Military History'),
7 (NULL, 'Robert C. Allen', 'robertallen@yahoo.com', 'M', 'Economic History'),
8 (NULL, 'Romila Thapar', 'thaparromila@gmail.com', 'F', 'Ancient India');
```

Figure 3.4 – Creating the lecturer table and inserting data into it.

Similarly, as mentioned earlier, the instructor, guide, and lecturer tables are created, and dummy data is inserted accordingly. These tables can be seen in figures; 3.2, 3.3, 3.4 respectively.

```

1 • CREATE TABLE magazine_subscription(
2   id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3   customer_id INT UNSIGNED NOT NULL,
4   FOREIGN KEY (customer_id) REFERENCES customer(id),
5   tittle VARCHAR(30) NOT NULL,
6   start_date DATE NOT NULL);
```



```

1 INSERT INTO magazine_subscription(id, customer_id, tittle, start_date) VALUES
2 (NULL, 1, 'Ancient Gallery', '2021-11-20'),
3 (NULL, 3, 'Sculptures', '2021-12-2'),
4 (NULL, 4, 'Ancient Ornaments', '2021-9-5'),
5 (NULL, 5, 'Mughals Kitchen', '2021-8-9'),
6 (NULL, 10, 'Greek Architecture', '2021-9-15'),
7 (NULL, 14, 'Evolution', '2021-12-18'),
8 (NULL, 14, 'The Silk Road', '2021-11-24'),
9 (NULL, 19, 'Tsars', '2022-01-20'),
10 (NULL, 8, 'Evolution of Fashion', '2021-5-20');
```

Figure 3.5 – Magazine subscription table and its data.

As seen in figure 3.5, the magazine subscription is linked with a customer via a foreign key and any one customer can have multiple subscriptions and vice versa.

```

1 • CREATE TABLE literature(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     customer_id INT UNSIGNED NOT NULL,
4     FOREIGN KEY (customer_id) REFERENCES customer(id),
5     institute VARCHAR(30) NOT NULL,
6     address VARCHAR(30) NOT NULL,
7     city VARCHAR(30) NOT NULL,
8     country VARCHAR(30) NOT NULL);

```

```

1   INSERT INTO literature VALUES
2   (NULL, 3, "NYU", "NY 10003", "New York City", "USA"),
3   (NULL, 5, "Cairo University", "12613 Cairo Uni Road", "Cairo", "Egypt")
4

```

Figure 3.6 – Literature table and its data.

Like the relation between magazine subscription and customer, the literature table works in the exact same way. Showing location of a specific scripture or literary work to a customer. The table and its dummy data can be seen in the figure 3.6 above.

```

1 • CREATE TABLE genology_test(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     customer_id INT UNSIGNED NOT NULL,
4     FOREIGN KEY (customer_id) REFERENCES customer(id),
5     facility VARCHAR(30) NOT NULL,
6     city VARCHAR(30) NOT NULL,
7     zip MEDIUMINT UNSIGNED NOT NULL);

```

```

1 • SELECT * FROM genology_test

```

	id	customer_id	facility	city	zip
▶	1	2	Umbrella Corp	London	2610
	2	5	Umbrella Corp	London	2610
	3	7	Stark Industries	New York	3650
	4	11	Johnson & Johnsons	Little Rock	4590
	5	1	Umbrella Corp	London	2610
	6	12	Stark Industries	New York	3650
	7	9	Umbrella Corp	London	2610
*	NULL	NULL	NULL	NULL	NULL

Figure 3.7 – Genealogy test table and its data.

The genealogy test table also includes a customers' foreign key for linking. However, one test can only serve a specific customer. The dummy data is inserted in a similar way and can be seen in figure 3.7 above.

```

1  CREATE TABLE historical_books(
2      id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3      ISBN BIGINT UNSIGNED NOT NULL,
4      tittle VARCHAR(30) NOT NULL,
5      author VARCHAR(30) NOT NULL,
6      publisher VARCHAR(30) NOT NULL);
```

```

1  INSERT INTO historical_books(id, ISBN, tittle, author, publisher) VALUES
2  (NULL, 9780393317558, 'Guns, Germs, and Steel', 'Jared Diamond', 'W. W. Norton'),
3  (NULL, 9781400040063, '1491: New Revelations of the Americas Before Columbus', 'Charles C. Mann', 'Knopf'),
4  (NULL, 9788073417765, 'Genghis Khan and the Making of the Modern World', 'Jack Weatherford', 'Crown and Three Rivers Press'),
5  (NULL, 9780062316097, 'Sapiens: A Brief History of Humankind', 'Yuval Noah Harari', 'HarperCollins Publishers'),
6  (NULL, 9781408839973, 'The Silk Roads: A New History of the World', 'Peter Frankopan', 'Bloomsbury'),
7  (NULL, 9780062397348, 'A People's History of the United States', 'Howard Zinn', 'HarperCollins Publishers'),
8  (NULL, 9780060787288, 'The Crusades: The Authoritative History of the War for the Holy Land', 'Thomas Asbridge', 'Ecco Press'),
9  (NULL, 9780151355372, 'The History of the Decline and Fall of the Roman Empire; Volume 3', 'Edward Gibbon', 'Strahan & Cadell'),
10 (NULL, 9781451651683, 'The Rise and Fall of the Third Reich', 'William L. Shirer', 'Simon & Schuster'),
11 (NULL, 9780140444780, 'The Communist Manifesto', 'Karl Marx and Friedrich Engels', 'Simon & Schuster'),
12 (NULL, 9780691099460, 'Muqaddimah', 'Ibn Khaldun', 'Pantheon Books');
```

Figure 3.8 – Table containing historical books and its data

The table that holds historical books can be seen in figure 3.8 above along with its appropriate data. The same holds true for the tables containing courses, tours, and lectures that have their own designated instructor, guide, and lecturer linked via a foreign key. The tables can be seen in the figures 3.9, 3.10, and 3.11 respectively, below:

```

1 • CREATE TABLE courses(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     customer_id INT UNSIGNED NOT NULL,
4     instructor_id INT UNSIGNED NOT NULL,
5     FOREIGN KEY (customer_id) REFERENCES customer(id),
6     FOREIGN KEY (instructor_id) REFERENCES instructor(id),
7     tittle VARCHAR(30) NOT NULL,
8     start_date DATE NOT NULL,
9     branch VARCHAR(30) NOT NULL,
10    city VARCHAR(30) NOT NULL);
```

```

1 • INSERT INTO courses(id, instructor_id, tittle, start_date, branch, city) VALUES
2  (NULL, 2, 'Ancient Civilization', '2021-10-29', 'B009', 'New York'),
3  (NULL, 5, 'Capitalism', '2021-11-24', 'B007', 'Chicago'),
4  (NULL, 1, 'Ancient Civilization', '2021-10-29', 'B100', 'Scranton'),
5  (NULL, 4, 'Introduction to Anthropology', '2021-12-29', 'B008', 'Sacramento'),
6  (NULL, 6, 'Archaeology Fundamentals', '2022-01-29', 'B001', 'Louiseville'),
7  (NULL, 3, 'History of Monetary Trade', '2021-02-15', 'B006', 'Washington D.C.');
```

Figure 3.9 – Table containing courses and its data

```
1 • CREATE TABLE tour(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     customer_id INT UNSIGNED NOT NULL,
4     guide_id INT UNSIGNED NOT NULL,
5     FOREIGN KEY (customer_id) REFERENCES customer(id),
6     FOREIGN KEY (guide_id) REFERENCES guide(id),
7     departure VARCHAR(30) NOT NULL,
8     destination VARCHAR(30) NOT NULL,
9     departure_date DATE NOT NULL);
```

```
1 • INSERT INTO tour (id, guide_id, departure, destination, departure_date) VALUES
2     (NULL, 1, 'Victoria Station London', 'Taj Mahal', '2021-12-28'),
3     (NULL, 2, 'Clapham Junction London', 'Pyramids of Giza', '2022-1-28'),
4     (NULL, 3, 'Copenhagen Airport', 'Machu Picchul', '2022-3-14'),
5     (NULL, 4, 'Stopera Amsterdam', 'Bagan', '2021-12-31'),
6     (NULL, 5, 'Brandenburg Gate Berlin', 'Angkor Wat', '2021-11-24'),
7     (NULL, 6, 'Bosphorus Istanbul', 'Great Wall of China', '2022-2-28'),
8     (NULL, 7, 'Zytglogge Bern', 'Roman Colosseum', '2022-2-3'),
9     (NULL, 8, 'Field Museum Chicago', 'Acropolis', '2022-5-28'),
10    (NULL, 9, 'Santa Monica Pier Los Angeles', 'Stonehenge', '2021-11-4'),
11    (NULL, 10, 'Moscow Metro', 'Borobudur', '2022-2-14'),
12    (NULL, 11, 'Calgary Tower', 'Mesa Verde', '2021-12-1'),
13    (NULL, 12, 'University of Toronto', 'Terracotta Army', '2021-11-17'),
14    (NULL, 13, 'Madrid Train Station', 'Petra', '2021-10-20');
```

Figure 3.10 – Table containing historical books and its data

```
1 • CREATE TABLE lecture(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     customer_id INT UNSIGNED NOT NULL,
4     lecturer_id INT UNSIGNED NOT NULL,
5     FOREIGN KEY (customer_id) REFERENCES customer(id),
6     FOREIGN KEY (lecturer_id) REFERENCES lecturer(id),
7     topic VARCHAR(30) NOT NULL,
8     venue VARCHAR(30) NOT NULL,
9     lecture_date DATE NOT NULL,
10    address VARCHAR(30) NOT NULL,
11    city VARCHAR(30) NOT NULL);
```

```

1  INSERT INTO lecture(id, lecturer_id, topic, venue, lecture_date, address, city) VALUES
2  (NULL, 2, 'Seljuk Turks', 'Havemeyer Hall', '2021-12-22', '3000 Broadway', 'New York'),
3  (NULL, 1, 'Renaissance', 'Royal Albert Hall', '2022-2-22', 'Kensington', 'London'),
4  (NULL, 2, 'Genghes Khan', 'DR Koncerthuset', '2022-1-15', 'Orestads Blvd', 'Copenhagen'),
5  (NULL, 5, 'World War', 'Tonhalle Dusseldorf', '2021-12-18', 'Ehrenhof 1', 'Dusseldorf'),
6  (NULL, 3, 'Holocaust', 'The Concertgebouw', '2022-9-3', 'Concertgebouwplein 10', 'Amsterdam'),
7  (NULL, 7, 'Pre-Mughal India', 'Wiener Royal Orchester', '2022-9-22', 'Haus der Industrie', 'Vienna'),
8  (NULL, 4, 'The Dark Ages', 'Central Hall', '2021-12-22', '2 W Tollcross', 'Edinburgh '),
9  (NULL, 6, 'Communism', 'Paris Philharmonic', '2021-10-22', '221 Av. Jean Jaures', 'Paris');

```

Figure 3.11 – Table containing all the lectures along with its respective lecturer and other attributes

After storing all the relevant data in its respective designated place, it is time to create tables for bookings. These new tables will determine which customer has booked a lecture, tour, course, or ordered a historical book.

