**Transport management**

A database management project

Abhiram k 2010080001

Rishitha d 2010080016

Sarvesh l 2010080018

Abhinav 2010080039

Under the guidance of DS Rao sir

INDEX

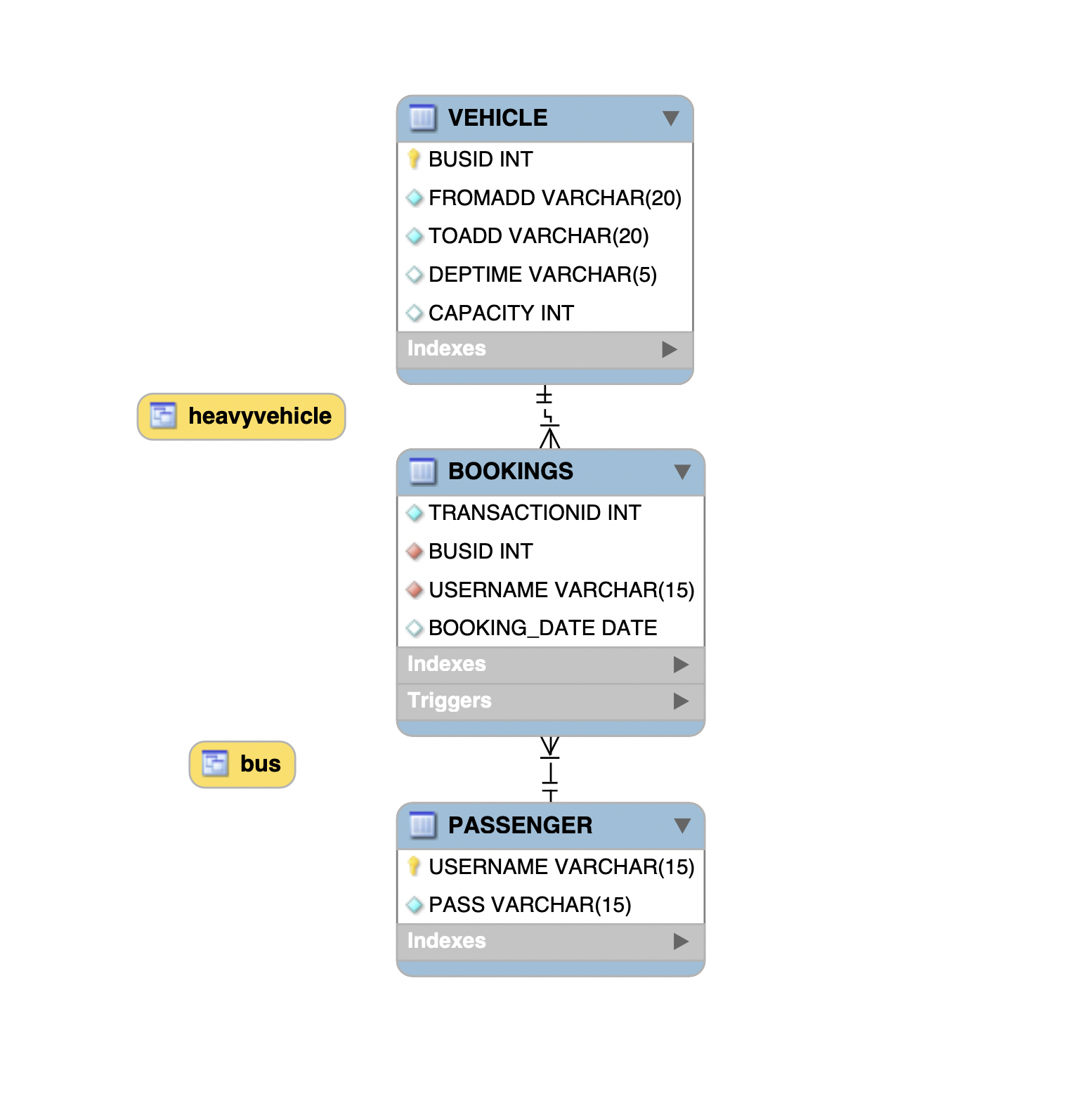
INTRODUCTION

The database our team has developed tries to tackle the problem of vehicle passenger management of a transport service company, where we provided the database with the passenger's account details, the vehicle details, and transaction details with separate tables for each property.

We use different database management concepts to achieve a good database to satisfy all needs and make it understandable to a layman. With proper table names and labels.

The application also has a special feature to view some important data directly with one click, and automatic checks to ensure the data entered is valid.

