THEATRE TICKET BOOKING SYSTEM

SQL Database

Abstract

This Report includes the Entity Diagram, SQL Code, Execution proof screenshots of the Database created for a Theatre ticket Booking System

Anusigan Sivananthan
sivananthananusigan@gmail.com

I. Table of Contents

II. List of Figures

01.Assumptions for Creating Data Base for Theatre Ticket Booking System

- A movie can be associated with multiple screenings in different theatres.
- A customer can make multiple seat bookings for different shows.
- Theatres can host multiple screenings simultaneously.
- Seat rows and their details are specific to a particular theatre.
- A movie can have multiple screenings across different theatres.
- Each theatre can have multiple seat rows, and a seat row is specific to a particular theatre.
- The relationship between Theatre and SeatRow is represented by ‘Has’ relationship.
- The relationship between Customer and Movie is represented by ‘Searches’ relationship.
- The relationship between between Movie and Movie_Show is represented by ‘Has’ relationship.
- The relationship between Movie and Screening is represented by ‘Screens’ relationship.

02.Entity Relationship Diagaram

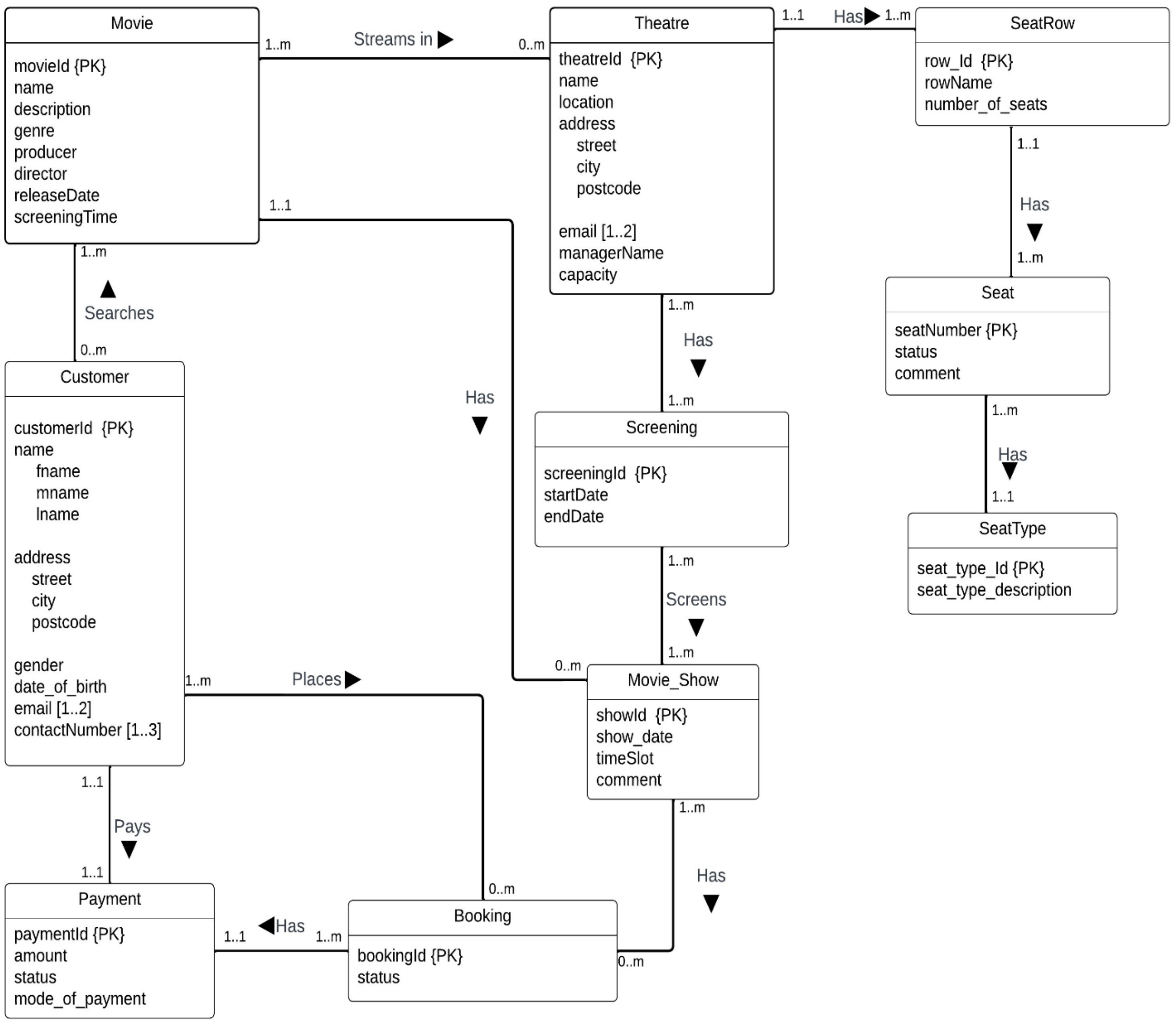


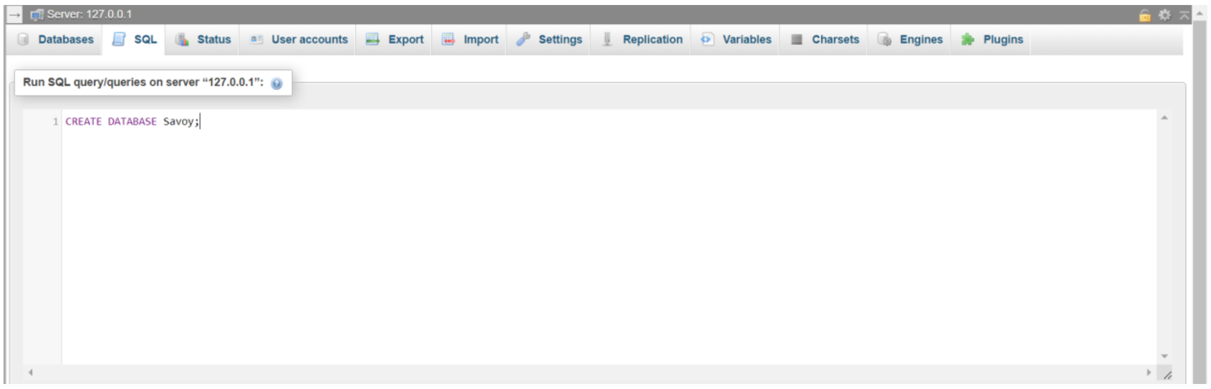
Figure 1-Entity relationship diagram

03.Creating Database for Theatre Seat booking System

DDL Statement

```
CREATE DATABASE Savoy;
```

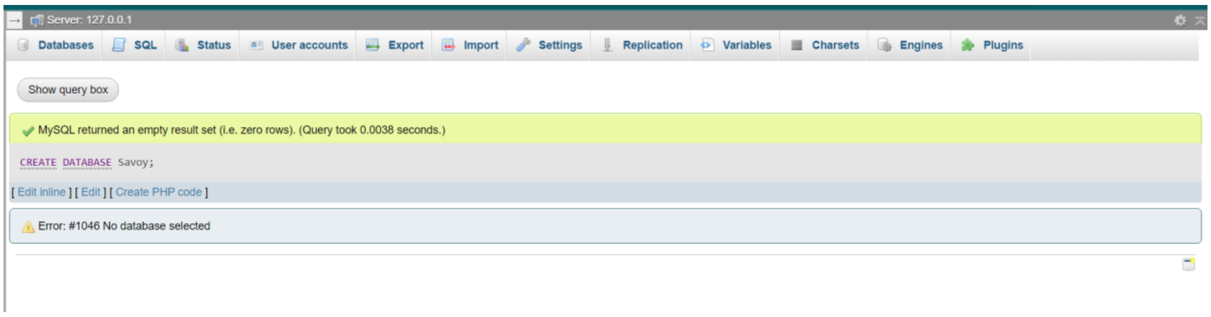
SQL Query



```
CREATE DATABASE Savoy;
```