```

1  CREATE TABLE books_order(
2      id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3      historical_books_id INT UNSIGNED NOT NULL,
4      FOREIGN KEY (historical_books_id) REFERENCES historical_books(id),
5      customer_id INT UNSIGNED NOT NULL,
6      FOREIGN KEY (customer_id) REFERENCES customer(id),
7      purchase_date DATE NOT NULL);
8

```

```

1  INSERT INTO books_order(id, historical_books_id, customer_id, purchase_date) VALUES
2  (NULL, 23, 1, '2021-10-29'),
3  (NULL, 24, 2, '2021-11-14'),
4  (NULL, 23, 2, '2021-10-24'),
5  (NULL, 25, 2, '2021-12-28'),
6  (NULL, 25, 2, '2021-09-24'),
7  (NULL, 26, 2, '2021-08-24'),
8  (NULL, 27, 2, '2021-11-25'),
9  (NULL, 30, 2, '2021-10-24'),
10 (NULL, 33, 2, '2021-12-24'),
11 (NULL, 31, 2, '2021-9-18'),
12 (NULL, 32, 2, '2021-8-8'),
13 (NULL, 29, 2, '2021-9-7'),
14 (NULL, 27, 2, '2021-5-24'),
15 (NULL, 25, 2, '2021-9-24'),
16 (NULL, 24, 2, '2021-7-24'),
17 (NULL, 29, 2, '2021-2-24'),
18 (NULL, 28, 2, '2021-11-30');

```

Figure 3.12 – Historical books order table

Looking at figure 3.12, it is quite clear that this table takes in two foreign keys, one for the book, and another for the customer who is placing the order on that book. These two foreign keys link with their respective parent tables and, hence, identifies the specific customer and the book.

The rest of the tables for booking are made in a similar fashion and can be seen in the figures below:

```

1 • Ⓜ CREATE TABLE booked_courses(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     courses_id INT UNSIGNED NOT NULL,
4     customer_id INT UNSIGNED NOT NULL,
5     FOREIGN KEY (courses_id) REFERENCES courses(id),
6     FOREIGN KEY (customer_id) REFERENCES customer(id)
7 )

```

```

1   INSERT INTO booked_course(id, courses_id, customer_id) VALUES
2   (NULL, 2, 1),
3   (NULL, 2, 2),
4   (NULL, 2, 3),
5   (NULL, 2, 5),
6   (NULL, 6, 20),
7   (NULL, 6, 15),
8   (NULL, 3, 4),
9   (NULL, 4, 7),
10  (NULL, 4, 9),
11  (NULL, 4, 11),
12  (NULL, 4, 8),
13  (NULL, 4, 10),
14  (NULL, 5, 13),
15  (NULL, 5, 17),
16  (NULL, 5, 16),
17  (NULL, 1, 14),
18  (NULL, 1, 19),
19  (NULL, 1, 18),
20  (NULL, 1, 6);

```

Figure 3.13 – Table for booked courses

```

1 • Ⓜ CREATE TABLE booked_lecture(
2     id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3     lecture_id INT UNSIGNED NOT NULL,
4     customer_id INT UNSIGNED NOT NULL,
5     FOREIGN KEY (customer_id) REFERENCES customer(id),
6     FOREIGN KEY (lecture_id) REFERENCES lecture(id)
7 )

```

```

1  INSERT INTO booked_lecture(id, customer_id, lecture_id) VALUES
2  (NULL, 1, 2),
3  (NULL, 2, 2),
4  (NULL, 3, 2),
5  (NULL, 5, 1),
6  (NULL, 8, 3),
7  (NULL, 6, 5),
8  (NULL, 7, 4),
9  (NULL, 4, 7),
10 (NULL, 14, 8),
11 (NULL, 10, 2),
12 (NULL, 13, 7),
13 (NULL, 16, 7),
14 (NULL, 19, 5),
15 (NULL, 11, 4),
16 (NULL, 15, 3),
17 (NULL, 9, 6),
18 (NULL, 20, 6);

```

Figure 3.14 – Table for booked lectures

```

1 • ⓧ CREATE TABLE booked_tour(
2   id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
3   customer_id INT UNSIGNED NOT NULL,
4   tour_id INT UNSIGNED NOT NULL,
5   FOREIGN KEY (customer_id) REFERENCES customer(id),
6   FOREIGN KEY (tour_id) REFERENCES tour(id)
7 )

1  INSERT INTO booked_tour(id, customer_id, tour_id) VALUES
2  (NULL, 1, 14),
3  (NULL, 2, 14),
4  (NULL, 3, 14),
5  (NULL, 4, 14),
6  (NULL, 5, 15),
7  (NULL, 6, 16),
8  (NULL, 7, 17),
9  (NULL, 8, 18),
10 (NULL, 9, 19),
11 (NULL, 10, 19),
12 (NULL, 11, 19),
13 (NULL, 12, 19),
14 (NULL, 13, 20),
15 (NULL, 14, 21),
16 (NULL, 15, 22),
17 (NULL, 16, 26);
18

```

Figure 3.15 – Booked tour data

4. Querying

This section starts off with simply changing the gender for an instructor with a unique ID, in its respective table:

```

1   UPDATE instructor
2   SET gender = 'F'
3   WHERE id = 4;

```

Figure 4.1

The following query demonstrates an inner join on lecture, the returned table links lecture with its lecturer and their specialty. Subsequently, the query after this shows the same returned table ordered by date:

```

1 • SELECT
2   lecturer.full_name Lecturer, lecturer.speciality Speciality, lecture.topic Topic, lecture.address Address, lecture.city City, lecture.venue Venue,
3   lecture.lecture_date Lecture_Date
4   FROM lecture
5   INNER JOIN lecturer
6   ON lecture.lecturer_id=lecturer.id

```

Result Grid						
Lecturer	Specialty	Topic	Address	City	Venue	Lecture_Date
Nial Ferguson	International History	Renaissance	Kensington	London	Royal Albert Hall	2022-02-22
Timothy D. Snyder	Central and Eastern Europe History	Seljuk Turks	3000 Broadway	New York	Havemeyer Hall	2021-12-22
Timothy D. Snyder	Central and Eastern Europe History	Genghis Khan	Orestads Blvd	Copenhagen	DR Koncerthuset	2022-01-15
Simon Schama	Art, Dutch, and Jewish History	Holocaust	Concertgebouwplein 10	Amsterdam	The Concertgebouw	2022-09-03
Eamon Duffy	Religious History of Britain	The Dark Ages	2 W Tollcross	Edinburgh	Central Hall	2021-12-22
Anthony Beevor	Military History	World War	Ehrenhof 1	Dusseldorf	Tonhalle Dusseldorf	2021-12-18
Robert C. Allen	Economic History	Communism	221 Av. Jean Jaures	Paris	Paris Philharmonic	2021-10-22
Romila Thapar	Ancient India	Pre-Mughal India	Haus der Industrie	Vienna	Wiener Royal Orchester	2022-09-22

Figure 4.2

```

1 • SELECT
2   lecturer.full_name Lecturer, lecturer.speciality Speciality, lecture.topic Topic, lecture.address Address, lecture.city City, lecture.venue Venue,
3   lecture.lecture_date Lecture_Date
4   FROM lecture
5   INNER JOIN lecturer
6   ON lecture.lecturer_id=lecturer.id
7   ORDER BY Lecture_Date

```

Result Grid						
Lecturer	Specialty	Topic	Address	City	Venue	Lecture_Date
Robert C. Allen	Economic History	Communism	221 Av. Jean Jaures	Paris	Paris Philharmonic	2021-10-22
Anthony Beevor	Military History	World War	Ehrenhof 1	Dusseldorf	Tonhalle Dusseldorf	2021-12-18
Timothy D. Snyder	Central and Eastern Europe History	Seljuk Turks	3000 Broadway	New York	Havemeyer Hall	2021-12-22
Eamon Duffy	Religious History of Britain	The Dark Ages	2 W Tollcross	Edinburgh	Central Hall	2021-12-22
Timothy D. Snyder	Central and Eastern Europe History	Genghis Khan	Orestads Blvd	Copenhagen	DR Koncerthuset	2022-01-15
Nial Ferguson	International History	Renaissance	Kensington	London	Royal Albert Hall	2022-02-22
Simon Schama	Art, Dutch, and Jewish History	Holocaust	Concertgebouwplein 10	Amsterdam	The Concertgebouw	2022-09-03
Romila Thapar	Ancient India	Pre-Mughal India	Haus der Industrie	Vienna	Wiener Royal Orchester	2022-09-22

Figure 4.3

The queries below return the guide to the tour table and customer to the magazine subscription table from their designated foreign keys with an inner join and uses an alias for guide:

```

1  SELECT guide.full_name AS 'Guide', guide.email, tour.departure, tour.destination, tour.departure_date
2  FROM tour
3  INNER JOIN guide
4  ON tour.guide_id=guide.id;

```

The screenshot shows a MySQL Workbench result grid with the following columns: Guide, email, departure, destination, and departure_date. The data consists of 15 rows, each representing a tour entry with its corresponding guide information.

	Guide	email	departure	destination	departure_date
▶	Christopher Wallace	christopher@bp.com	Victoria Station London	Taj Mahal	2021-12-28
	Thomas Shelby	shelby@peaky.com	Clapham Junction London	Pyramids of Giza	2022-01-28
	Micheal Scott	michael@mifflin.com	Copenhagen Airport	Machu Picchul	2022-03-14
	David Wallace	david@dunder.com	Stopera Amsterdam	Bagan	2021-12-31
	Lena Grey	lena@gmail.com	Brandenburg Gate Berlin	Angkor Wat	2021-11-24
	Stephanie Marvel	stephanie@comicon.com	Bosphorus Istanbul	Great Wall of China	2022-02-28
	Bruce Wayne	wayne@billionaire.com	Zytglogge Bern	Roman Colosseum	2022-02-03
	Lauren Potter	laurenpotter@hogwarts.com	Field Museum Chicago	Acropolis	2022-05-28
	Richard Nixon	nixon@usa.com	Santa Monica Pier Los Angeles	Stonehenge	2021-11-04
	Pheobe Buffet	pheobe@centralperk.com	Moscow Metro	Borobudur	2022-02-14
	Brittany Amber	brittany@derby.com	Calgary Tower	Mesa Verde	2021-12-01
	Elaine Benes	elaine@amazon.com	University of Toronto	Terracotta Army	2021-11-17
	Lauren James	tiffany@netflix.com	Madrid Train Station	Petra	2021-10-20

Figure 4.4

```

1 •  SELECT magazine_subscription.tittle, customer.full_name, magazine_subscription.start_date
2  FROM magazine_subscription
3  INNER JOIN customer
4  ON magazine_subscription.customer_id = customer.id;

```

The screenshot shows a MySQL Workbench result grid with the following columns: title, full_name, and start_date. The data consists of 10 rows, each representing a magazine subscription entry with its corresponding customer information.

	title	full_name	start_date
▶	Ancient Gallery	Christopher Jones	2021-11-20
	Sculptures	Melissa Moore	2021-12-02
	Ancient Ornaments	Melissa Brown	2021-09-05
	Mughals Kitchen	Jennifer Thomas	2021-08-09
	Greek Architecture	Ashley Johnson	2021-09-15
	Evolution	Justin Smith	2021-12-18
	The Silk Road	Justin Smith	2021-11-24
	Tsars	James Martinez	2022-01-20
	Evolution of Fashion	Lauren Anderson	2021-05-20

Figure 4.5

Furthermore, the column name of 'tittle' in magazine subscription table is changed to '*Magazine Tittle*' with the query below:

```

1 ALTER TABLE magazine_subscription CHANGE tittle Magazine_Tittle VARCHAR(30);
2 • SELECT * FROM magazine_subscription;

```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The query `SELECT * FROM magazine_subscription;` has been run. The result grid displays the following data:

	id	customer_id	Magazine_Title	start_date
▶	1	1	Ancient Gallery	2021-11-20
	2	3	Sculptures	2021-12-02
	3	4	Ancient Ornaments	2021-09-05
	4	5	Mughals Kitchen	2021-08-09
	5	10	Greek Architecture	2021-09-15
	6	14	Evolution	2021-12-18
	7	14	The Silk Road	2021-11-24
	8	19	Tsars	2022-01-20
	9	8	Evolution of Fashion	2021-05-20
✳	NULL	NULL	NULL	NULL

Figure 4.6

The following two queries joins three tables, both retuning customers that are going on a tour in one and those signed up for a course in another, also using aliases for all the columns:

```

1 • SELECT
2     customer.full_name Adventurer, customer.email Email, tour.departure Departure, tour.destination Destination,
3     tour.departure_date Departure_Date
4     FROM booked_tour
5     INNER JOIN customer
6     ON booked_tour.customer_id=customer.id
7     JOIN tour
8     on booked_tour.tour_id=tour.id;

```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The query has been run, and the result grid displays the following data:

	Adventurer	Email	Departure	Destination	Departure_Date
▶	Christopher Jones	christopherjones@bp.com	Victoria Station London	Taj Mahal	2021-12-28
	Matthew Martinez	matthewmartinez@ge.com	Victoria Station London	Taj Mahal	2021-12-28
	Melissa Moore	melissamoore@aramark.com	Victoria Station London	Taj Mahal	2021-12-28
	Melissa Brown	melissabrown@verizon.com	Victoria Station London	Taj Mahal	2021-12-28
	Jennifer Thomas	jenniferthomas@aramark.com	Clapham Junction London	Pyramids of Giza	2022-01-28
	Stephanie Martinez	stephaniemartinez@albertsons.com	Copenhagen Airport	Machu Picchul	2022-03-14
	Daniel Williams	danielwilliams@tjx.com	Stopera Amsterdam	Bagan	2021-12-31
	Lauren Anderson	laurenanderson@pepsi.com	Brandenburg Gate Berlin	Angkor Wat	2021-11-24
	Michael Jackson	michaeljackson@disney.com	Bosphorus Istanbul	Great Wall of China	2022-02-28
	Ashley Johnson	ashleyjohnson@boeing.com	Bosphorus Istanbul	Great Wall of China	2022-02-28
	Brittany Thomas	brittanythomas@walmart.com	Bosphorus Istanbul	Great Wall of China	2022-02-28
	Tiffany Smith	tiffanysmith@ups.com	Bosphorus Istanbul	Great Wall of China	2022-02-28
	Lauren Wilson	laurenwilson@target.com	Zytglogge Bern	Roman Colosseum	2022-02-03
	Justin Smith	justinsmith@boeing.com	Field Museum Chicago	Acropolis	2022-05-28
	Jessica Garcia	jessicagarcia@toyota.com	Santa Monica Pier Los Angeles	Stonehenge	2021-11-04
	Matthew Jackson	matthewjackson@bp.com	Madrid Train Station	Petra	2021-10-20

Figure 4.7

```

1 •   SELECT
2     booked_course.id Serial_Number, customer.full_name Student, customer.id Student_ID ,customer.email Email, courses.tittle Course, courses.start_date Begins,
3     courses.city City
4   FROM booked_course
5   INNER JOIN customer
6   ON booked_course.customer_id=customer.id
7   JOIN courses
8   on booked_course.courses_id=courses.id;

```

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is the one shown above, selecting data from booked_course, customer, and courses tables. The result grid displays 19 rows of data with columns: Serial_Number, Student, Student_ID, Email, Course, Begins, and City. The data includes various student names, their emails, course titles like 'Capitalism' and 'History of Monetary Trade', start dates ranging from 2021-10-29 to 2021-12-29, and cities like Chicago, Washington D.C., and New York.

	Serial_Number	Student	Student_ID	Email	Course	Begins	City
▶	1	Christopher Jones	1	christopherjones@bp.com	Capitalism	2021-11-24	Chicago
	2	Matthew Martinez	2	matthewmartinez@ge.com	Capitalism	2021-11-24	Chicago
	3	Melissa Moore	3	mellissamoore@aramark.com	Capitalism	2021-11-24	Chicago
	4	Jennifer Thomas	5	jenniferthomas@aramark.com	Capitalism	2021-11-24	Chicago
	5	Christopher Robinson	20	christopherrobinson@ibm.com	History of Monetary Trade	2021-02-15	Washington D.C.
	6	Jessica Garcia	15	jessicagarcia@toyota.com	History of Monetary Trade	2021-02-15	Washington D.C.
	7	Melissa Brown	4	melisabrown@verizon.com	Ancient Civilization	2021-10-29	Scranton
	8	Daniel Williams	7	danielwilliams@tx.com	Introduction to Anthropology	2021-12-29	Sacramento
	9	Michael Jackson	9	michaeljackson@disney.com	Introduction to Anthropology	2021-12-29	Sacramento
	10	Brittany Thomas	11	brittanythomas@walmart.com	Introduction to Anthropology	2021-12-29	Sacramento
	11	Lauren Anderson	8	laurenanderson@pepsi.com	Introduction to Anthropology	2021-12-29	Sacramento
	12	Ashley Johnson	10	ashleyjohnson@boeing.com	Introduction to Anthropology	2021-12-29	Sacramento
	13	Lauren Wilson	13	laurenwilson@target.com	Archaeology Fundamentals	2022-01-29	Louisville
	14	Stephanie Thomas	17	stephanie.thomas@apple.com	Archaeology Fundamentals	2022-01-29	Louisville
	15	Matthew Jackson	16	matthew.jackson@bp.com	Archaeology Fundamentals	2022-01-29	Louisville
	16	Justin Smith	14	justin.smith@boeing.com	Ancient Civilization	2021-10-29	New York
	17	James Martinez	19	jamesmartinez@kroger.com	Ancient Civilization	2021-10-29	New York
	18	Jessica Jackson	18	jessicajackson@aramark.com	Ancient Civilization	2021-10-29	New York
	19	Stephanie Martinez	6	stephaniemartinez@albertsons.com	Ancient Civilization	2021-10-29	New York

Figure 4.8

The following two queries counts the number of customers signed up to a magazine and similarly, the number of magazine subscriptions a specific customer has, grouped by the tittle and customer respectively:

```

1 •   SELECT COUNT(customer_id), tittle
2     FROM magazine_subscription
3   GROUP BY tittle;

```

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is the one shown above, which groups magazine titles by the number of subscribers. The result grid displays 8 rows of data with columns: COUNT(customer_id) and tittle. The data shows titles like 'Ancient Gallery', 'Sculptures', 'Ancient Ornaments', etc., each having a count of 1.

	COUNT(customer_id)	tittle
▶	1	Ancient Gallery
	1	Sculptures
	1	Ancient Ornaments
	1	Mughals Kitchen
	1	Greek Architecture
	1	Evolution
	1	The Silk Road
	1	Tsars
	1	Evolution of Fashion

Figure 4.9

```

1   SELECT COUNT(tittle) Number_of_Magazines, customer.full_name Customer_Name, magazine_subscription.start_date Start_Date
2   FROM magazine_subscription
3   INNER JOIN customer
4   ON magazine_subscription.customer_id = customer.id
5   GROUP BY Customer_Name

```

The screenshot shows a database query results grid with the following columns: Number_of_Magazines, Customer_Name, and Start_Date. The data is as follows:

	Number_of_Magazines	Customer_Name	Start_Date
▶	1	Christopher Jones	2021-11-20
	1	Melissa Moore	2021-12-02
	1	Melissa Brown	2021-09-05
	1	Jennifer Thomas	2021-08-09
	1	Ashley Johnson	2021-09-15
	2	Justin Smith	2021-12-18
	1	James Martinez	2022-01-20
	1	Lauren Anderson	2021-05-20

Figure 4.10

The query below combines full name from customer and tittle from magazine subscription:

```

1 •    select full_name from customer
2     UNION
3     select tittle from magazine_subscription
4

```

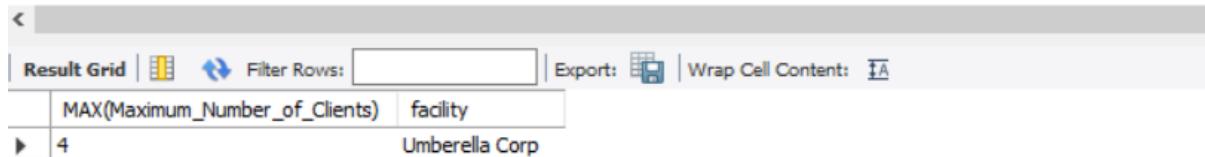
The screenshot shows a database query results grid with the following column: full_name. The data is as follows:

full_name
Melissa Brown
Jennifer Thomas
Stephanie Martinez
Daniel Williams
Lauren Anderson
Michael Jackson
Ashley Johnson
Brittany Thomas
Tiffany Smith
Lauren Wilson
Justin Smith
Jessica Garcia
Matthew Jackson
Stephanie Thomas
Jessica Jackson
James Martinez
Christopher Robinson
Ancient Gallery
Sculptures
Ancient Ornaments
Mughals Kitchen
Greek Architecture
Evolution
The Silk Road

Figure 4.11

Finally, the query below finds out the maximum customers doing their genealogy test at a specific facility. A sub-query is also inserted that counts the customers:

```
1 •  SELECT MAX(Maximum_Number_of_Clients), facility
2   FROM (SELECT facility, customer_id, COUNT(customer_id) Maximum_Number_of_Clients
3   FROM genology_test WHERE facility = "Umberella Corp") Genology_Test_Facility;
4
```



The screenshot shows a MySQL Workbench result grid. The title bar includes icons for back, forward, and search, followed by 'Result Grid' and other export options. The main area displays a single row of data in a table format:

MAX(Maximum_Number_of_Clients)	facility
4	Umberella Corp

Figure 4.12

5. Optimization

Several methods can be used to optimize a database, the one used for this case study is indexing and can be seen in the figures below:

```
1   CREATE INDEX lecture_index  
2     ON lecture(topic, venue, city);
```



```
✓ 51 11:00:15 CREATE INDEX lecture_index ON lecture(topic, venue, city)
```



```
1 • CREATE INDEX magazine_index  
2   ON magazine_subscription(Magazine_Tittle, start_date);  
3   |
```



```
✓ 58 11:05:26 CREATE INDEX magazine_index ON magazine_subscription(Magazine_Tittle, start_date)
```