ER DIAGRAM



EER DIAGRAM

DDL COMMANDS

SYNTAX

Create database <database name>;

Create table <table name>(

<column name> <datatype> <notnull,primary key,other constrains>,

<column name> <datatype> <notnull,primary key,other constrains>);

DROP TABLE `<tablename>`.`<database>`

DROP TABLE `<database>’

CREATE DATABASE

CREATE DATABASE TRANSPORT;

USE TRANSPORT;

CREATE TABEL 1

CREATE TABLE PASSENGER(

USERNAME VARCHAR(15) NOT NULL PRIMARY KEY,

PASS VARCHAR(15) NOT NULL);

CREATE TABEL 2

CREATE TABLE VEHICLE(

BUSID INT NOT NULL PRIMARY KEY,

FROMADD VARCHAR(20) NOT NULL,

TOADD VARCHAR(20) NOT NULL,

DEPTIME VARCHAR(5),

CAPACITY INT DEFAULT 30);

CREATE TABEL 3

CREATE TABLE BOOKINGS(

TRANSACTIONID INT NOT NULL,

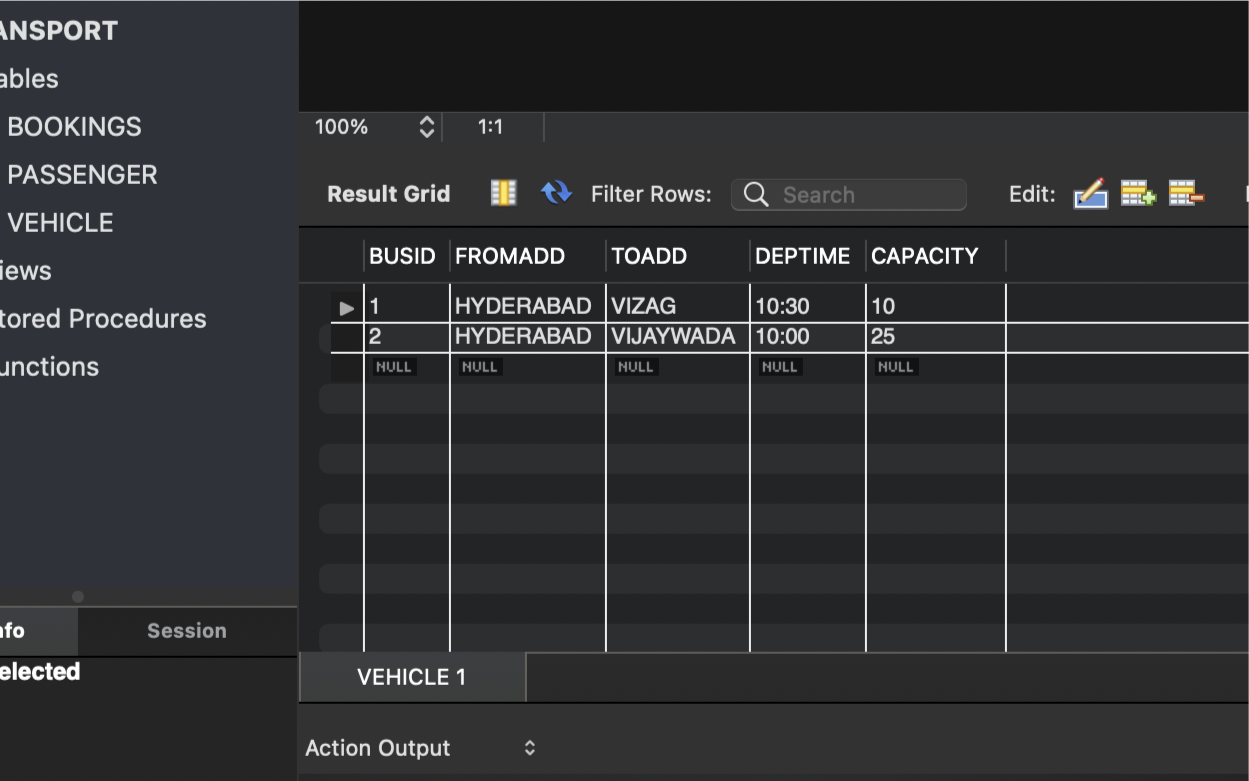
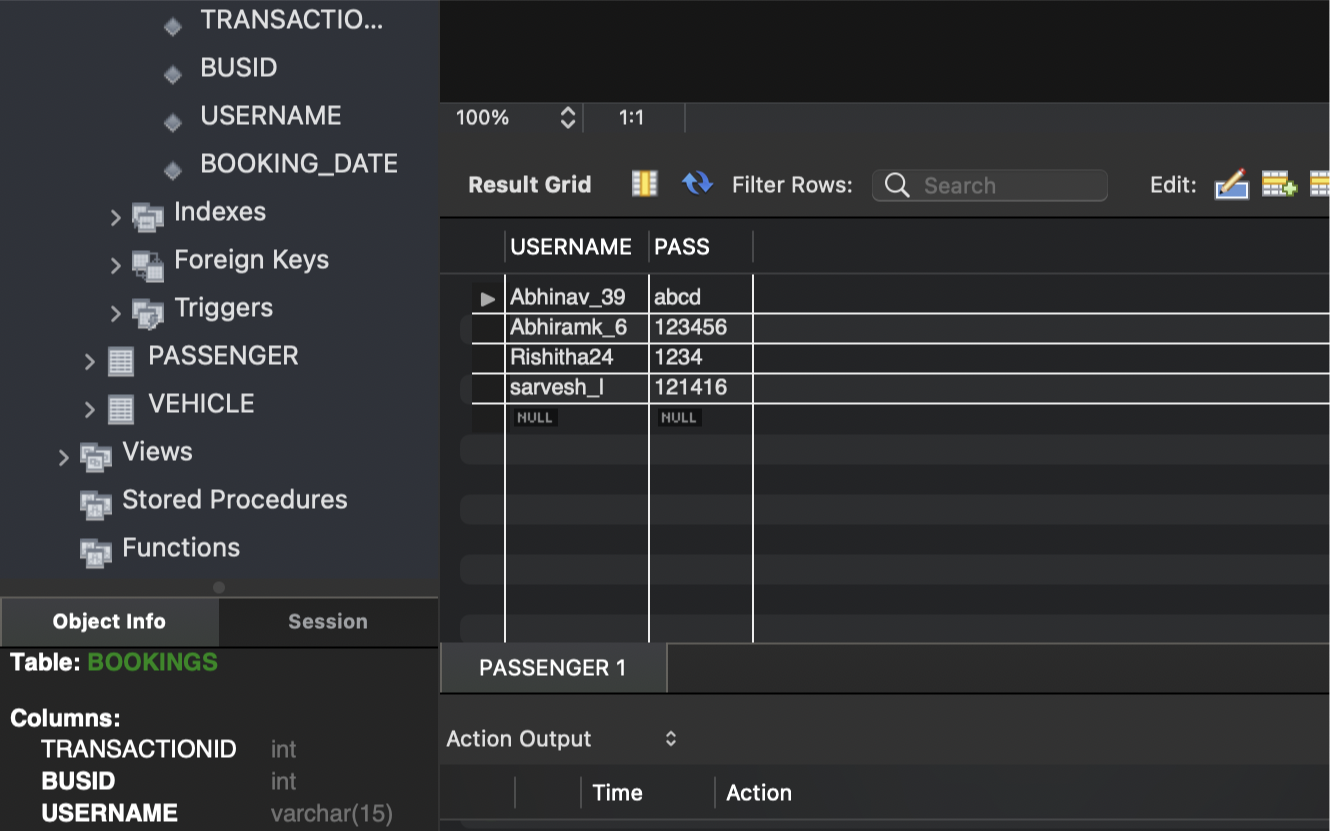
BUSID INT NOT NULL,