Figure 2-SQL Query for Savoy Database

Query execution proof



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0038 seconds.)
CREATE DATABASE Savoy;
Error: #1046 No database selected
```

Figure 3-Query execution proof of database

Table Structure

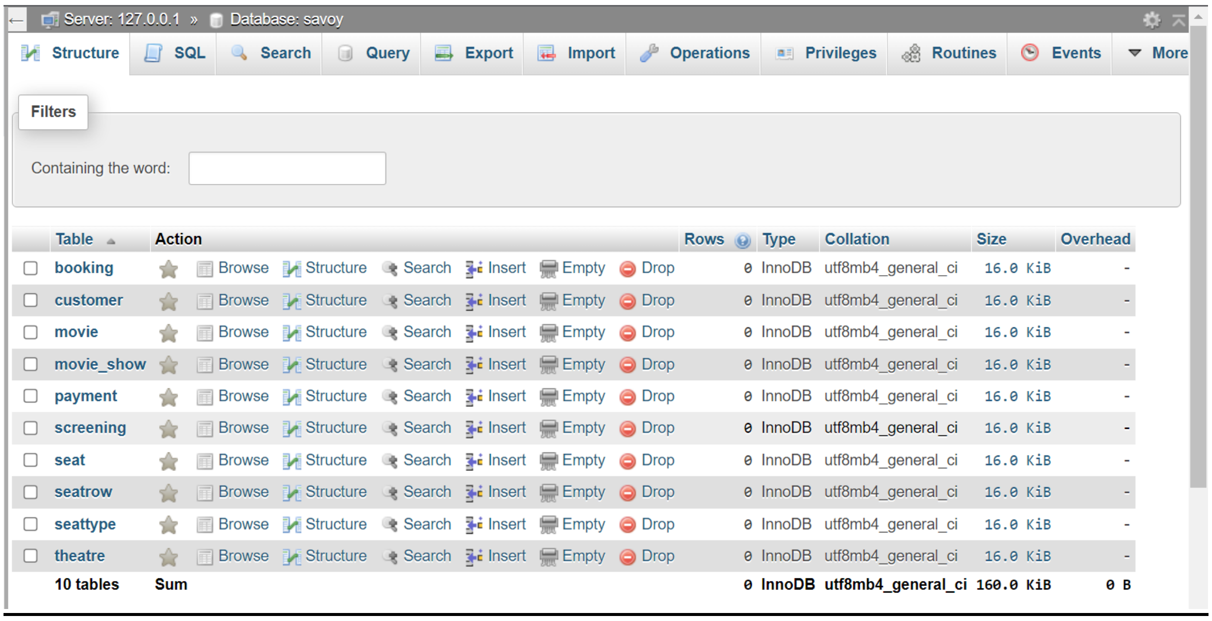


Table	Action	Rows	Type	Collation	Size	Overhead
booking	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
customer	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
movie	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
movie_show	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
payment	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
screening	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
seat	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
seatrow	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
seattype	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
theatre	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
10 tables	Sum	0	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

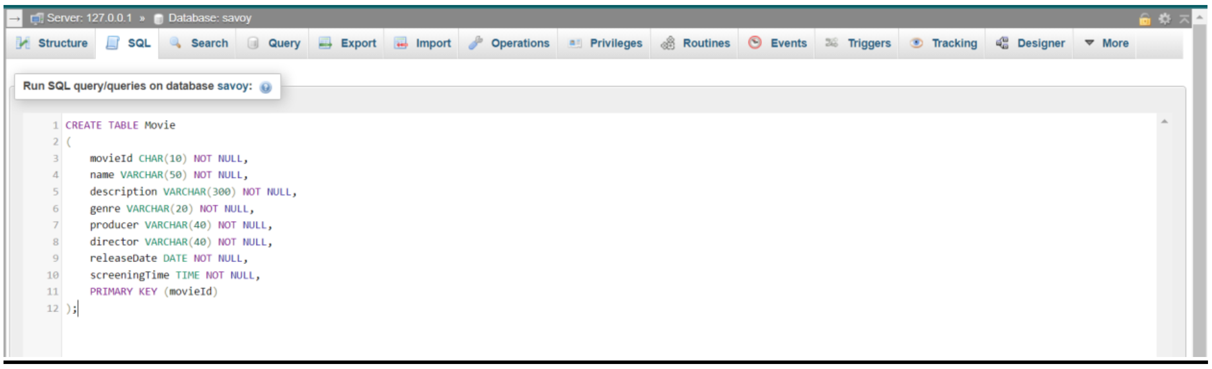
Figure 4-Table Structure of Savoy database

Creating Movie Table

DDL Statement

```
CREATE TABLE Movie
(
        movieId CHAR(10) NOT NULL,
        name VARCHAR(50) NOT NULL,
        description VARCHAR(300) NOT NULL,
        genre VARCHAR(20) NOT NULL,
        producer VARCHAR(40) NOT NULL,
        director VARCHAR(40) NOT NULL,
        releaseDate DATE NOT NULL,
        screeningTime TIME NOT NULL,
        PRIMARY KEY (movieId)
);
```

SQL Query

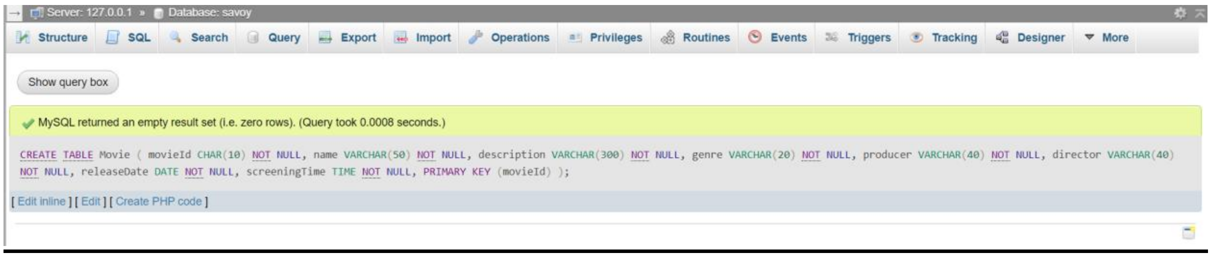


Run SQL query/queries on database savoy:

```
CREATE TABLE Movie
(
    movieId CHAR(10) NOT NULL,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(300) NOT NULL,
    genre VARCHAR(20) NOT NULL,
    producer VARCHAR(40) NOT NULL,
    director VARCHAR(40) NOT NULL,
    releaseDate DATE NOT NULL,
    screeningTime TIME NOT NULL,
    PRIMARY KEY (movieId)
);
```

Figure 5-SQL Query for movie table

Query execution proof

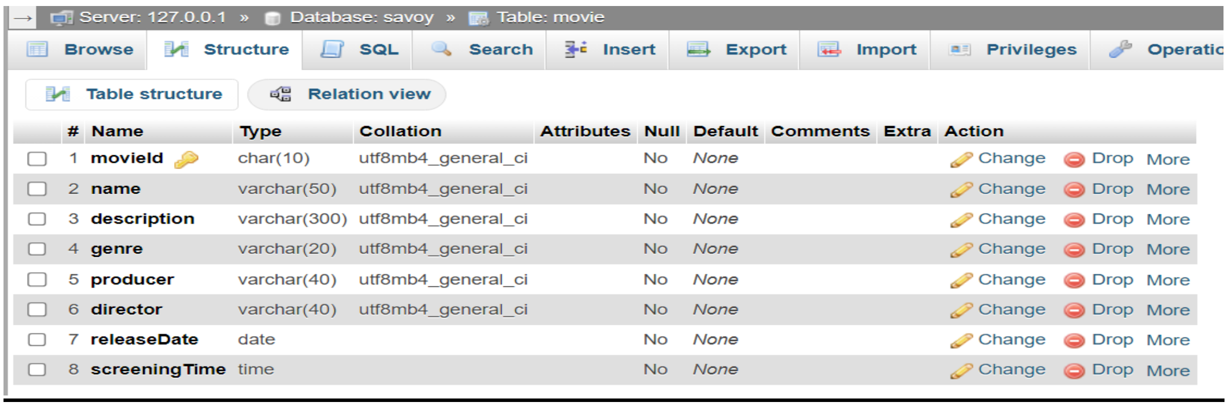


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
CREATE TABLE Movie ( movieId CHAR(10) NOT NULL, name VARCHAR(50) NOT NULL, description VARCHAR(300) NOT NULL, genre VARCHAR(20) NOT NULL, producer VARCHAR(40) NOT NULL, director VARCHAR(40)
NOT NULL, releaseDate DATE NOT NULL, screeningTime TIME NOT NULL, PRIMARY KEY (movieId) );
```