```
1 • CREATE INDEX destination_index  
2   ON tour(destination);  
3   |
```

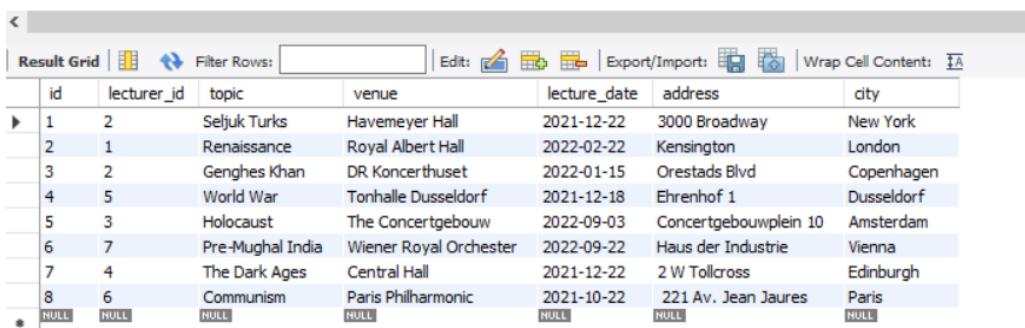


```
✓ 56 11:03:19 CREATE INDEX destination_index ON tour(destination)
```

Figure 5.1 – Creating several indexes

These indexes can be invoked as follows:

```
1 • SELECT *  
2   FROM lecture USE INDEX(lecture_index) |
```



	ID	Lecturer ID	Topic	Venue	Lecture Date	Address	City
▶	1	2	Seljuk Turks	Havemeyer Hall	2021-12-22	3000 Broadway	New York
	2	1	Renaissance	Royal Albert Hall	2022-02-22	Kensington	London
	3	2	Genghis Khan	DR Koncerthuset	2022-01-15	Orestads Blvd	Copenhagen
	4	5	World War	Tonhalle Dusseldorf	2021-12-18	Ehrenhof 1	Dusseldorf
	5	3	Holocaust	The Concertgebouw	2022-09-03	Concertgebouwplein 10	Amsterdam
	6	7	Pre-Mughal India	Wiener Royal Orchester	2022-09-22	Haus der Industrie	Vienna
	7	4	The Dark Ages	Central Hall	2021-12-22	2 W Tollcross	Edinburgh
*	8	6	Communism	Paris Philharmonic	2021-10-22	221 Av. Jean Jaures	Paris
*		NULL	NULL	NULL	NULL	NULL	NULL

```

1 •   SELECT *
2     FROM tour USE INDEX(destination) |

```

	id	guide_id	departure	destination	departure_date
▶	14	1	Victoria Station London	Taj Mahal	2021-12-28
	15	2	Clapham Junction London	Pyramids of Giza	2022-01-28
	16	3	Copenhagen Airport	Machu Picchul	2022-03-14
	17	4	Stopera Amsterdam	Bagan	2021-12-31
	18	5	Brandenburg Gate Berlin	Angkor Wat	2021-11-24
	19	6	Bosphorus Istanbul	Great Wall of China	2022-02-28
	20	7	Zytglogge Bern	Roman Colosseum	2022-02-03
	21	8	Field Museum Chicago	Acropolis	2022-05-28
	22	9	Santa Monica Pier Los Angeles	Stonehenge	2021-11-04
	23	10	Moscow Metro	Borobudur	2022-02-14
	24	11	Calgary Tower	Mesa Verde	2021-12-01
	25	12	University of Toronto	Terracotta Army	2021-11-17
	26	13	Madrid Train Station	Petra	2021-10-20
*	NULL	NULL	NULL	NULL	NULL

Figure 5.2 – Invoking indexes

Conversely, these indexes can be suppressed by dropping them:

```

1 •   ALTER TABLE magazine_subscription
2     DROP INDEX magazine_index; |

```

8 08:20:49 ALTER TABLE magazine_subscription DROP INDEX magazine_index

```

1 •   ALTER TABLE tour
2     DROP INDEX destination_index; |

```

6 08:18:47 ALTER TABLE tour DROP INDEX destination_index

Figure 5.3 – Dropping the indexes that were made earlier

6. Securing Database

To demonstrate securing the database, three new users are created with different privileges:

```
1 • CREATE USER "david" IDENTIFIED BY "dundermifflin";
2 • CREATE USER "mr.robot" IDENTIFIED BY "cyberspace";
3 • CREATE USER "jerry" IDENTIFIED BY "seinfeld";
```

The screenshot shows the MySQL Workbench interface with the 'Output' tab selected. It displays the history of SQL statements executed. The first three statements create users 'david', 'mr.robot', and 'jerry' respectively, each with a green checkmark. The next two statements grant all privileges to 'david' and flush privileges, also with green checkmarks. The final statement grants insert privilege to 'jerry' with a green checkmark.

#	Time	Action
1	08:56:12	CREATE USER "david" IDENTIFIED BY "dundermifflin"
2	08:56:12	CREATE USER "mr.robot" IDENTIFIED BY "cyberspace"
3	08:56:12	CREATE USER "jerry" IDENTIFIED BY "seinfeld"
44	09:42:29	GRANT ALL PRIVILEGES ON * . * TO 'david'@'localhost';
45	09:42:29	FLUSH PRIVILEGES;
1	11 09:07:30	GRANT INSERT ON mysql.user TO 'jerry'@'%';

Figure 6.1 – Creating new users and allocating them different privileges

Moreover, several views are made that overviews the booking for lectures, courses and ordered historical books with different aliases for several columns:

```
1 CREATE VIEW lecture_overview AS
2 SELECT customer.full_name Attendee, customer.email Attendee_Email, lecture.topic Lecture,
3 lecturer.full_name Lecturer, lecturer.speciality Lecturer_Speciality, lecture.venue Venue, lecture.city City
4 FROM booked_lecture
5 JOIN customer
6 ON booked_lecture.customer_id = customer.id
7 JOIN lecture
8 ON booked_lecture.lecture_id = lecture.id
9 JOIN lecturer
10 ON lecture.lecturer_id = lecturer.id
```

```
1   SELECT * FROM lecture_overview
```

Attendee	Attendee_Email	Lecture	Lecturer	Lecturer_Specialty	Venue	City
Christopher Jones	christopherjones@bp.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Matthew Martinez	matthewmartinez@ge.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Melissa Moore	melissamoore@aramark.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Jennifer Thomas	jenniferthomas@aramark.com	Seljuk Turks	Timothy D. Snyder	Central and Eastern Europe History	Havemeyer Hall	New York
Lauren Anderson	laurenanderson@pepsi.com	Genghis Khan	Timothy D. Snyder	Central and Eastern Europe History	DR Koncerthuset	Copenhagen
Stephanie Martinez	stephaniemartinez@albertsons.com	Holocaust	Simon Schama	Art, Dutch, and Jewish History	The Concertgebouw	Amsterdam
Daniel Williams	danielwilliams@tx.com	World War	Anthony Beevor	Military History	Tonhalle Dusseldorf	Dusseldorf
Melissa Brown	melissabrown@verizon.com	The Dark Ages	Eamon Duffy	Religious History of Britain	Central Hall	Edinburgh
Justin Smith	justinsmith@boeing.com	Communism	Robert C. Allen	Economic History	Paris Philharmonic	Paris
Ashley Johnson	ashleyjohnson@boeing.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Lauren Wilson	laurenwilson@target.com	The Dark Ages	Eamon Duffy	Religious History of Britain	Central Hall	Edinburgh
Matthew Jackson	matthewjackson@bp.com	The Dark Ages	Eamon Duffy	Religious History of Britain	Central Hall	Edinburgh
James Martinez	jamesmartinez@kroger.com	Holocaust	Simon Schama	Art, Dutch, and Jewish History	The Concertgebouw	Amsterdam
Brittany Thomas	brittanythomas@walmart.com	World War	Anthony Beevor	Military History	Tonhalle Dusseldorf	Dusseldorf
Jessica Garcia	jessicagarcia@toyota.com	Genghis Khan	Timothy D. Snyder	Central and Eastern Europe History	DR Koncerthuset	Copenhagen
Michael Jackson	michaeljackson@disney.com	Pre-Mughal India	Romila Thapar	Ancient India	Wiener Royal Orchester	Vienna
Christopher Robinson	christopherrobinson@ibm.com	Pre-Mughal India	Romila Thapar	Ancient India	Wiener Royal Orchester	Vienna

Figure 6.2 – Lectures overview

```
1   CREATE VIEW courses_overview AS
2   SELECT customer.full_name Student, customer.email Student_Email, customer.date_of_birth Student_DoB,
3   courses.title Course_Tittle, instructor.full_name Instructor, courses.start_date Course_Starts, courses.city City
4   FROM booked_course
5   JOIN customer
6   ON booked_course.customer_id = customer.id
7   JOIN courses
8   ON booked_course.courses_id = courses.id
9   JOIN instructor
10  ON courses.instructor_id = instructor.id
```

```
1 •  SELECT * FROM courses_overview
```

Student	Student_Email	Student_DoB	Course_Tittle	Instructor	Course_Starts	City
Christopher Jones	christopherjones@bp.com	1958-09-11	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Matthew Martinez	matthewmartinez@ge.com	1961-09-04	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Melissa Moore	melissamoore@aramark.com	1967-08-27	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Jennifer Thomas	jenniferthomas@aramark.com	1998-03-14	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Christopher Robinson	christopherrobinson@ibm.com	1972-06-25	History of Monetary Trade	Mark Zuckerberg	2021-02-15	Washington D.C.
Jessica Garcia	jessicagarcia@toyota.com	1996-08-05	History of Monetary Trade	Mark Zuckerberg	2021-02-15	Washington D.C.
Melissa Brown	melissabrown@verizon.com	1948-06-14	Ancient Civilization	Samuel Jacson	2021-10-29	Scranton
Daniel Williams	danielwilliams@tx.com	1985-07-20	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento
Michael Jackson	michaeljackson@disney.com	1951-03-03	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento
Brittany Thomas	brittanythomas@walmart.com	1986-10-22	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento
Lauren Anderson	laurenanderson@pepsi.com	1973-09-09	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento
Ashley Johnson	ashleyjohnson@boeing.com	1977-05-10	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento
Lauren Wilson	laurenwilson@target.com	1965-12-26	Archaeology Fundamentals	Warren Buffet	2022-01-29	Louisville
Stephanie Thomas	stephaniethomas@apple.com	1988-08-26	Archaeology Fundamentals	Warren Buffet	2022-01-29	Louisville
Matthew Jackson	matthewjackson@bp.com	1996-02-26	Archaeology Fundamentals	Warren Buffet	2022-01-29	Louisville
Justin Smith	justinsmith@boeing.com	1956-03-16	Ancient Civilization	Miranda Kerr	2021-10-29	New York
James Martinez	jamesmartinez@kroger.com	1987-12-22	Ancient Civilization	Miranda Kerr	2021-10-29	New York
Jessica Jackson	jessicajackson@aramark.com	1991-07-22	Ancient Civilization	Miranda Kerr	2021-10-29	New York
Stephanie Martinez	stephaniemartinez@albertso...	1998-01-24	Ancient Civilization	Miranda Kerr	2021-10-29	New York

Figure 6.3 – Courses overview