USERNAME VARCHAR(15) NOT NULL,

FOREIGN KEY (BUSID) REFERENCES VEHICLE(BUSID),

FOREIGN KEY (USERNAME) REFERENCES PASSENGER(USERNAME) );

OUTPUTS

DML COMMANDS

SYNTAX

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, …);

DELETE FROM table\_name WHERE condition;

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

INSERT

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1001,01,"Abhiramk\_6","10-17-2021");

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1002,02,"Rishitha24","10-17-2021");

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1003,03,"Abhinav\_39","2022-04-16");

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1004,04,"sarvesh\_l","2022-04-16");

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1005,05,"Abhinav\_39","2022-04-16");

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1006,005,"Abhiramk\_6","2022-04-16");

INSERT INTO BOOKINGS(TRANSACTIONID,BUSID,USERNAME,BOOKING\_DATE) VALUES(1007,02,”Rishitha24","2022-04-16");

INSERT INTO PASSENGER(USERNAME,PASS)VALUES("Abhiramk\_6","123456");

INSERT INTO PASSENGER(USERNAME,PASS)VALUES("Rishitha24","1234");

INSERT INTO PASSENGER(USERNAME,PASS)VALUES("Abhinav\_39","abcd");

INSERT INTO PASSENGER(USERNAME,PASS)VALUES("sarvesh\_l","121416");

INSERT INTO VEHICLE(BUSID,FROMADD,TOADD,DEPTIME,CAPACITY)VALUES(02,”HYDERABAD","VIJAYWADA","10:00",25);

DELETE

DELETE FROM VEHICLE WHERE BUSID=02;