Figure 6-Query execution proof for movie table

Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	movieId	char(10)	utf8mb4_general_ci		No	None			Change Drop More
2	name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	description	varchar(300)	utf8mb4_general_ci		No	None			Change Drop More
4	genre	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
5	producer	varchar(40)	utf8mb4_general_ci		No	None			Change Drop More
6	director	varchar(40)	utf8mb4_general_ci		No	None			Change Drop More
7	releaseDate	date			No	None			Change Drop More
8	screeningTime	time			No	None			Change Drop More

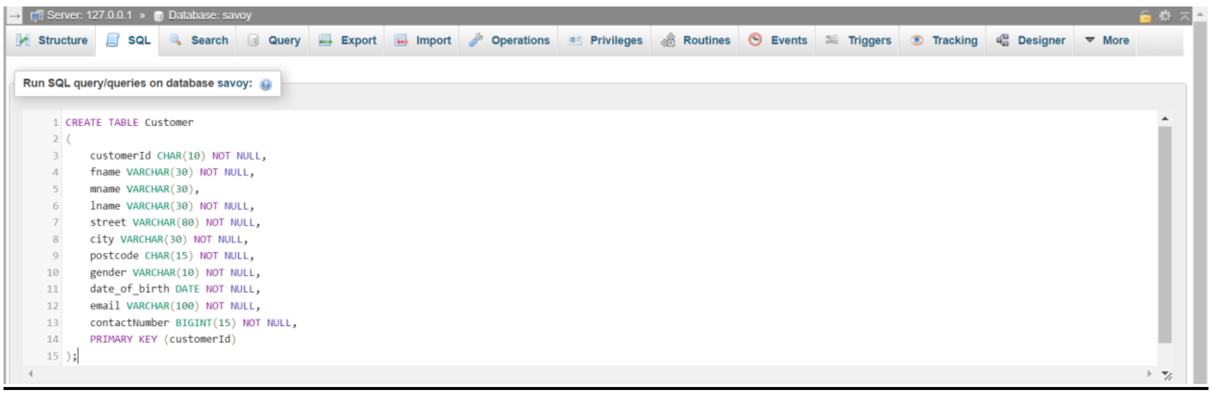
Figure 7-Table Structure of movie

Creating Customer Table

DDL Statement

```
CREATE TABLE Customer
(
        customerId CHAR(10) NOT NULL,
	fname VARCHAR(30) NOT NULL,
	mname VARCHAR(30),
	lname VARCHAR(30) NOT NULL,
	street VARCHAR(80) NOT NULL,
	city VARCHAR(30) NOT NULL,
	postcode CHAR(15) NOT NULL,
	gender VARCHAR(10) NOT NULL,
	date_of_birth DATE NOT NULL,
	email VARCHAR(100) NOT NULL,
	contactNumber BIGINT(15) NOT NULL,
	PRIMARY KEY (customerId)
);
```

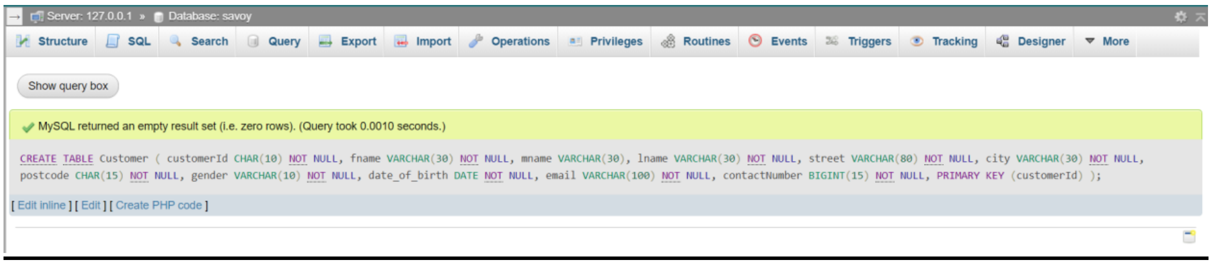
<u>SQL Query</u>



```
CREATE TABLE Customer
(
    customerId CHAR(10) NOT NULL,
    fname VARCHAR(30) NOT NULL,
    mname VARCHAR(30),
    lname VARCHAR(30) NOT NULL,
    street VARCHAR(80) NOT NULL,
    city VARCHAR(30) NOT NULL,
    postcode CHAR(15) NOT NULL,
    gender VARCHAR(10) NOT NULL,
    date_of_birth DATE NOT NULL,
    email VARCHAR(100) NOT NULL,
    contactNumber BIGINT(15) NOT NULL,
    PRIMARY KEY (customerId)
);
```

Figure 8-SQL Query for Customer table

<u>Query execution proof</u>


Server: 127.0.0.1 » Database: savoy
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More
Show query box

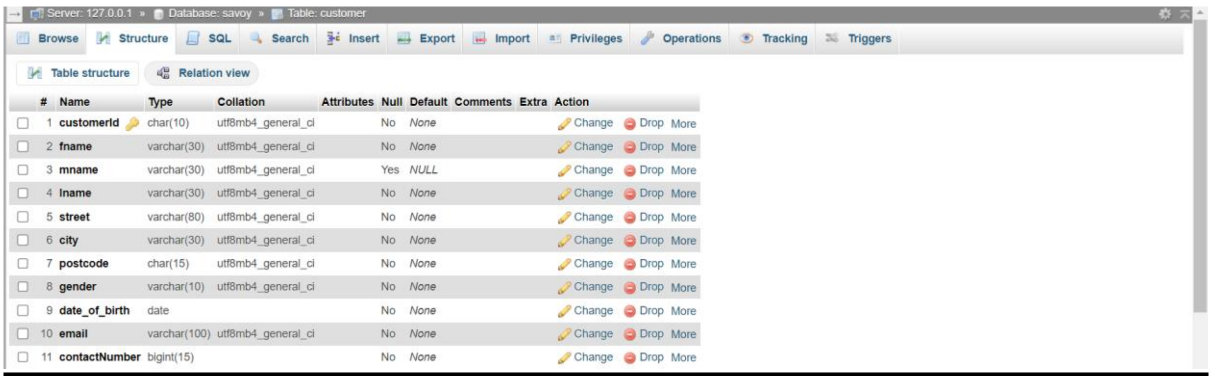
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.)

```
CREATE TABLE Customer ( customerId CHAR(10) NOT NULL, fname VARCHAR(30) NOT NULL, mname VARCHAR(30), lname VARCHAR(30) NOT NULL, street VARCHAR(80) NOT NULL, city VARCHAR(30) NOT NULL,
postcode CHAR(15) NOT NULL, gender VARCHAR(10) NOT NULL, date_of_birth DATE NOT NULL, email VARCHAR(100) NOT NULL, contactNumber BIGINT(15) NOT NULL, PRIMARY KEY (customerId) );
```

[Edit inline] [Edit] [Create PHP code]

Figure 9-Query execution proof for Customer table

Table Structure


Server: 127.0.0.1 » Database: savoy » Table: customer
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	customerId	char(10)	utf8mb4_general_ci		No	None			Change Drop More
2	fname	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	mname	varchar(30)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	lname	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
5	street	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More
6	city	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
7	postcode	char(15)	utf8mb4_general_ci		No	None			Change Drop More
8	gender	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
9	date_of_birth	date			No	None			Change Drop More
10	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
11	contactNumber	bigint(15)			No	None			Change Drop More