```

1 CREATE VIEW historical_books_orders AS
2   SELECT customer.full_name Customer, historical_books.title Book, historical_books.author Author,
3   customer.email Customer_Email, customer.address Customer_Address, books_order.purchase_date Purchase_Date
4   FROM books_order
5   JOIN customer
6   ON books_order.customer_id = customer.id
7   JOIN historical_books
8   ON books_order.historical_books_id = historical_books.id

```

1 • select * from historical_books_orders

The screenshot shows a database query results grid titled "Result Grid". The grid has columns: Customer, Book, Author, Customer_Email, Customer_Address, and Purchase_Date. The data consists of 19 rows, each representing a book purchase. The "Customer" column shows two distinct names: Christopher Jones and Matthew Martinez. The "Book" column lists various historical books, including "Guns, Germs, and Steel", "1491: New Revelations of the Americas Before Columbus", "Genghis Khan and the Making of the Modern World", "Sapiens: A Brief History of Humankind", "The Silk Roads: A New History of the World", "A People's History of the United States", "The Crusades: The Authoritative History of the War for the Holy Land", "The History of the Decline and Fall of the Roman Empire", and "The Rise and Fall of the Third Reich". The "Author" column lists the authors corresponding to the books. The "Customer_Email" column shows email addresses such as christopherjones@bp.com and matthewmartinez@ge.com. The "Customer_Address" column shows addresses like 347 Cedar St and 602 Main Place. The "Purchase_Date" column shows dates ranging from 2021-07-24 to 2021-11-14.

Result Grid						
	Customer	Book	Author	Customer_Email	Customer_Address	Purchase_Date
▶	Christopher Jones	Guns, Germs, and Steel	Jared Diamond	christopherjones@bp.com	347 Cedar St	2021-10-29
	Matthew Martinez	Guns, Germs, and Steel	Jared Diamond	matthewmartinez@ge.com	602 Main Place	2021-10-24
	Matthew Martinez	1491: New Revelations of the Americas Before ...	Charles C. Mann	matthewmartinez@ge.com	602 Main Place	2021-11-14
	Matthew Martinez	1491: New Revelations of the Americas Before ...	Charles C. Mann	matthewmartinez@ge.com	602 Main Place	2021-07-24
	Matthew Martinez	Genghis Khan and the Making of the Modern W...	Jack Weatherford	matthewmartinez@ge.com	602 Main Place	2021-12-28
	Matthew Martinez	Genghis Khan and the Making of the Modern W...	Jack Weatherford	matthewmartinez@ge.com	602 Main Place	2021-09-24
	Matthew Martinez	Genghis Khan and the Making of the Modern W...	Jack Weatherford	matthewmartinez@ge.com	602 Main Place	2021-09-24
	Matthew Martinez	Sapiens: A Brief History of Humankind	Yuval Noah Harari	matthewmartinez@ge.com	602 Main Place	2021-08-24
	Matthew Martinez	The Silk Roads: A New History of the World	Peter Frankopan	matthewmartinez@ge.com	602 Main Place	2021-11-25
	Matthew Martinez	The Silk Roads: A New History of the World	Peter Frankopan	matthewmartinez@ge.com	602 Main Place	2021-05-24
	Matthew Martinez	A People's History of the United States	Howard Zinn	matthewmartinez@ge.com	602 Main Place	2021-11-30
	Matthew Martinez	The Crusades: The Authoritative History of the ...	Thomas Asbridge	matthewmartinez@ge.com	602 Main Place	2021-09-07
	Matthew Martinez	The Crusades: The Authoritative History of the ...	Thomas Asbridge	matthewmartinez@ge.com	602 Main Place	2021-02-24
	Matthew Martinez	The History of the Decline and Fall of the Roman Empire	Edward Gibbon	matthewmartinez@ge.com	602 Main Place	2021-10-24
	Matthew Martinez	The Rise and Fall of the Third Reich	William L. Shirer	matthewmartinez@ge.com	602 Main Place	2021-09-18

Figure 6.3 – Historical books order overview

7. Web Interface

To create the web interface all the data is exported from MySQL Workbench to XAMPP as follows:

The figure consists of three vertically stacked screenshots illustrating the data import process:

- Top Screenshot:** MySQL Workbench Data Export window. It shows a progress bar indicating "Export Completed" with "18 of 18 exported". The log area displays command-line output for dumping and exporting data to a file named "stepping_into_history.sql".
- Middle Screenshot:** XAMPP Control Panel Database tab. A green status message at the top states "Import has been successfully finished, 219 queries executed (stepping_into_history.sql)".
- Bottom Screenshot:** phpMyAdmin interface showing the "import_database" database structure. The "Tables" section lists 18 tables with their respective details like Type, Collation, and Size.

Table	Action	Rows	Type	Collation	Size	Overhead
booked_course	Browse Structure Search Insert Empty Drop	19	InnoDB	utf8_general_ci	48.0 Kib	-
booked_lecture	Browse Structure Search Insert Empty Drop	17	InnoDB	utf8_general_ci	48.0 Kib	-
booked_tour	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8_general_ci	48.0 Kib	-
books_order	Browse Structure Search Insert Empty Drop	17	InnoDB	utf8_general_ci	48.0 Kib	-
courses	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	32.0 Kib	-
courses_overview	Browse Structure Search Insert Edit Drop	~0	InnoDB	utf8_general_ci	-	-
customer	Browse Structure Search Insert Empty Drop	20	InnoDB	utf8_general_ci	16.0 Kib	-
genology_test	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	32.0 Kib	-
guide	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8_general_ci	16.0 Kib	-
historical_books	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8_general_ci	16.0 Kib	-
historical_books_orders	Browse Structure Search Insert Edit Drop	~0	InnoDB	utf8_general_ci	-	-
instructor	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	16.0 Kib	-
lecture	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	48.0 Kib	-
lecturer	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	16.0 Kib	-
lecture_overview	Browse Structure Search Insert Edit Drop	~0	InnoDB	utf8_general_ci	-	-
literature	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	32.0 Kib	-
magazine_subscription	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8_general_ci	32.0 Kib	-
tour	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8_general_ci	32.0 Kib	-
Sum		~169	InnoDB	utf8mb4_general_ci	488.0 Kib	0 B

Figure 7.1 – Importing data to XAMPP and the database

The following PHP code is used to develop the web interface that shows several tables from the database and a new customer (referred to as member on the front-end) can be added and deleted:

```

1  <?php
2  $serverName = "localhost";
3  $serverUsername = "root";
4  $serverPassword = "";
5  $dbName = "import_database";
6
7  //Create a connection
8  $conn = mysqli_connect($serverName, $serverUsername, $serverPassword, $dbName);
9
10 //Die if connection was not successful
11 if(!$conn){
12     die("Connection failed:".mysqli_connect_error());
13 }
14

```

Figure 7.2 – Connecting to the database

```

1  <?php
2      include_once('connection.php');
3      $query = "SELECT * FROM customer";
4      $result = mysqli_query($conn, $query); ?>
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <title>Staff</title>
9      <meta charset="utf-8">
10     <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
11    <style>
12        table{
13            width:1000px; line-height:55px; margin:auto; margin-top:60px; text-align: center; border: 1px solid black; border-collapse: collapse;
14        }
15        h1{
16            text-align: center; padding-top: 20px; font: helvetica;
17        }
18        tr {background-color: #ADD8E6;}
19        td{
20            border: 1px solid #F8F8FF;
21        }
22    </style>
23  </head>
24  <body style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner2.jpg);">
25
26      <?php include 'navbar.php'?>
27      <h1 style="color:white">Members:</h1>
28      <table align = "center">
29          <tr style="font-size:20px; background-color: #0072A0; color: white;">
30              <th style="text-align:center">ID</th>
31              <th style="text-align:center">Name</th>
32              <th style="text-align:center">Address</th>
33              <th style="text-align:center">Email</th>
34              <th style="text-align:center">Zip</th>
35              <th style="text-align:center">Gender</th>
36              <th style="text-align:center">Date of Birth</th>
37              <th style="text-align:center">Delete</th>
38          </tr>
39          <?php
40              while($rows = mysqli_fetch_array($result)){
41                  <?>
42                      <tr style="background-color:#ADD8E6; border: 1px solid azure;">
43                          <td><?php echo $rows['id']?></td>
44                          <td><?php echo $rows['full_name']?></td>
45                          <td><?php echo $rows['address']?></td>
46                          <td><?php echo $rows['email']?></td>
47                          <td><?php echo $rows['zip']?></td>
48                          <td><?php echo $rows['gender']?></td>
49                          <td><?php echo $rows['date_of_birth']?></td>
50                          <td><a href="delete.php?del=<?php echo $rows['id']?>">Delete</a></td>
51          </tr>
52          <?php
53          }?
54      </table><br>
55  </body>

```

Figure 7.3 – Displaying the customers' table

```

1  <?php
2
3  ▼ function emptyInputSignup($name, $address, $age, $zip, $gender, $date_of_birth){
4      $result;
5      if(empty($name) || empty($address) || empty($age) || empty($zip) || empty($gender) || empty($date_of_birth)){
6          $result = true;
7      }
8  ▼ else{
9      $result = false;
10 }
11 return $result;
12 }
13
14 ▼ function createUser($conn, $name, $address, $age, $zip, $gender, $date_of_birth){
15     $sql = "INSERT INTO customer (full_name, address, email, zip, gender, date_of_birth) VALUES(?, ?, ?, ?, ?, ?);";
16     $stmt = mysqli_stmt_init($conn);
17  ▼ if(!mysqli_stmt_prepare($stmt, $sql)){
18     header("location: customer.php?error=stmtfailed");
19     exit();
20 }
21
22     mysqli_stmt_bind_param($stmt, "ssssss", $name, $address, $age, $zip, $gender, $date_of_birth);
23     mysqli_stmt_execute($stmt);
24     mysqli_stmt_close($stmt);
25     header("location: customer.php?new_member_added_successfully");
26     exit();
27 }
28
29 |

```

Figure 7.4 – Functions required to add a new customer

```

1  <?php
2      session_start()
3  ?>
4  <!-- the actual sign up page with which the user interacts, the header is almost identical to index page-->
5  ▼ <head lang="en">
6      <title>New Member</title>
7      <link rel="stylesheet" href="style.css">
8      <meta charset="utf-8">
9      <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
10     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
11     </head>
12
13 ▼ <body>
14
15 ▼     <section class="header" style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner.jpg);>
16         <h1>New Member Form</h1><br>
17         <div class="input-group"> <!-- this is where user inputs data, action is taken in the signup script-->
18             <form action="assign.php" class="form-control" method="post">
19                 <p><input type="text" name="name" placeholder=" Name"></p>
20                 <p><input type="text" name="address" placeholder=" Address"></p>
21                 <p><input type="text" name="email" placeholder=" Email"></p>
22                 <p><input type="int" name="zip" placeholder=" Zip"></p>
23                 <p><input type="text" name="gender" placeholder=" Gender"></p>
24                 <p><input type="date" name="date_of_birth" placeholder=" Date of birth"></p>
25                 <button type="submit" class="input-group-text btn" name="submit">Submit!</button><br>
26                 <p><a href="customer.php">Existing Members</a></p>
27             </form>
28         </div>
29
30         <?php
31             // what the user will be shown upon omitting errors or successfully signing up
32             if(isset($_GET["error"])){
33                 if ($_GET["error"] == "emptyinput"){
34                     echo "<p><em>Please fill in all the fields!</em></p>";
35                 }
36                 else if ($_GET["error"] == "none"){
37                     echo "<p><em>Oh! Member added successfully</em></p>";
38                 }
39                 else if ($_GET["error"] == "stmtfailed"){
40                     echo "<p><em>Oh! Something went wrong. Please try again.</em></p>";
41                 }
42             }
43
44         ?>
45
46     </section>
47
48     </body>

```

Figure 7.5 – Form for a new customer, includes error handlers