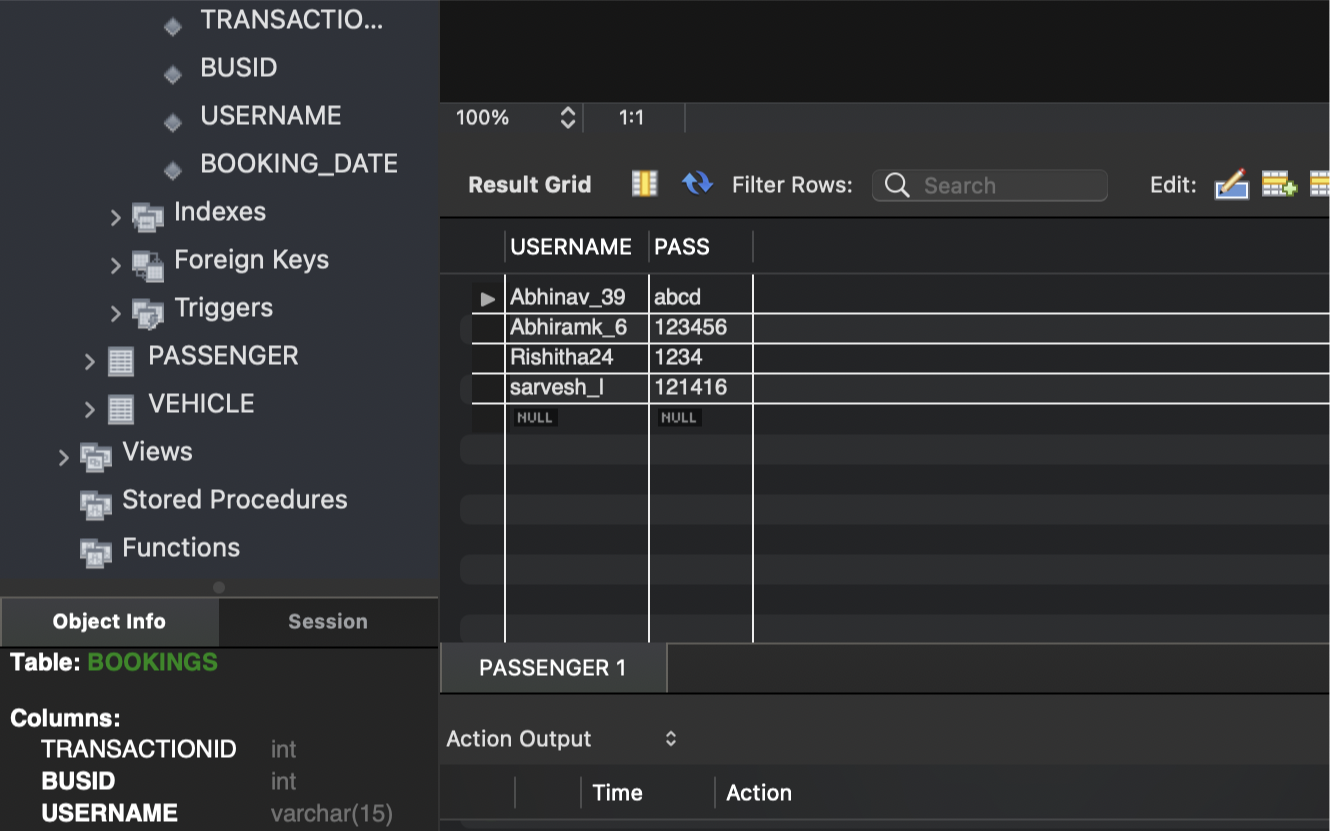
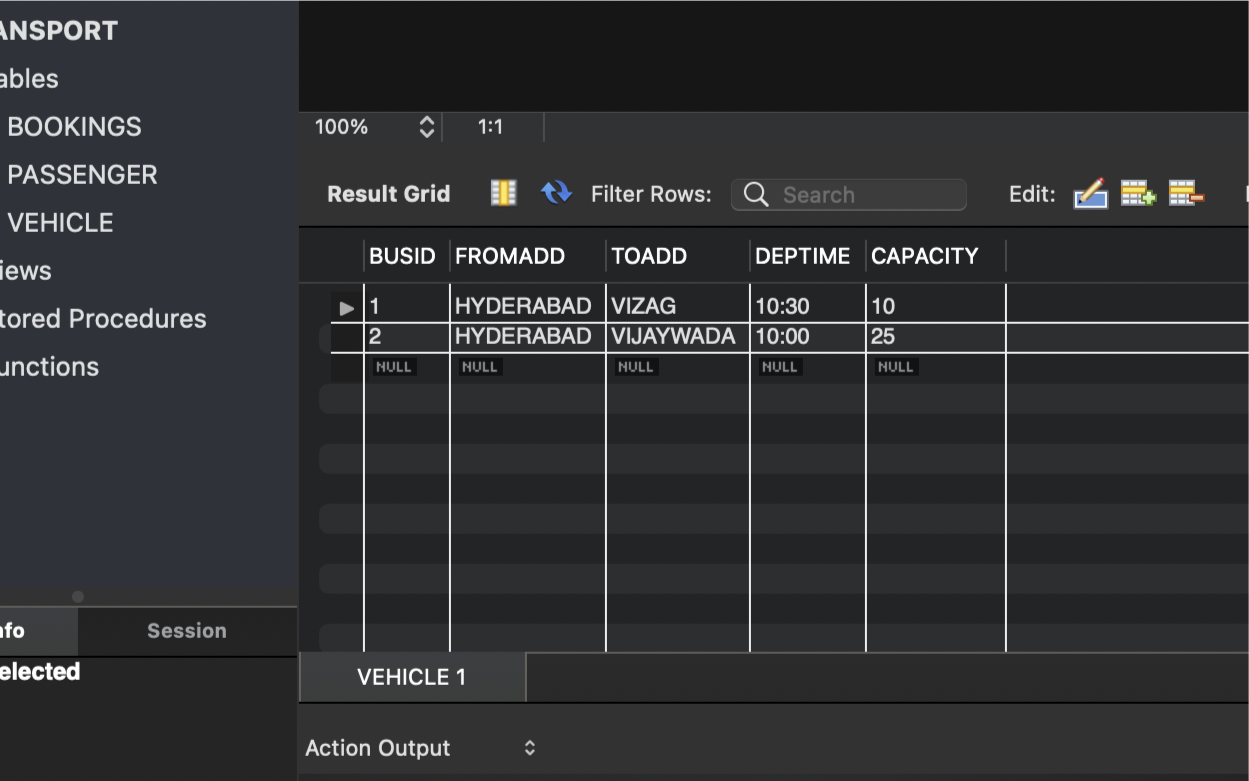
UPDATE