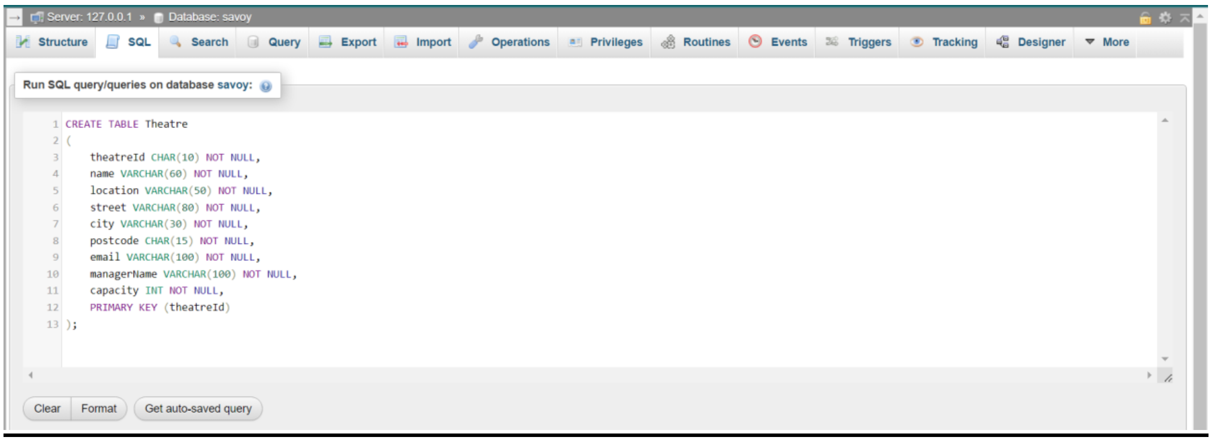
Figure 10-Table Structure of Customer table

Creating Theatre Table

DDL Statement

```
CREATE TABLE Theatre

(

        theatreId CHAR(10) NOT NULL,

        name VARCHAR(60) NOT NULL,

        location VARCHAR(50) NOT NULL,

        street VARCHAR(80) NOT NULL,

        city VARCHAR(30) NOT NULL,

        postcode CHAR(15) NOT NULL,

        email VARCHAR(100) NOT NULL,
        managerName VARCHAR(100) NOT NULL,
        capacity INT NOT NULL,
        PRIMARY KEY (theatreId)
);
```

SQL Query

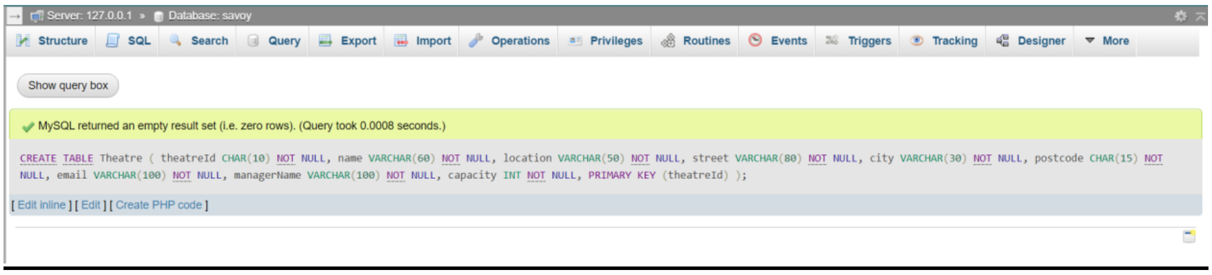


Server: 127.0.0.1 » Database: savoy
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More
Run SQL query/queries on database savoy:
Clear Format Get auto-saved query

```
CREATE TABLE Theatre
(
    theatreId CHAR(10) NOT NULL,
    name VARCHAR(60) NOT NULL,
    location VARCHAR(50) NOT NULL,
    street VARCHAR(80) NOT NULL,
    city VARCHAR(30) NOT NULL,
    postcode CHAR(15) NOT NULL,
    email VARCHAR(100) NOT NULL,
    managerName VARCHAR(100) NOT NULL,
    capacity INT NOT NULL,
    PRIMARY KEY (theatreId)
);
```

Figure 11-SQL Query for Theatre table

Query execution proof

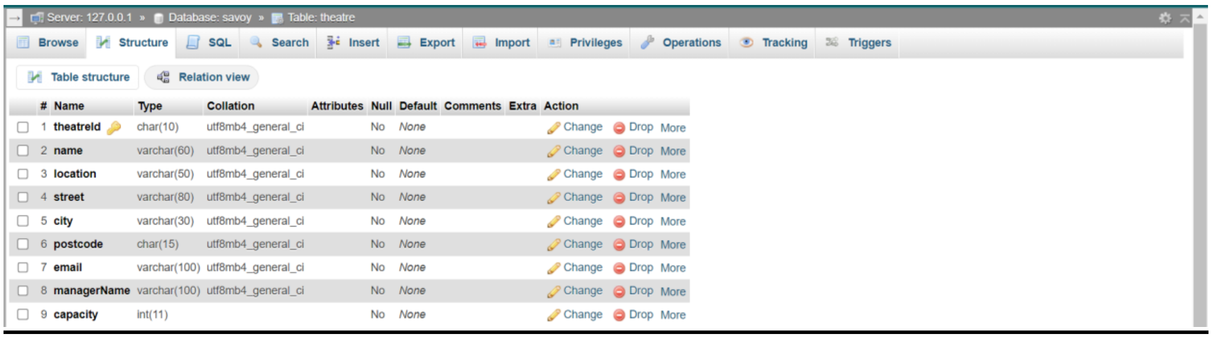


Server: 127.0.0.1 » Database: savoy
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More
Show query box
[Edit inline] [Edit] [Create PHP code]

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

CREATE TABLE Theatre ( theatreId CHAR(10) NOT NULL, name VARCHAR(60) NOT NULL, location VARCHAR(50) NOT NULL, street VARCHAR(80) NOT NULL, city VARCHAR(30) NOT NULL, postcode CHAR(15) NOT
NULL, email VARCHAR(100) NOT NULL, managerName VARCHAR(100) NOT NULL, capacity INT NOT NULL, PRIMARY KEY (theatreId) );
```

Figure 12-Query execution proof for Theatre Table

Table Structure



Server: 127.0.0.1 » Database: savoy » Table: theatre
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	theatreId	char(10)	utf8mb4_general_ci		No	*None*			Change Drop More
2	name	varchar(60)	utf8mb4_general_ci		No	*None*			Change Drop More
3	location	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
4	street	varchar(80)	utf8mb4_general_ci		No	*None*			Change Drop More
5	city	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
6	postcode	char(15)	utf8mb4_general_ci		No	*None*			Change Drop More
7	email	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
8	managerName	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
9	capacity	int(11)			No	*None*			Change Drop More

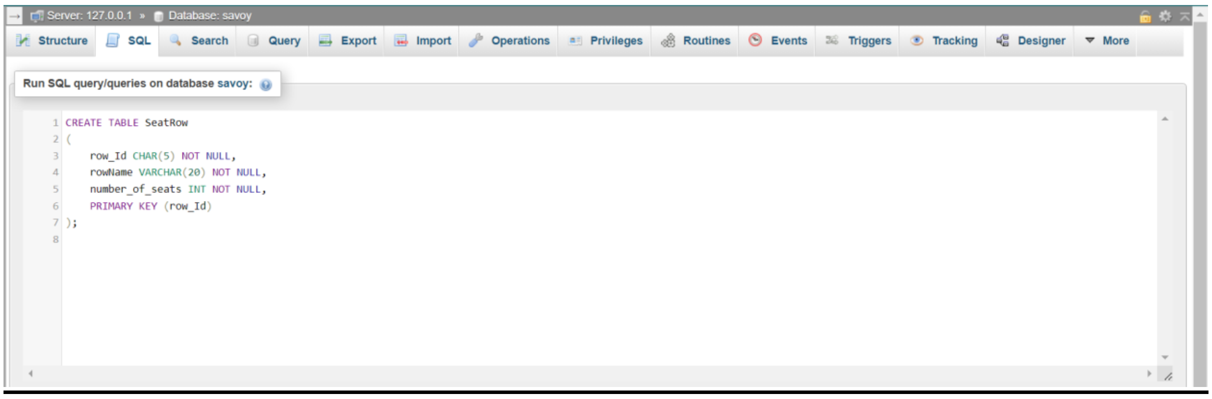
Figure 13-Table structure of Theatre

Creating SeatRow Table

DDL Statement