```

1  <?php
2  //occur when submit
3 ▼ if(isset($_POST["submit"])){
4      //the different inputs submitted via post function
5      $name = $_POST["name"];
6      $address = $_POST["address"];
7      $age = $_POST["email"];
8      $zip = $_POST["zip"];
9      $gender = $_POST["gender"];
10     $date_of_birth = $_POST["date_of_birth"];
11
12     require_once 'connection.php';
13     require_once 'functions.php';
14     //error handlers
15     ▼ if(emptyInputSignup($name, $address, $age, $zip, $gender, $date_of_birth) !== false){
16         header("location: add.php?error=emptyinput");
17         exit();
18     }
19
20     //if everything goes well with no errors a user is created, i.e. placed in the database
21     createUser($conn, $name, $address, $age, $zip, $gender, $date_of_birth);
22 }
23
24 ▼ else{
25     //redirects user to customer page upon success
26     header("location: customer.php");
27 }
28

```

Figure 7.6 – Adding the new customer to the database, also includes an error handler

```

1  <?php
2
3      require_once("connection.php");
4
5 ▼ if(isset($_GET['del'])){
6
7      $rows['id'] = $_GET['del'];
8      $query = "DELETE FROM customer WHERE id = '". $rows['id'] ."'";
9      $result = mysqli_query($conn, $query);
10
11    ▼ if($result){
12        header("location:customer.php");
13    }
14    ▼ else{
15        echo "Deletion failed!";
16    }
17
18    ▼ else{
19        header("location:customer.php");
20    }
21
22 ?>

```

Figure 7.7 – Delete an existing customer

```

1  <?php
2      include_once('connection.php');
3      $query = "SELECT * FROM courses_overview";
4      $result = mysqli_query($conn, $query);
5  ?>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <title>Staff</title>
10     <meta charset="utf-8">
11     <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
12     <style>
13         table{
14             width:1000px; line-height:55px; margin:auto; margin-top:60px; text-align: center; border: 1px solid black; border-collapse: collapse;
15         }
16         h1{
17             text-align: center; padding-top: 20px; font: helvetica;
18         }
19         tr: {background-color: #ADD8E6;}
20         td{
21             border: 1px solid #F8F8FF;
22         }
23     </style>
24  </head>
25  <body style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner2.jpg);">
26
27      <?php include 'navbar.php'?>
28      <h1 style="color:white">Courses:</h1>
29      <table align = "center">
30          <tr style="font-size:20px; background-color: #0072A0; color: white;">
31              <th style="text-align:center">Student</th>
32              <th style="text-align:center">Email</th>
33              <th style="text-align:center">Dob</th>
34              <th style="text-align:center">Course Title</th>
35              <th style="text-align:center">Instructor</th>
36              <th style="text-align:center">Start Date</th>
37              <th style="text-align:center">City</th>
38          </tr>
39          <?php
40              while($rows = mysqli_fetch_array($result)){
41
42                  <tr style="background-color:#ADD8E6; border: 1px solid azure;">
43                      <td><?php echo $rows['Student']?></td>
44                      <td><?php echo $rows['Student_Email']?></td>
45                      <td><?php echo $rows['Student_Dob']?></td>
46                      <td><?php echo $rows['Course_Title']?></td>
47                      <td><?php echo $rows['Instructor']?></td>
48                      <td><?php echo $rows['Course_Starts']?></td>
49                      <td><?php echo $rows['City']?></td>
50                  </tr>
51
52              </?php
53          ?>
54      </table><br>
55  </body>
56  </html>

```

Figure 7.8 – Displaying the view of courses overview

```

1  <?php
2      include_once('connection.php');
3      $query = "SELECT * FROM lecturer";
4      $result = mysqli_query($conn, $query);
5  ?>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <title>Staff</title>
10     <meta charset="utf-8">
11     <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
12     <style>
13         table{
14             width:1000px; line-height:55px; margin:auto; margin-top:60px; text-align: center; border: 1px solid black; border-collapse: collapse;
15         }
16         h1{
17             text-align: center; padding-top: 20px; font: helvetica;
18         }
19         tr: {background-color: #ADD8E6;}
20         td{
21             border: 1px solid #F8F8FF;
22         }
23     </style>
24  </head>
25  <body style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner2.jpg);">
26
27      <?php include 'navbar.php'?>
28      <h1 style="color:white">Lecturers:</h1>
29      <table align = "center">
30          <tr style="font-size:20px; background-color: #0072A0; color: white;">
31              <th style="text-align:center">Lecturer_ID</th>
32              <th style="text-align:center">Name</th>
33              <th style="text-align:center">Email</th>
34              <th style="text-align:center">Gender</th>
35              <th style="text-align:center">Speciality</th>
36          </tr>
37          <?php
38              while($rows = mysqli_fetch_array($result)){
39
40                  <tr style="background-color:#ADD8E6; border: 1px solid azure;">
41                      <td><?php echo $rows['id']?></td>
42                      <td><?php echo $rows['full_name']?></td>
43                      <td><?php echo $rows['email']?></td>
44                      <td><?php echo $rows['gender']?></td>
45                      <td><?php echo $rows['speciality']?></td>
46                  </tr>
47
48              </?php
49          ?>
50      </table><br>
51
52  </body>
53  </html>

```

Figure 7.9 – The lecturer's table

```

1  <?php
2      include_once('connection.php');
3      $query = "SELECT * FROM lecture_overview";
4      $result = mysqli_query($conn, $query);
5  ?>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <title>Staff</title>
10     <meta charset="utf-8">
11     <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
12  <style>
13      table{
14          width:1000px; line-height:55px; margin:auto; margin-top:60px; text-align: center; border: 1px solid black; border-collapse: collapse;
15      }
16      h1{
17          text-align: center; padding-top: 20px; font: helvetica;
18      }
19      tr{ background-color: #ADD8E6; }
20      td{
21          border: 1px solid #F8F8FF;
22      }
23  </style>
24  </head>
25  <body style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner2.jpg);">
26
27      <?php include 'navbar.php'?>
28      <h1 style="color:white">Lectures:</h1>
29  <table align = "center">
30      <tr style="font-size:20px; background-color: #0072A0; color: white; text-align:center">
31          <th style="text-align:center">Attendee</th>
32          <th style="text-align:center">Attendee Email</th>
33          <th style="text-align:center">Lecture</th>
34          <th style="text-align:center">Lecturer</th>
35          <th style="text-align:center">Lecturer Speciality</th>
36          <th style="text-align:center">Venue</th>
37          <th style="text-align:center">City</th>
38      </tr>
39
40      <?php
41          while($rows = mysqli_fetch_array($result)){
42
43              <tr style="background-color:#ADD8E6; border: 1px solid azure;">
44                  <td><?php echo $rows['Attendee']?></td>
45                  <td><?php echo $rows['Attendee_Email']?></td>
46                  <td><?php echo $rows['Lecture']?></td>
47                  <td><?php echo $rows['Lecturer']?></td>
48                  <td><?php echo $rows['Lecturer_Speciality']?></td>
49                  <td><?php echo $rows['Venue']?></td>
50                  <td><?php echo $rows['City']?></td>
51
52          <?php
53      }
54  </table><br>
55  </body>
56  </html>

```

Figure 7.10 – Displaying the view of lectures overview

```

1  <?php
2      include_once('connection.php');
3      $query = "SELECT * FROM historical_books";
4      $result = mysqli_query($conn, $query);
5  ?>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <title>Staff</title>
10     <meta charset="utf-8">
11     <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
12  <style>
13      table{
14          width:1000px; line-height:55px; margin:auto; margin-top:60px; text-align: center; border: 1px solid black; border-collapse: collapse;
15      }
16      h1{
17          text-align: center; padding-top: 20px; font: helvetica;
18      }
19      tr{ background-color: #ADD8E6; }
20      td{
21          border: 1px solid #F8F8FF;
22      }
23  </style>
24  </head>
25  <body style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner2.jpg);">
26
27      <?php include 'navbar.php'?>
28      <h1 style="color:white">Historical Books:</h1>
29  <table align = "center">
30      <tr style="font-size:20px; background-color: #0072A0; color: white;">
31          <th style="text-align:center">ISBN</th>
32          <th style="text-align:center">Title</th>
33          <th style="text-align:center">Author</th>
34          <th style="text-align:center">Publisher</th>
35      </tr>
36
37      <?php
38          while($rows = mysqli_fetch_array($result)){
39
40              <tr style="background-color:#ADD8E6; border: 1px solid azure;">
41
42                  <td><?php echo $rows['ISBN']?></td>
43                  <td><?php echo $rows['title']?></td>
44                  <td><?php echo $rows['author']?></td>
45                  <td><?php echo $rows['publisher']?></td>
46
47          <?php
48      }
49
50  </table><br>
51
52  </body>
53  </html>

```

Figure 7.11 – Table for historical books