UPDATE PASSENGER

SET PASS=“ABHI”

WHERE USERNAME=“Abhiramk\_6”

OUTPUT



SUB QUERIES

SYNTAX

SELECT column\_name [, column\_name ]

FROM table1 [, table2 ]

WHERE column\_name OPERATOR

(SELECT column\_name [, column\_name ]

FROM table1 [, table2 ]

[WHERE])

INSERT INTO table\_name [ (column1 [, column2 ]) ]

SELECT [ \*|column1 [, column2 ]

FROM table1 [, table2 ]

[ WHERE VALUE OPERATOR ]

DELETE FROM TABLE\_NAME

[ WHERE OPERATOR [ VALUE ]

(SELECT COLUMN\_NAME

FROM TABLE\_NAME)

[ WHERE) ]

QUERIES

SELECT BUSID,FROMADD,TOADD

FROM VEHICLE

WHERE CAPACITY>20;

SELECT USERNAME

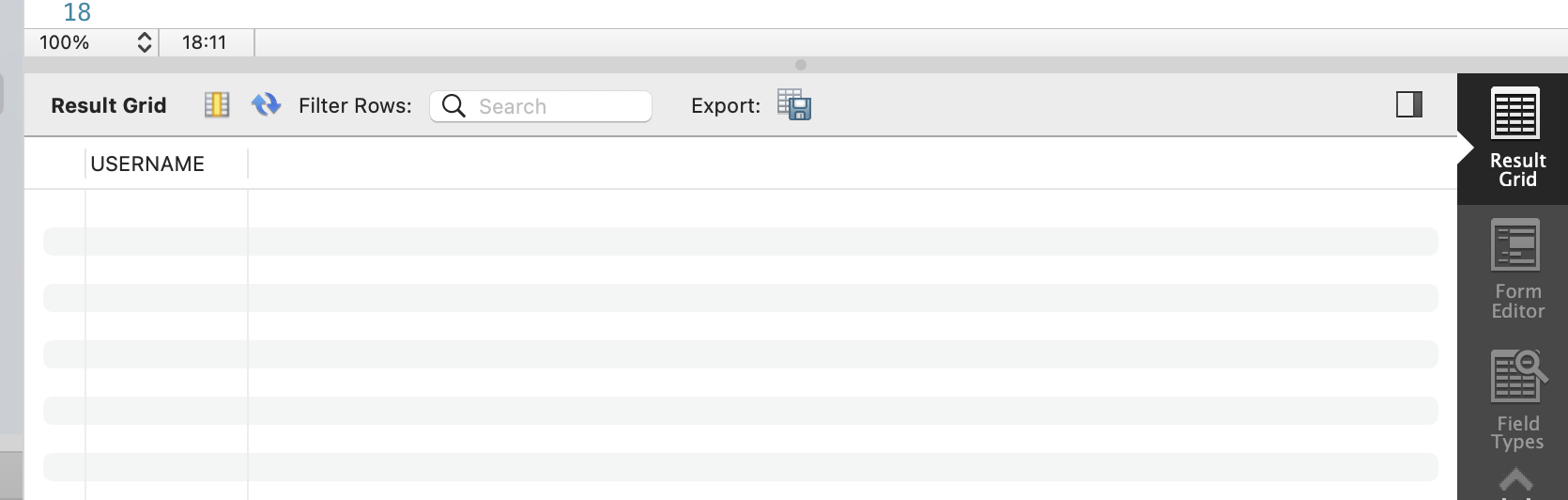
FROM BOOKINGS

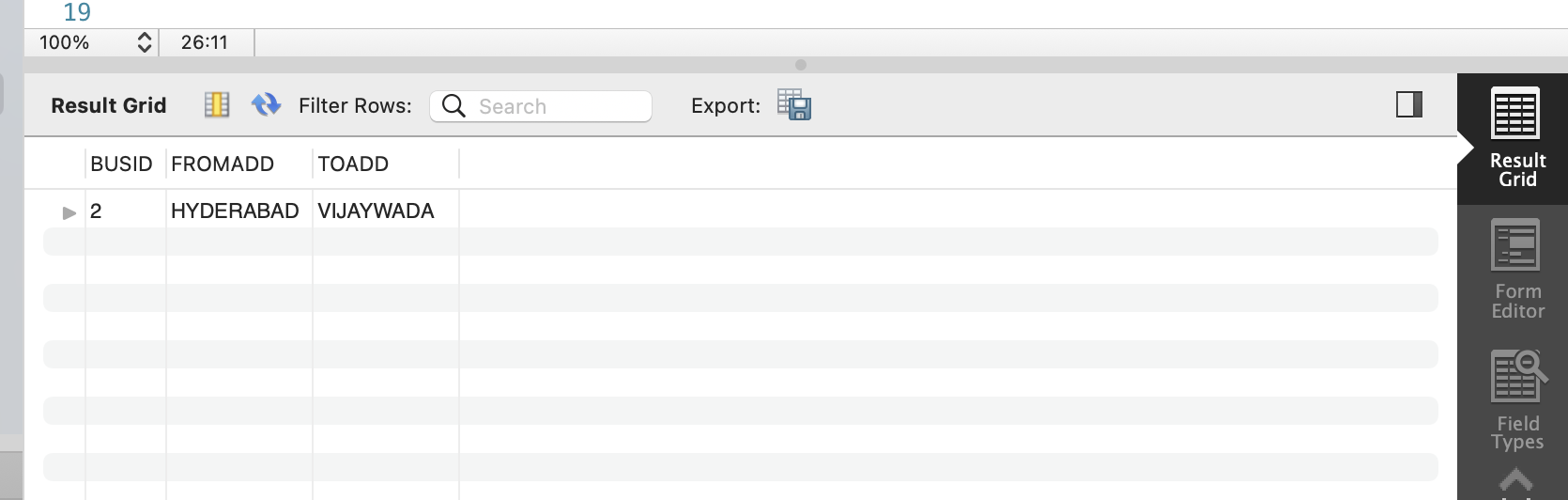
WHERE BUSID=1;

SELECT BUSID,FROMADD,TOADD

FROM VEHICLE

WHERE CAPACITY>10;

OUTPUT



JOINS

SYNTAX

SELECT column\_name(s)

FROM table1

LEFT JOIN *table2*

ON table1.column\_name = table2.column\_name;

SELECT column\_name(s)

FROM table1

FULL OUTER JOIN *table2*

ON table1.column\_name = table2.column\_name

WHERE condition;