```
CREATE TABLE SeatRow
(
	row_Id CHAR(5) NOT NULL,
	rowName VARCHAR(20) NOT NULL,
	number_of_seats INT NOT NULL,
	PRIMARY KEY (row_Id)
);
```

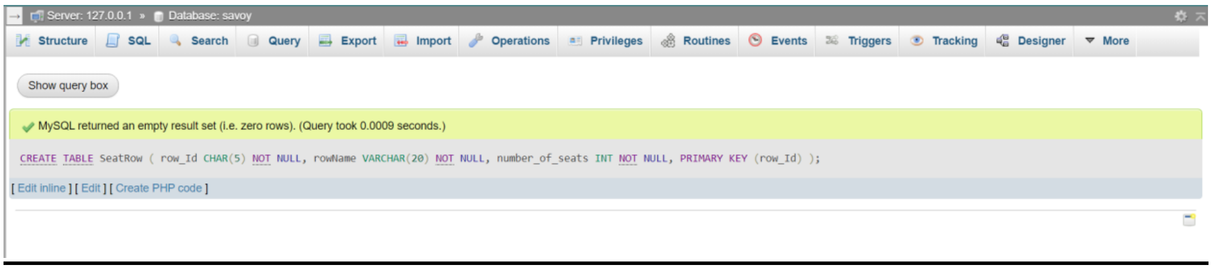
SQL Query



```
CREATE TABLE SeatRow
(
    row_Id CHAR(5) NOT NULL,
    rowName VARCHAR(20) NOT NULL,
    number_of_seats INT NOT NULL,
    PRIMARY KEY (row_Id)
);
```

Figure 14-SQL Query for SeatRow Table

Query execution proof

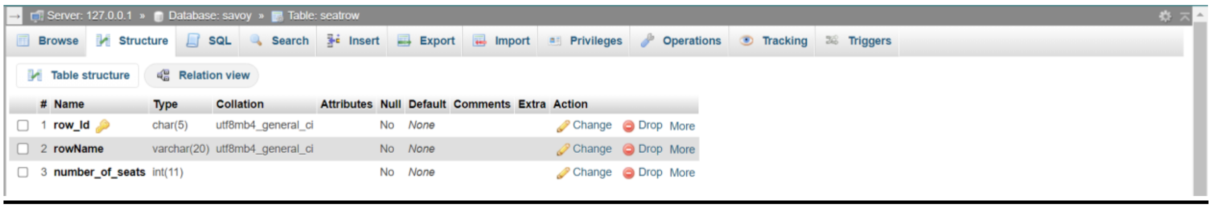


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE SeatRow ( row_Id CHAR(5) NOT NULL, rowName VARCHAR(20) NOT NULL, number_of_seats INT NOT NULL, PRIMARY KEY (row_Id) );
```

Figure 15-Query execution proof for SeatRow Table

Table Structure


Server: 127.0.0.1 » Database: savoy » Table: seatrow
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	row_Id	char(5)	utf8mb4_general_ci		No	None			Change Drop More
2	rowName	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
3	number_of_seats	int(11)			No	None			Change Drop More

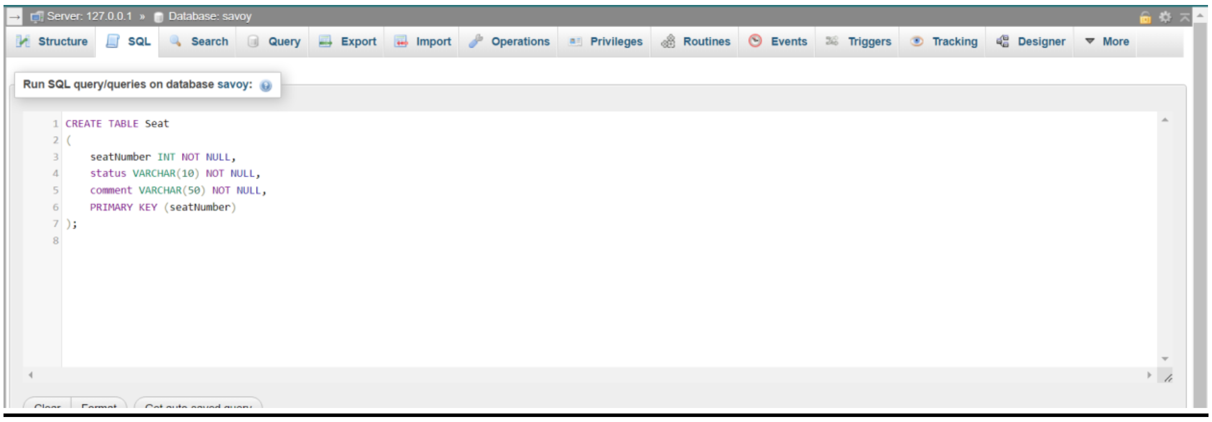
Figure 16-Table Structure of SeatRow

Creating Seat Table

DDL Statement

```
CREATE TABLE Seat
(
	seatNumber INT NOT NULL,
	status VARCHAR(10) NOT NULL,
	comment VARCHAR(50) NOT NULL,
	PRIMARY KEY (seatNumber)
);
```

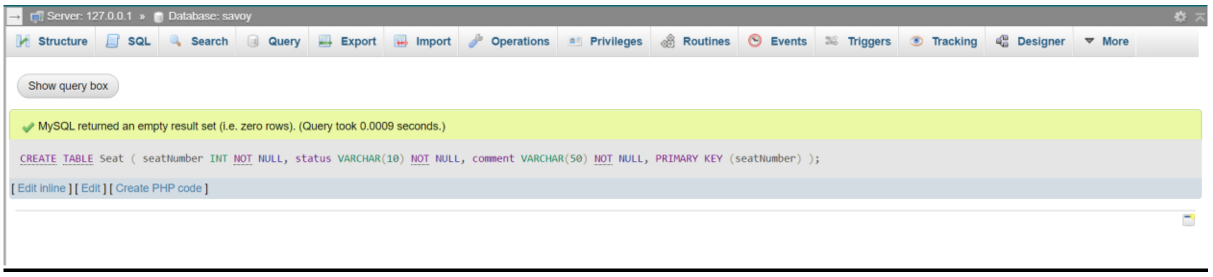
SQL Query


Server: 127.0.0.1 » Database: savoy
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More
Run SQL query/queries on database savoy:

```
CREATE TABLE Seat
(
    seatNumber INT NOT NULL,
    status VARCHAR(10) NOT NULL,
    comment VARCHAR(50) NOT NULL,
    PRIMARY KEY (seatNumber)
);
```

Figure 17-SQL Query for Seat table

Query execution proof


Server: 127.0.0.1 » Database: savoy
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More
Show query box