```

1  <?php
2      include_once('connection.php');
3      $query = "SELECT * FROM historical_books_orders";
4      $result = mysqli_query($conn, $query);
5  ?>
6  <!DOCTYPE html>
7  <html>
8  <head>
9      <title>Staff</title>
10     <meta charset="utf-8">
11     <meta name="viewport" content="width = device-width, initial-scale = 1.0, maximum-scale=1.0, user-scalable = no">
12     <style>
13         table{
14             width:1000px; line-height:55px; margin:auto; margin-top:60px; text-align: center; border: 1px solid black; border-collapse: collapse;
15         }
16         h1{
17             text-align: center; padding-top: 20px; font: helvetica;
18         }
19         tr {background-color: #ADD8E6;}
20         td{
21             border: 1px solid #F8F8FF;
22         }
23     </style>
24 </head>
25 <body style="background-image: linear-gradient(rgba(0,0,0,0.6), rgba(0,0,0,0.6)), url(banner2.jpg);">
26
27     <?php include 'navbar.php'?>
28
29     <h1 style="color:white">Books Order:</h1>
30     <table align = "center">
31         <tr style="font-size:20px; background-color: #0072A0; color: white;">
32             <th style="text-align:center">Customer</th>
33             <th style="text-align:center">Book</th>
34             <th style="text-align:center">Author</th>
35             <th style="text-align:center">Email</th>
36             <th style="text-align:center">Address</th>
37             <th style="text-align:center">Purchase Date</th>
38         </tr>
39         <?php
40             while($rows = mysqli_fetch_array($result)){
41                 <tr style="background-color:#ADD8E6; border: 1px solid azure;">
42                     <td><?php echo $rows['Customer']?></td>
43                     <td><?php echo $rows['Book']?></td>
44                     <td><?php echo $rows['Author']?></td>
45                     <td><?php echo $rows['Customer_Email']?></td>
46                     <td><?php echo $rows['Customer_Address']?></td>
47                     <td><?php echo $rows['Purchase_Date']?></td>
48                 </tr>
49                 <?php
50                 }
51             </?php
52             ?
53         </table><br>
54     </body>
55 </html>

```

Figure 7.12 – The view representing orders placed on historical books

The front-end web interface can be seen as follows:

ID	Name	Address	Email	Zip	Gender	Date of Birth	Delete
1	Christopher Jones	347 Cedar St	christopherjones@bp.com	30044	M	1958-09-11	Delete
2	Matthew Martinez	602 Main Place	matthewmartinez@ge.com	92336	M	1961-09-04	Delete
3	Melissa Moore	463 Park Rd	melissamoore@aramark.com	8701	M	1967-08-27	Delete
4	Melissa Brown	712 View Ave	melissabrown@verizon.com	77084	F	1948-06-14	Delete
5	Jennifer Thomas	231 Elm St	jenniferthomas@aramark.com	78572	F	1998-03-14	Delete
6	Stephanie Martinez	386 Second St	stephaniemartinez@albertsons.com	8701	M	1998-01-24	Delete
7	Daniel Williams	107 Pine St	danielwilliams@tjx.com	77449	F	1985-07-20	Delete
8	Lauren Anderson	13 Maple Ave	laurenanderson@pepsi.com	92503	F	1973-09-09	Delete
9	Michael Jackson	818 Pine Ave	michaeljackson@disney.com	78572	F	1951-03-03	Delete
10	Ashley Johnson	874 Oak Ave	ashleyjohnson@boeing.com	91331	F	1977-05-10	Delete
11	Brittany Thomas	187 Maple Ave	brittanythomas@walmart.com	78521	F	1986-10-22	Delete

Figure 7.13 – Customer table, portrayed as members, on front-end

localhost/db_web_interface/add.php

New Member Form

Khizar Mukhtiar
4000 Broadway
khizar@mukhtiar.com
25000
M
24/11/1995

Submit!

[*Existing Members](#)

Figure 7.14 – Form for signing up a new member

> localhost/db_web_interface/customer.php?new_member_added_successfully

9	Michael Jackson	818 Pine Ave	michaeljackson@disney.com	78572	F	1951-03-03	Delete
10	Ashley Johnson	874 Oak Ave	ashleyjohnson@boeing.com	91331	F	1977-05-10	Delete
11	Brittany Thomas	187 Maple Ave	brittanythomas@walmart.com	78521	F	1986-10-22	Delete
12	Tiffany Smith	123 Lake St	tiffanysmith@ups.com	78521	F	1950-06-16	Delete
13	Lauren Wilson	942 Fifth Ave	laurenwilson@target.com	78572	M	1965-12-26	Delete
14	Justin Smith	844 Lake Ave	justinsmith@boeing.com	30044	F	1956-03-16	Delete
15	Jessica Garcia	123 Pine Place	jessicagarcia@toyota.com	92336	F	1996-08-05	Delete
16	Matthew Jackson	538 Cedar Ave	matthewjackson@bp.com	77449	M	1996-02-26	Delete
17	Stephanie Thomas	804 Fourth Place	stephaniethomas@apple.com	78521	F	1988-08-26	Delete
18	Jessica Jackson	235 Pine Place	jessicajackson@aramark.com	60629	F	1991-07-22	Delete
19	James Martinez	831 Oak St	jamesmartinez@kroger.com	78521	F	1987-12-22	Delete
20	Christopher Robinson	754 Cedar St	christopherrobinson@ibm.com	78577	F	1972-06-25	Delete
23	Khizar Mukhtiar	4000 Broadway	khizar@mukhtiar.com	25000	M	1995-11-24	Delete

Figure 7.15 – New member successfully added

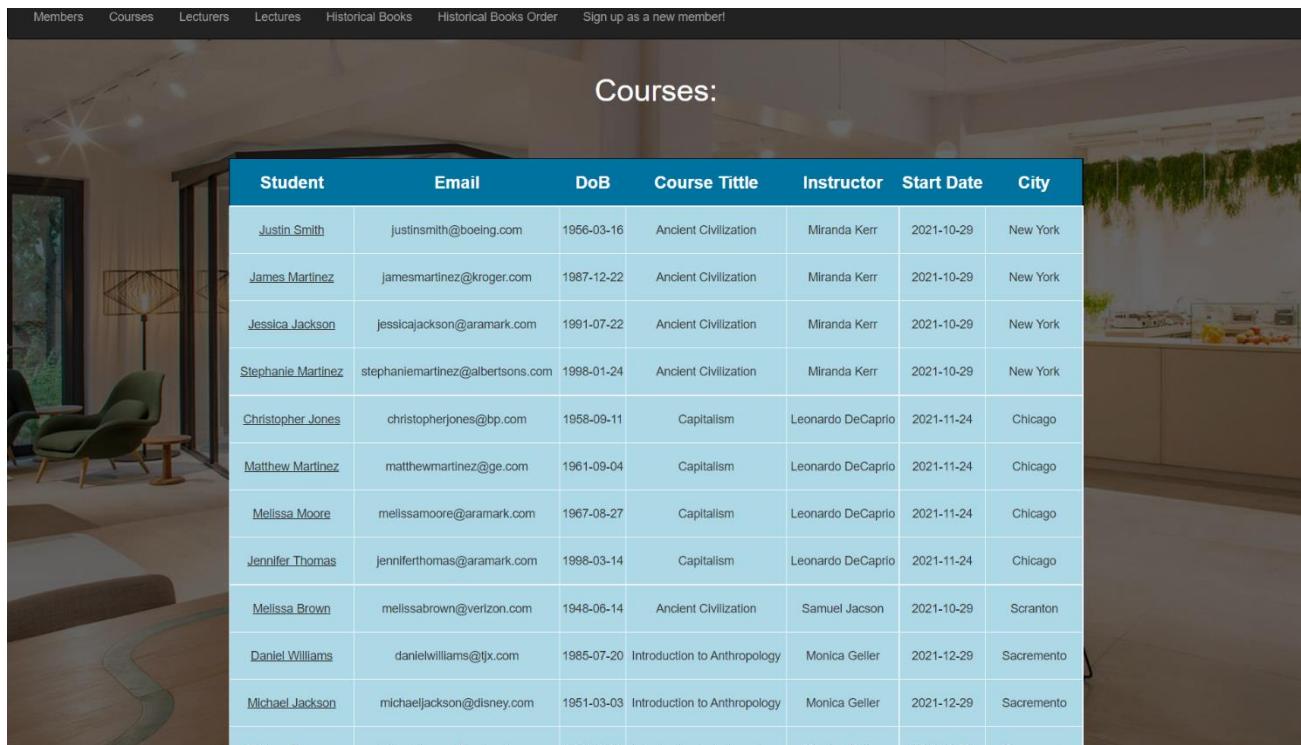


17	Stephanie Thomas	804 Fourth Place	stephaniethomas@apple.com	78521	F	1988-08-26	Delete
18	Jessica Jackson	235 Pine Place	jessicajackson@aramark.com	60629	F	1991-07-22	Delete
19	James Martinez	831 Oak St	jamesmartinez@kroger.com	78521	F	1987-12-22	Delete
20	Christopher Robinson	754 Cedar St	christopherrobinson@ibm.com	78577	F	1972-06-25	Delete
23	Khizar Mukhtiar	4000 Broadway	khizar@mukhtiar.com	25000	M	1995-11-24	Delete

localhost/db_web_interface/delete.php?del=23

16	Matthew Jackson	538 Cedar Ave	matthewjackson@bp.com	77449	M	1996-02-26	Delete
17	Stephanie Thomas	804 Fourth Place	stephaniethomas@apple.com	78521	F	1988-08-26	Delete
18	Jessica Jackson	235 Pine Place	jessicajackson@aramark.com	60629	F	1991-07-22	Delete
19	James Martinez	831 Oak St	jamesmartinez@kroger.com	78521	F	1987-12-22	Delete
20	Christopher Robinson	754 Cedar St	christopherrobinson@ibm.com	78577	F	1972-06-25	Delete

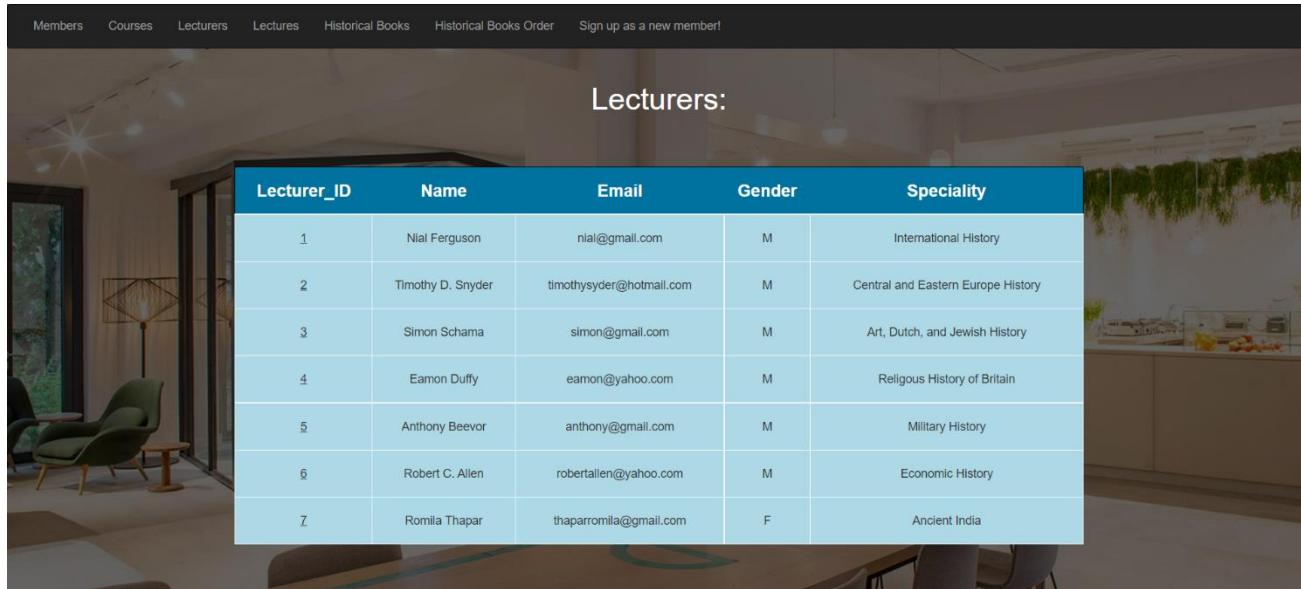
Figure 7.16 – Member deleted from the list with the delete link next to it



Courses:

Student	Email	DoB	Course Title	Instructor	Start Date	City
Justin Smith	justinsmith@boeing.com	1956-03-16	Ancient Civilization	Miranda Kerr	2021-10-29	New York
James Martinez	jamesmartinez@kroger.com	1987-12-22	Ancient Civilization	Miranda Kerr	2021-10-29	New York
Jessica Jackson	jessicajackson@aramark.com	1991-07-22	Ancient Civilization	Miranda Kerr	2021-10-29	New York
Stephanie Martinez	stephaniemartinez@albertsons.com	1998-01-24	Ancient Civilization	Miranda Kerr	2021-10-29	New York
Christopher Jones	christopherjones@bp.com	1958-09-11	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Matthew Martinez	matthewmartinez@ge.com	1961-09-04	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Melissa Moore	melissamoore@aramark.com	1967-08-27	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Jennifer Thomas	jenniferthomas@aramark.com	1998-03-14	Capitalism	Leonardo DeCaprio	2021-11-24	Chicago