SELECT column\_name(s)

FROM table1 T1, table1 T2

WHERE condition;

QUERIES

SELECT USERNAME

FROM PASSENGER, BOOKING

WHERE USERNAME = “Abhiramk\_6”

SQL COMMANDS

BASIC SQL SYNTAX

SELECT column1, column2....columnN

FROM table\_name;

SELECT column1, column2....columnN

FROM table\_name

WHERE CONDITION;

SELECT column1, column2....columnN

FROM table\_name

WHERE CONDITION-1 {AND|OR} CONDITION-2;

SELECT column1, column2....columnN

FROM table\_name

WHERE column\_name BETWEEN val-1 AND val-2;

SELECT COUNT(column\_name)

FROM table\_name

WHERE CONDITION;

SELECT SUM(column\_name)

FROM table\_name

WHERE CONDITION

GROUP BY column\_name

HAVING (arithmetic function condition);

ALTER TABLE table\_name {ADD|DROP|MODIFY} column\_name {data\_ype};

QUERIES

ALTER TABLE BOOKINGS

ADD BOOKING\_DATE DATE AFTER USERNAME;

CREATE VIEW BUS2 AS

SELECT USERNAME

FROM BOOKINGS

WHERE BUSID=01;

select\* from HEAVY VEHICLE;

CREATE VIEW HEAVYVEHICLE AS

SELECT BUSID,FROMADD,TOADD

FROM VEHICLE

WHERE CAPACITY>20;