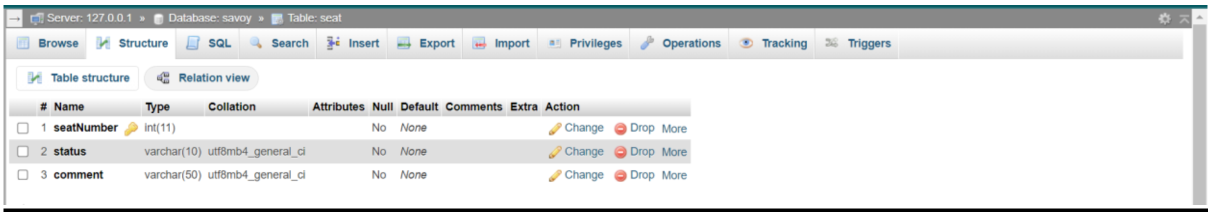
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE Seat ( seatNumber INT NOT NULL, status VARCHAR(10) NOT NULL, comment VARCHAR(50) NOT NULL, PRIMARY KEY (seatNumber) );
```

[Edit inline] [Edit] [Create PHP code]

Figure 18-Query execution proof for Seat table

Table Structure


Server: 127.0.0.1 » Database: savoy » Table: seat
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	seatNumber	int(11)			No	*None*			Change Drop More
2	status	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
3	comment	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More

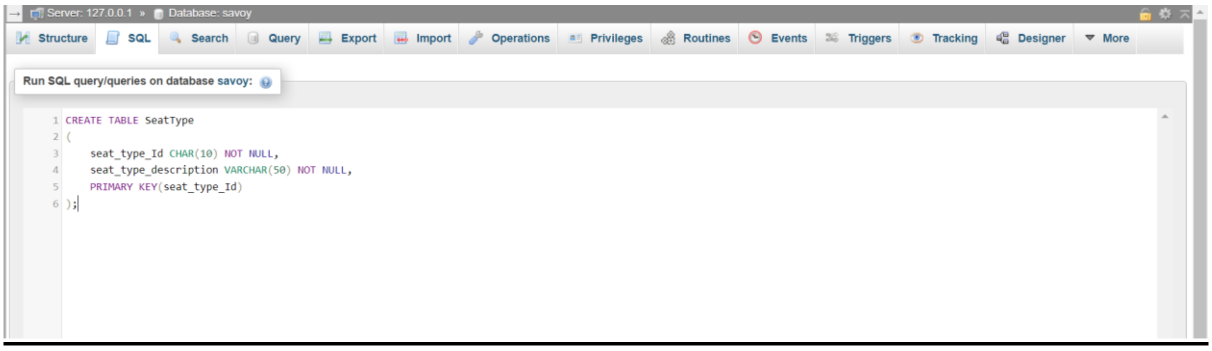
Figure 19-Table Structure of Seat

Creating SeatType Table

DDL Statement

```
CREATE TABLE SeatType
(
	seat_type_Id CHAR(10) NOT NULL,
	seat_type_description VARCHAR(50) NOT NULL,
	PRIMARY KEY(seat_type_Id)
);
```

SQL Query

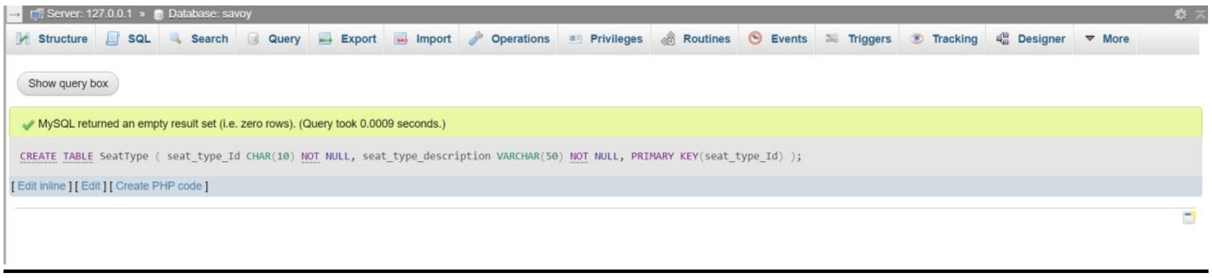


Run SQL query/queries on database savoy:

```
CREATE TABLE SeatType
(
    seat_type_Id CHAR(10) NOT NULL,
    seat_type_description VARCHAR(50) NOT NULL,
    PRIMARY KEY(seat_type_Id)
);
```

Figure 20-SQL Query for SeatType table

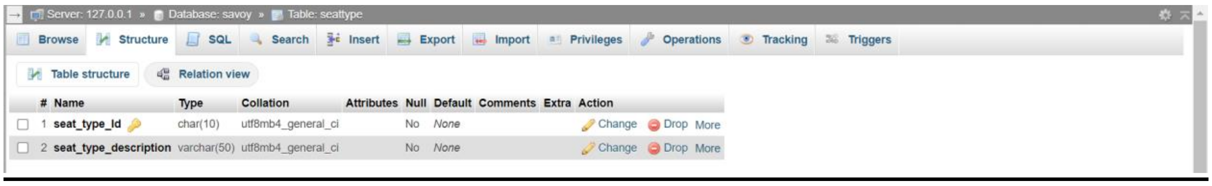
Query execution proof



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

CREATE TABLE SeatType ( seat_type_Id CHAR(10) NOT NULL, seat_type_description VARCHAR(50) NOT NULL, PRIMARY KEY(seat_type_Id) );
```

Figure 21-Query execution proof for SeatType table

Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	seat_type_Id	char(10)	utf8mb4_general_ci		No	None			Change Drop More
2	seat_type_description	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

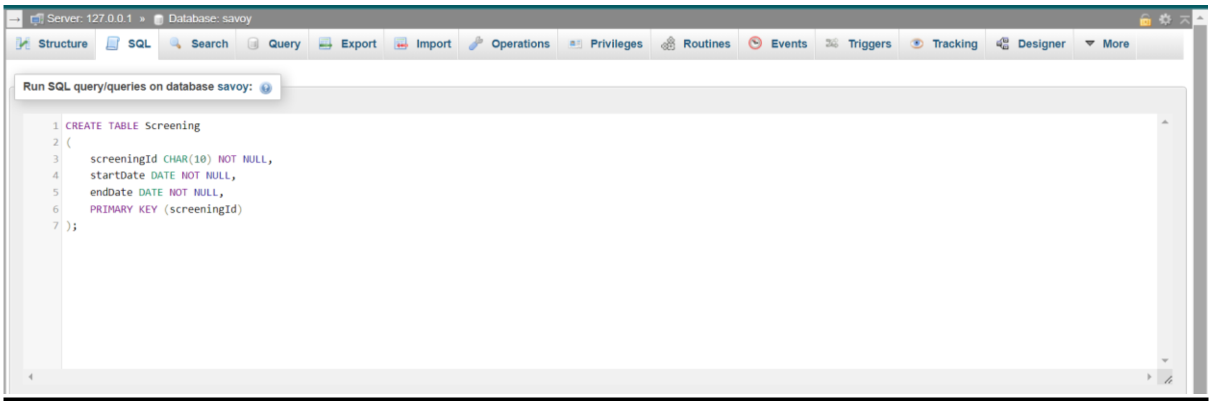
Figure 22-Table Structure of SeatType

Creating Screening Table

DDL Statement

```
CREATE TABLE Screening
(
	screeningId CHAR(10) NOT NULL,
	startDate DATE NOT NULL,
	endDate DATE NOT NULL,
	PRIMARY KEY (screeningId)
);
```

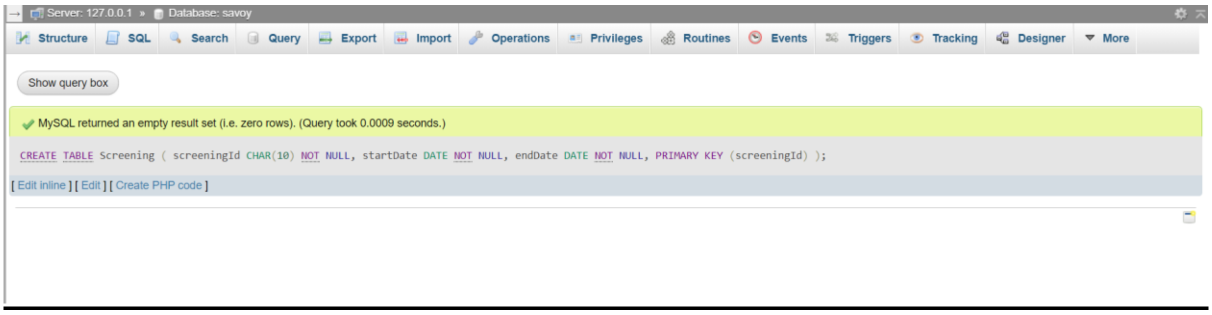
SQL Query



```
CREATE TABLE Screening
(
    screeningId CHAR(10) NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE NOT NULL,
    PRIMARY KEY (screeningId)
);
```