Melissa Brown	melissabrown@verizon.com	1948-06-14	Ancient Civilization	Samuel Jackson	2021-10-29	Scranton
Daniel Williams	danielwilliams@ljx.com	1985-07-20	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento
Michael Jackson	michaeljackson@disney.com	1951-03-03	Introduction to Anthropology	Monica Geller	2021-12-29	Sacramento

Figure 7.17 – Fetching the view of courses overview



Lecturers:

Lecturer_ID	Name	Email	Gender	Speciality
1	Nial Ferguson	nial@gmail.com	M	International History
2	Timothy D. Snyder	timothysnyder@hotmail.com	M	Central and Eastern Europe History
3	Simon Schama	simon@gmail.com	M	Art, Dutch, and Jewish History
4	Eamon Duffy	eamon@yahoo.com	M	Religious History of Britain
5	Anthony Beevor	anthony@gmail.com	M	Military History
6	Robert C. Allen	robertallen@yahoo.com	M	Economic History
7	Romila Thapar	thaparromila@gmail.com	F	Ancient India

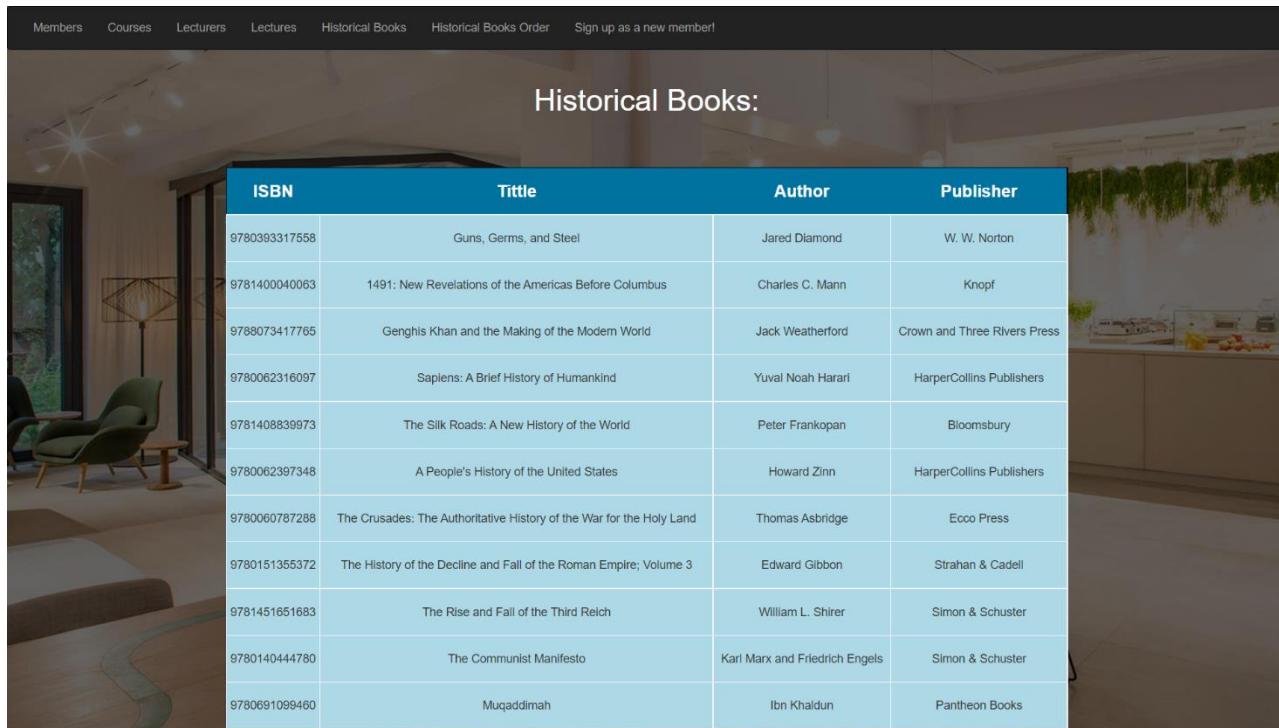
Figure 7.18 – Lecturers table



Lectures:

Attendee	Attendee Email	Lecture	Lecturer	Lecturer Speciality	Venue	City
Jennifer Thomas	jenniferthomas@aramark.com	Seljuk Turks	Timothy D. Snyder	Central and Eastern Europe History	Havemeyer Hall	New York
Christopher Jones	christopherjones@bp.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Matthew Martinez	matthewmartinez@ge.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Melissa Moore	melissamoore@aramark.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Ashley Johnson	ashleyjohnson@boeing.com	Renaissance	Nial Ferguson	International History	Royal Albert Hall	London
Lauren Anderson	laurenanderson@pepsi.com	Genghis Khan	Timothy D. Snyder	Central and Eastern Europe History	DR Koncerthuset	Copenhagen
Jessica Garcia	jessicagarcia@toyota.com	Genghis Khan	Timothy D. Snyder	Central and Eastern Europe History	DR Koncerthuset	Copenhagen
Daniel Williams	danielwilliams@tjx.com	World War	Anthony Beevor	Military History	Tonhalle Dusseldorf	Dusseldorf

Figure 7.19 – Fetching the view of lectures overview



Historical Books:

ISBN	Title	Author	Publisher
9780393317558	Guns, Germs, and Steel	Jared Diamond	W. W. Norton
9781400040063	1491: New Revelations of the Americas Before Columbus	Charles C. Mann	Knopf
9788073417765	Genghis Khan and the Making of the Modern World	Jack Weatherford	Crown and Three Rivers Press
9780062316097	Sapiens: A Brief History of Humankind	Yuval Noah Harari	HarperCollins Publishers
9781408839973	The Silk Roads: A New History of the World	Peter Frankopan	Bloomsbury
9780062397348	A People's History of the United States	Howard Zinn	HarperCollins Publishers
9780060787288	The Crusades: The Authoritative History of the War for the Holy Land	Thomas Asbridge	Ecco Press
9780151355372	The History of the Decline and Fall of the Roman Empire; Volume 3	Edward Gibbon	Strahan & Cadell
9781451651683	The Rise and Fall of the Third Reich	William L. Shirer	Simon & Schuster
9780140444780	The Communist Manifesto	Karl Marx and Friedrich Engels	Simon & Schuster
9780691099460	Muqaddimah	Ibn Khaldun	Pantheon Books

Figure 7.20 – Historical books table



The screenshot shows a web application interface. At the top, there is a navigation bar with links: Members, Courses, Lecturers, Lectures, Historical Books, Historical Books Order, and Sign up as a new member!. Below the navigation bar, the title "Books Order:" is displayed. The main content is a table titled "Books Order:" with a blue header row containing the columns: Customer, Book, Author, Email, Address, and Purchase Date.

Customer	Book	Author	Email	Address	Purchase Date
Christopher Jones	Guns, Germs, and Steel	Jared Diamond	christopherjones@bp.com	347 Cedar St	2021-10-29
Matthew Martinez	Guns, Germs, and Steel	Jared Diamond	matthewmartinez@ge.com	602 Main Place	2021-10-24
Matthew Martinez	1491: New Revelations of the Americas Before Columbus	Charles C. Mann	matthewmartinez@ge.com	602 Main Place	2021-11-14
Matthew Martinez	1491: New Revelations of the Americas Before Columbus	Charles C. Mann	matthewmartinez@ge.com	602 Main Place	2021-07-24
Matthew Martinez	Genghis Khan and the Making of the Modern World	Jack Weatherford	matthewmartinez@ge.com	602 Main Place	2021-12-28
Matthew				602 Main	

Figure 7.21 – Fetching the view of ordered books

8. Conclusion

This report commenced with an overview of what was necessary prior to the development of the database for the case study. This was followed by the implementation with MySQL and after running some queries, optimization, and security measures on the database, an actual working prototype of a web-interface was developed that showed the necessary information from the database.

9. References

1. Internet Live Stats, 2021:
<https://www.internetlivestats.com/internet-users/>
[Last accessed: 18/10/2021]
2. EssentialSQL, 2014:
<http://www.essentialsql.com/what-is-a-database-table/>
[Last accessed: 18/10/2021]
3. Kejser, 2011:
<https://kejserbi.wordpress.com/2011/04/18/defining-the-good-data-model/>
[Last accessed: 18/10/2021]
4. Thomas Connolly & Carolyn Begg. Database System: A Practical Approach to Design, Implementation, and Management 2014; Chapter 2.
5. Biscobing, 2019:
<https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD>
[Last accessed: 18/10/2021]
6. Isaac, 2021:
https://isaaccomputerscience.org/concepts/data_dbs_normalisation?examBoard=all&stage=all
[Last accessed: 19/10/2021]
7. Jayaram, 2018:
<https://www.sqlshack.com/what-is-database-normalization-in-sql-server/>
[Last accessed: 19/10/2021]
8. Barnes, 2018:
<https://www.tutorialspoint.com/Boyce-Codd-Normal-Form-BCNF>
[Last accessed: 19/10/2021]

10. Appendix

1. Link for SQL text file that contains all the queries of the whole database (file obtained by the export function in MySQL Workbench):

<https://drive.google.com/file/d/1jmAyAahBFvWVt0ljHu2gOHJIQFOioVBY/view?usp=sharing>

2. Background images used as banners for the web-interface:



3. Folder containing all the files for the web-interface:

This PC > OS (C:) > xampp > htdocs > db_web_interface				
	Name	Date modified	Type	Size
data	add	27/10/2021 09:12	PHP File	3 KB
face	assign	26/10/2021 08:39	PHP File	1 KB
	banner	23/07/2021 15:18	JPG File	113 KB
	banner2	23/07/2021 16:15	JPG File	206 KB
	books_order	27/10/2021 08:12	PHP File	3 KB
	connection	26/10/2021 07:45	PHP File	1 KB
ts	courses_overview	27/10/2021 08:09	PHP File	3 KB
ts	customer	26/10/2021 13:25	PHP File	3 KB
ts	delete	27/10/2021 08:29	PHP File	1 KB
ts	functions	26/10/2021 08:45	PHP File	1 KB
	historical_books	26/10/2021 10:08	PHP File	2 KB
	lecturer	26/10/2021 10:08	PHP File	3 KB
	lectures_overview	27/10/2021 08:08	PHP File	3 KB
	navbar	26/10/2021 09:33	PHP File	2 KB
	style	27/10/2021 09:11	CSS File	5 KB