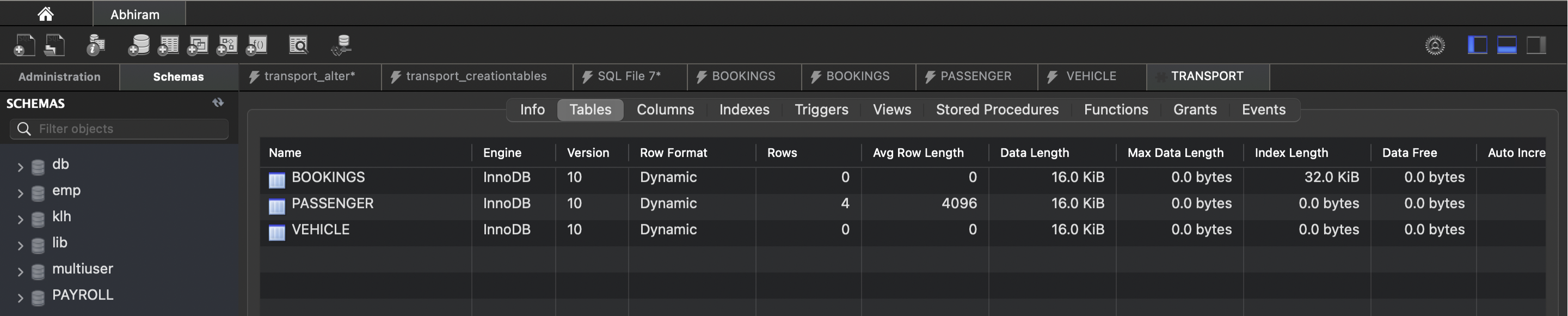
SELECT LOWER(FROMADD)

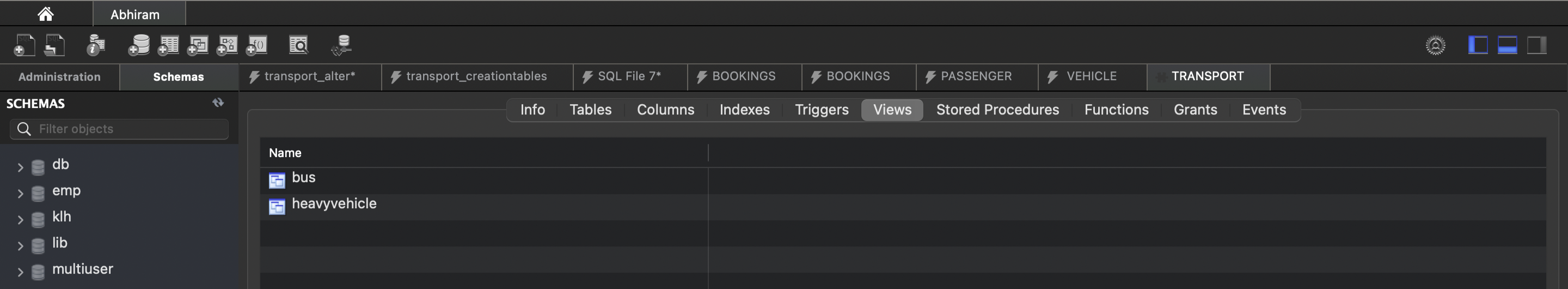
FROM VEHICLE;

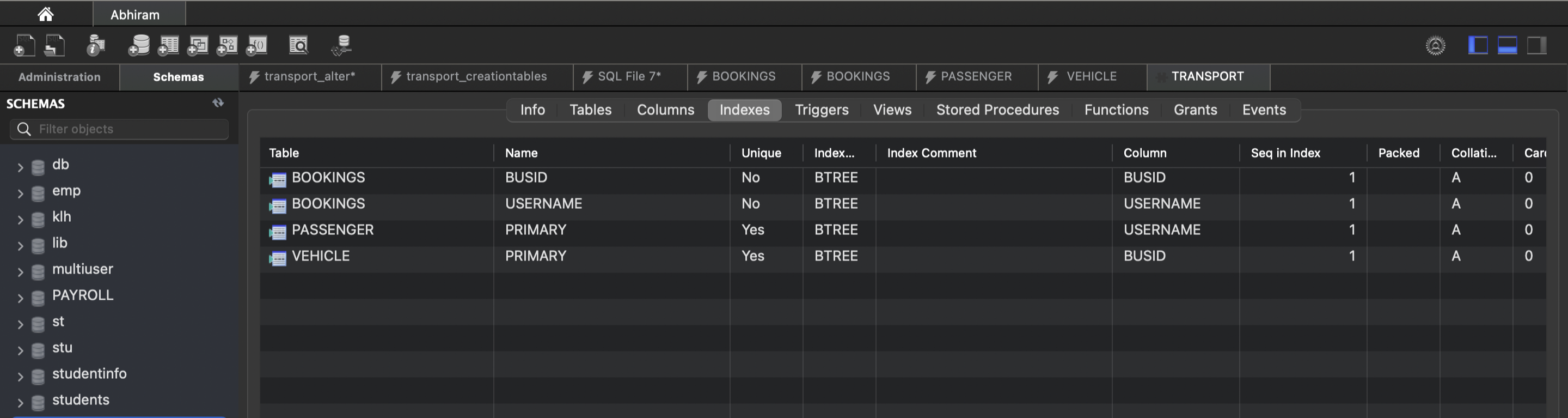
SELECT LOWER (TOADD)