Figure 23-SQL Query for Screening table

Query Execution proof

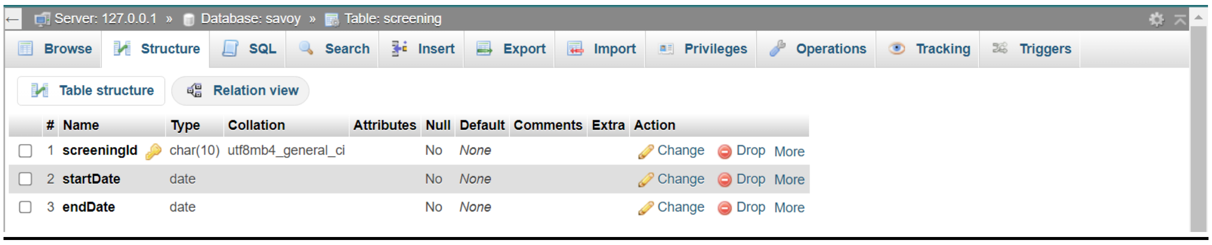


✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE Screening ( screeningId CHAR(10) NOT NULL, startDate DATE NOT NULL, endDate DATE NOT NULL, PRIMARY KEY (screeningId) );
```

Figure 24-Query execution proof for Screening table

Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	screeningId	char(10)	utf8mb4_general_ci		No	*None*			Change Drop More
2	startDate	date			No	*None*			Change Drop More
3	endDate	date			No	*None*			Change Drop More

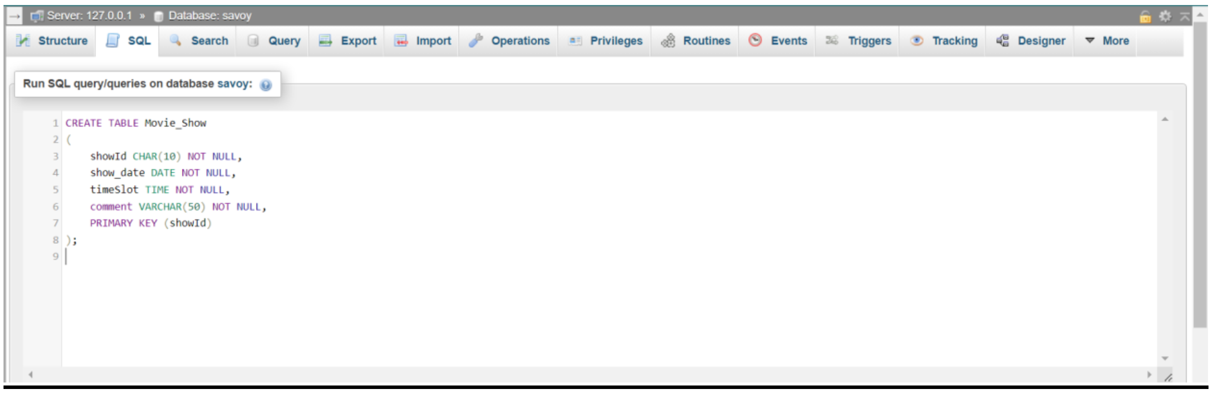
Figure 25-Table Structure of Screening

Creating Movie_Show Table

DDL Statement

```
CREATE TABLE Movie_Show
(
        showId CHAR(10) NOT NULL,
        show_date DATE NOT NULL,
        timeSlot TIME NOT NULL,
        comment VARCHAR(50) NOT NULL,

        PRIMARY KEY (showId)

);
```

SQL Query

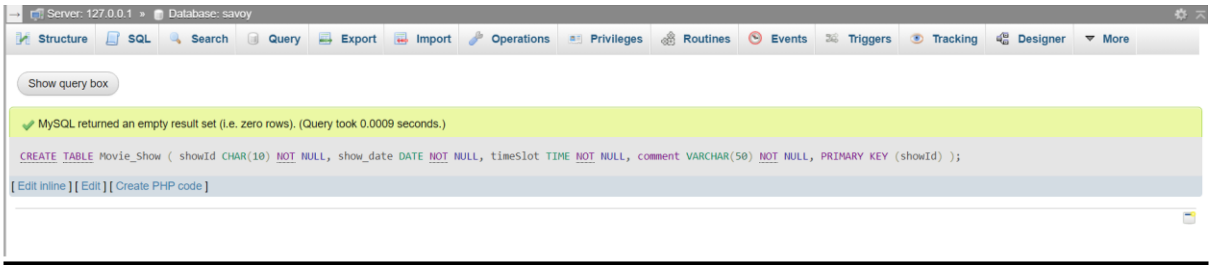


Run SQL query/queries on database savoy:

```
CREATE TABLE Movie_Show
(
    showId CHAR(10) NOT NULL,
    show_date DATE NOT NULL,
    timeSlot TIME NOT NULL,
    comment VARCHAR(50) NOT NULL,
    PRIMARY KEY (showId)
);
```

Figure 26-SQL Query for Movie_Show table

Query execution proof



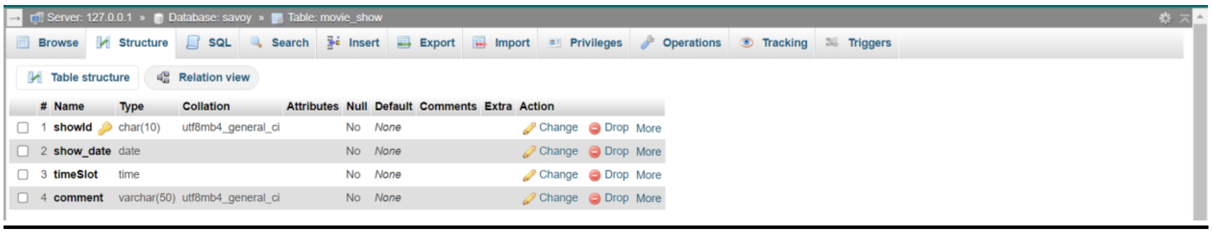
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE Movie_Show ( showId CHAR(10) NOT NULL, show_date DATE NOT NULL, timeSlot TIME NOT NULL, comment VARCHAR(50) NOT NULL, PRIMARY KEY (showId) );
```

[Edit inline] [Edit] [Create PHP code]

Figure 27-Query execution proof for Movie_Show table

Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	showId	char(10)	utf8mb4_general_ci		No	None			Change Drop More
2	show_date	date			No	None			Change Drop More
3	timeSlot	time			No	None			Change Drop More
4	comment	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

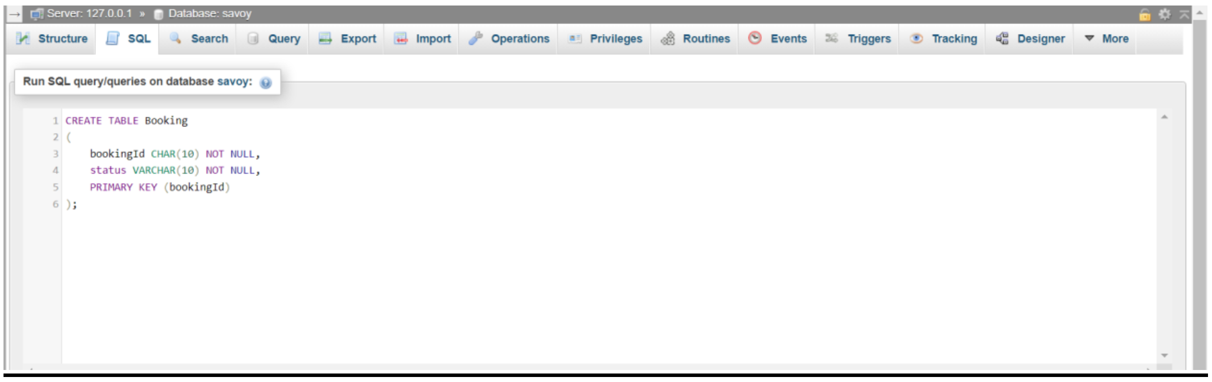
Figure 28-Table Structure of Movie_Show

Creating Booking Table

DDL Statements

```
CREATE TABLE Booking
(
	bookingId CHAR(10) NOT NULL,
	status VARCHAR(10) NOT NULL,
	PRIMARY KEY (bookingId)
);
```