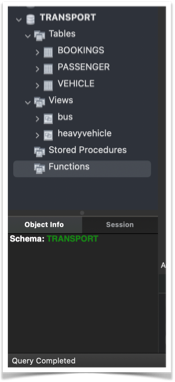
FROM VEHICLE;

OVERVIEW OF DATABASE









TRIGGERS

SYNTAX

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]

[trigger\_body]

QUERY

CREATE TRIGGER BOOKING\_TIME

after insert

ON BOOKINGS

FOR EACH ROW

insert into BOOKING\_DATE values(curdate());

CREATE TRIGGER BUSID

after insert

ON VEHICLE

FOR EACH ROW

insert into BUSID values(BUSID+1);

CURSOR

SYNTAX

**DECLARE** cursor\_name **CURSOR** **FOR**

**Select** statement;

**Select** statement;

**FETCH** [ **NEXT** [ **FROM** ] ] cursor\_name **INTO** variable\_list;

**Close** cursor\_name;

DECLARE @name VARCHAR(50) -- database name

DECLARE @path VARCHAR(256) -- path for backup files

DECLARE @fileName VARCHAR(256) -- filename for backup

DECLARE @fileDate VARCHAR(20) -- used for file name

SET @path = 'C:\Backup\'

SELECT @fileDate = CONVERT(VARCHAR(20),GETDATE(),112)

DECLARE db\_cursor CURSOR FOR

SELECT name

FROM MASTER.dbo.sysdatabases

WHERE name NOT IN ('master','model','msdb','tempdb')

OPEN db\_cursor

FETCH NEXT FROM db\_cursor INTO @name

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @fileName = @path + @name + '\_' + @fileDate + '.BAK'

BACKUP DATABASE @name TO DISK = @fileName

FETCH NEXT FROM db\_cursor INTO @name

END

CLOSE db\_cursor

DEALLOCATE db\_cursor

A cursor in SQL Server is a database object that allows us to retrieve each row at a time and manipulate its data. A cursor is nothing more than a pointer to a row. It's always used in conjunction with a SELECT statement. It is usually a collection of SQL.

PROCEDURES

SYNTAX

CREATE PROCEDURE *procedure\_name*

AS

sql\_statement

GO;

EXEC procedure\_name;

QUERY

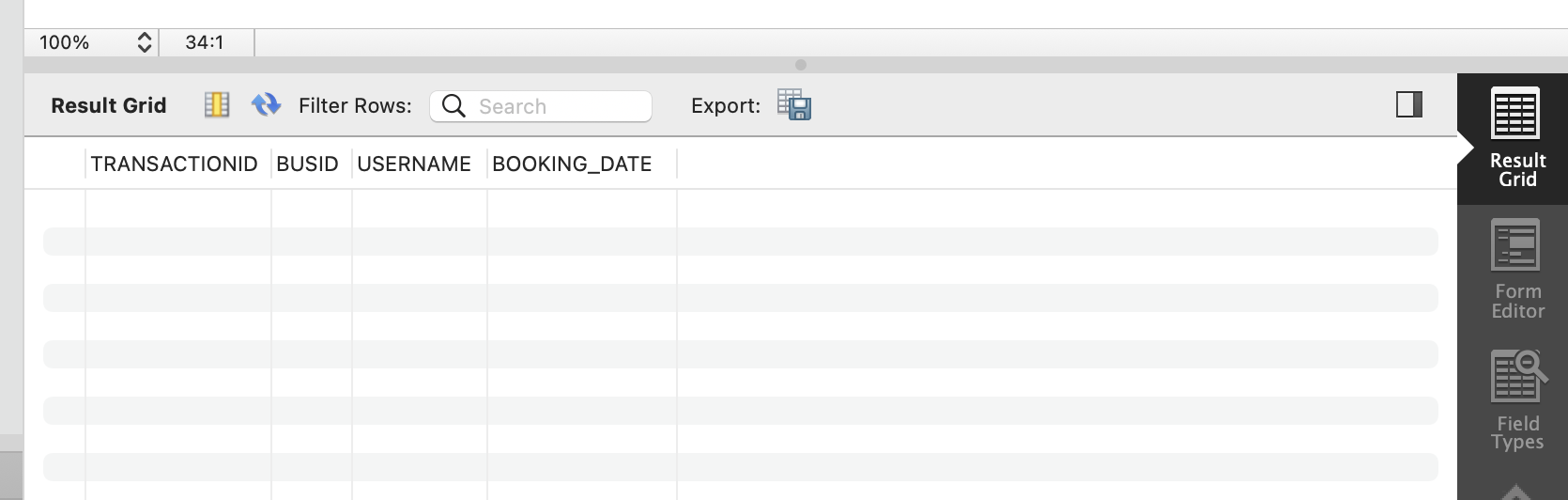
CREATE PROCEDURE SelectAllUSERS

AS

SELECT \* FROM PASSENGER

GO;

EXEC SelectAllUSERS;

OUTPUT