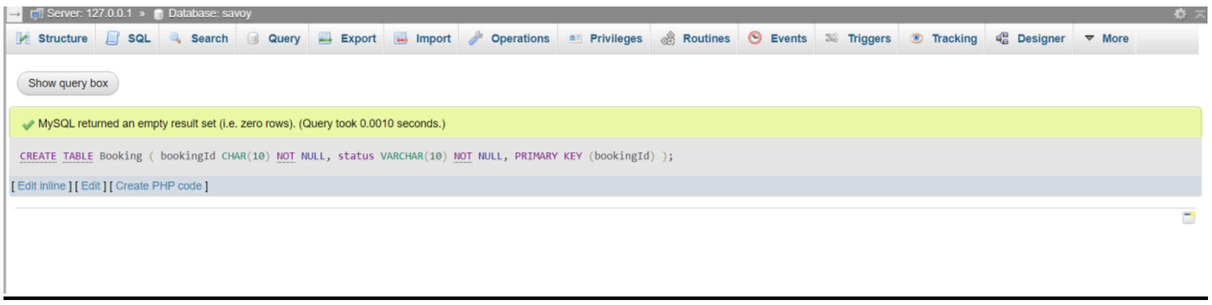
SQL Query



```
CREATE TABLE Booking
(
    bookingId CHAR(10) NOT NULL,
    status VARCHAR(10) NOT NULL,
    PRIMARY KEY (bookingId)
);
```

Figure 29-SQL Query for booking table

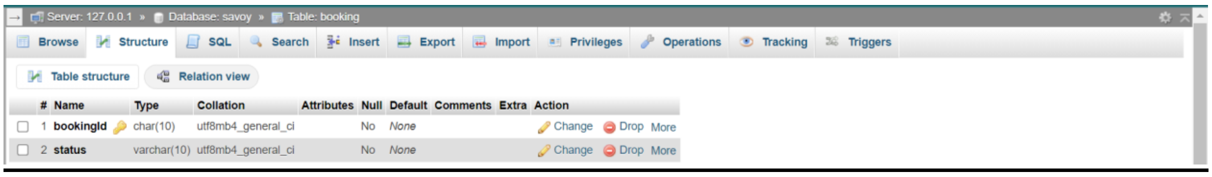
Query execution proof



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.)
CREATE TABLE Booking ( bookingId CHAR(10) NOT NULL, status VARCHAR(10) NOT NULL, PRIMARY KEY (bookingId) );
```

Figure 30-Query execution proof for Booking table

Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	bookingId	char(10)	utf8mb4_general_ci		No	None			Change Drop More
2	status	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More

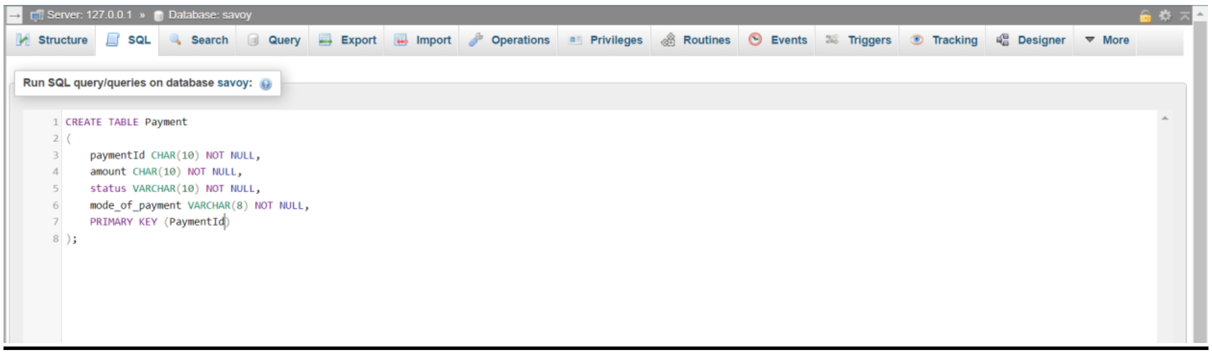
Figure 31-Table Structure of Booking

Creating Payment Table

DDL Statement

```
CREATE TABLE Payment
(
	paymentId CHAR(10) NOT NULL,
	amount CHAR(10) NOT NULL,
	status VARCHAR(10) NOT NULL,
	mode_of_payment VARCHAR(8) NOT NULL,
	PRIMARY KEY (Payment)
);
```

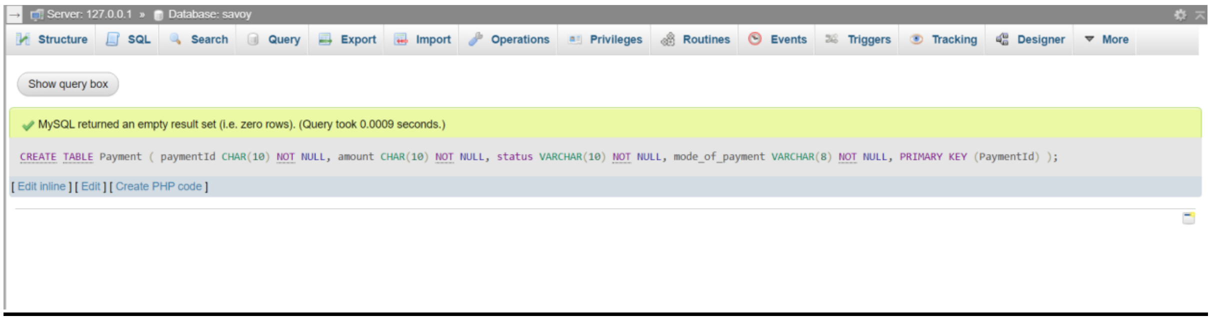
SQL Query



```
CREATE TABLE Payment
(
    paymentId CHAR(10) NOT NULL,
    amount CHAR(10) NOT NULL,
    status VARCHAR(10) NOT NULL,
    mode_of_payment VARCHAR(8) NOT NULL,
    PRIMARY KEY (PaymentId)
);
```

Figure 32-SQL Query for Payment table

Query execution proof


Server: 127.0.0.1 » Database: savoy
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More
Show query box

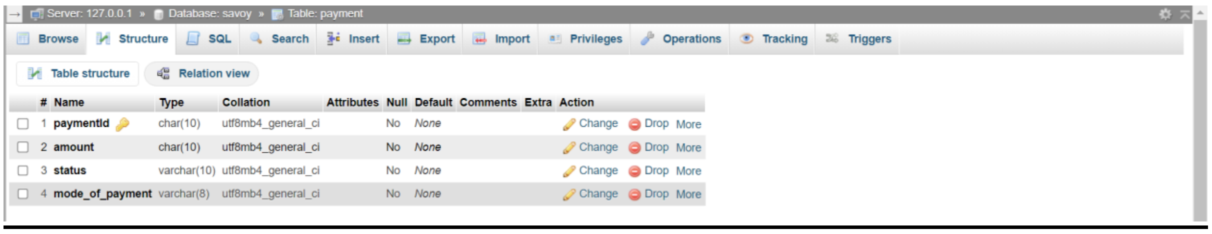
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE Payment ( paymentId CHAR(10) NOT NULL, amount CHAR(10) NOT NULL, status VARCHAR(10) NOT NULL, mode_of_payment VARCHAR(8) NOT NULL, PRIMARY KEY (PaymentId) );
```

[Edit inline] [Edit] [Create PHP code]

Figure 33-Query execution proof for Payment table

Table Structure


Server: 127.0.0.1 » Database: savoy » Table: payment
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	paymentId	char(10)	utf8mb4_general_ci		No	*None*			Change Drop More
2	amount	char(10)	utf8mb4_general_ci		No	*None*			Change Drop More
3	status	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
4	mode_of_payment	varchar(8)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 34-Table Structure of Payment