

Information filled by student:

Course basic information

Code	Course Name	Credit Hours		
		Lecture	Practice	Total
<u>(Is 212)</u>	<u>database</u>	<u>2</u>	<u>1</u>	<u>3</u>

Research Title

(Car Rental System)

Student Name: ibrahim shreif ahmed

Student ID:1618120170100196

Level: 2nd Level

Department : General

Table of contents

1. System Description
2. Data Dictionaries
3. Entity Relationship Diagram
4. Different Functions
5. Sub Query
6. Count and Group Functions
7. Different Joins
8. Insert Statement
9. Update Statement
10. Delete statement
11. References
12. Git-Hub Repository Link.

System Description

Background

In this research paper I will work in data base system to car Rental

Goal

The objective of this master's project is to create a database to centrally handle the information of all the car rental in the car agency , and to provide access to this information with an easy to use web-based interface that can be accessed by any device with basic html rendering capabilities.

Requirements

Requirements for the system fall into three categories, those tending towards the usability of the system, those towards the maintenance and alteration of the system, and those towards the security of the system. For the first requirement, accessibility was addressed by making the system accessible from the web via a standard web browser, and no required extensions, such as java, javascript, or flash. The system was also designed so that the users would be able to complete the repeatable tasks in a streamlined manner to cut down on wasted time, and in a concise way to switch between tasks. To address the maintenance of the system, a modular design was used. This was done so that bugs can easily be found and additional features can easily be added to the system. To address the security of the system, users are required to run sessions over Hypertext Transfer Protocol over Secure Socket Layer, https.

Functional Description

The car rental system is an online platform that serves as a tool for the car owners to post their vehicle for hire or rent, this will also serve as market ground for customers who are looking to rent a vehicle.

The purpose of this article is to provide a guide on how to plan and prepare the database model of a car or vehicle rental system.

Data Dictionaries

Column	Type	null
Admin_id	Int	Not
Name	Varchar (11)	Not
Contact	Varchar (50)	Not
Address	Varchar (100)	Not
Username	Varchar (30)	Not

Password	Varchar (30)	Not
-----------------	--------------	-----

tblcar (car_id, car_name, description, car_model_year, car_brand, color, capacity, plate_number, rate, owner_id, status)

Description: The owner of the car is the one who encodes the details into the system. Car information and details are encoded/recorded in the tblcar database table; it includes the year model of the car, the brand, color, capacity and plate number. The tblcar is linked or connected to the tblowner table through its foreign key (owner_id).

Column	Type	null
Car_id	Int (11)	Not
Car_name	Varchar(30)	Not
Description	Varchar(100)	Not
Car_model_year	Int (4)	Not
Car_brand	Varchar(25)	Not
Color	Varchar(15)	Not
Capacity	Int (3)	Not
Plate_number	Varchar(11)	Not
Rate	float	Not
Owner_id	Int (11)	Not
Status	Int (2)	Not

tblcarimage (image_id, image_description, car_id)

Description: the car owner can upload multiple images of the car and it will be stored in tblcarimage database table. The table includes three entities; image_id is the primary key, image_description is

the name and details of the image and the car_id (foreign key) which serves as the link between the tblcar and tblcarimage.

Column	Type	Null
Imge_id	Int(11)	Not
imge_discription	Varchar(50)	Not
Car_id	Int(11)	Not

tblcarreview (review_id, review, review_score, date, customer_id)

Description: the customers of the system can rate and give feedback to the car rented through the review and comment section of the system. The table consists of five entities; the primary key of the table (review_id), the comment, the score (1-5, five as the highest), the date, and the customer_id which is the foreign key and the primary key of the tblcustomer table.

Column	Type	Null
review_id	int(11)	NOT
Review	varchar(100)	NOT
review_score	int(1)	NOT
Date	date	NOT
customer_id	int(11)	NOT
car_id	int(11)	NOT

tblcustomer (customer_id, customer_name, address, contact, profile_image, fb_account, username, password, admin_id, account_status)

Description: customers of the system can register into the platform and it will be verified by the admin once the customer submits all the credentials needed. The table includes the name, address, contact, image, the facebook account, username and password. The account_status field serves as the flag as to whether the account is active or deactivated.

Column	Type	Null
customer_id	int(11)	Not
customer_name	varchar(50)	Not
Address	varchar(100)	Not
Contact	varchar(11)	Not
profile_image	blob	Not
fb_account	varchar(50)	Not
Username	varchar(30)	Not
Password	varchar(30)	Not
admin_id	int(11)	Not
account_status	int(1)	Not

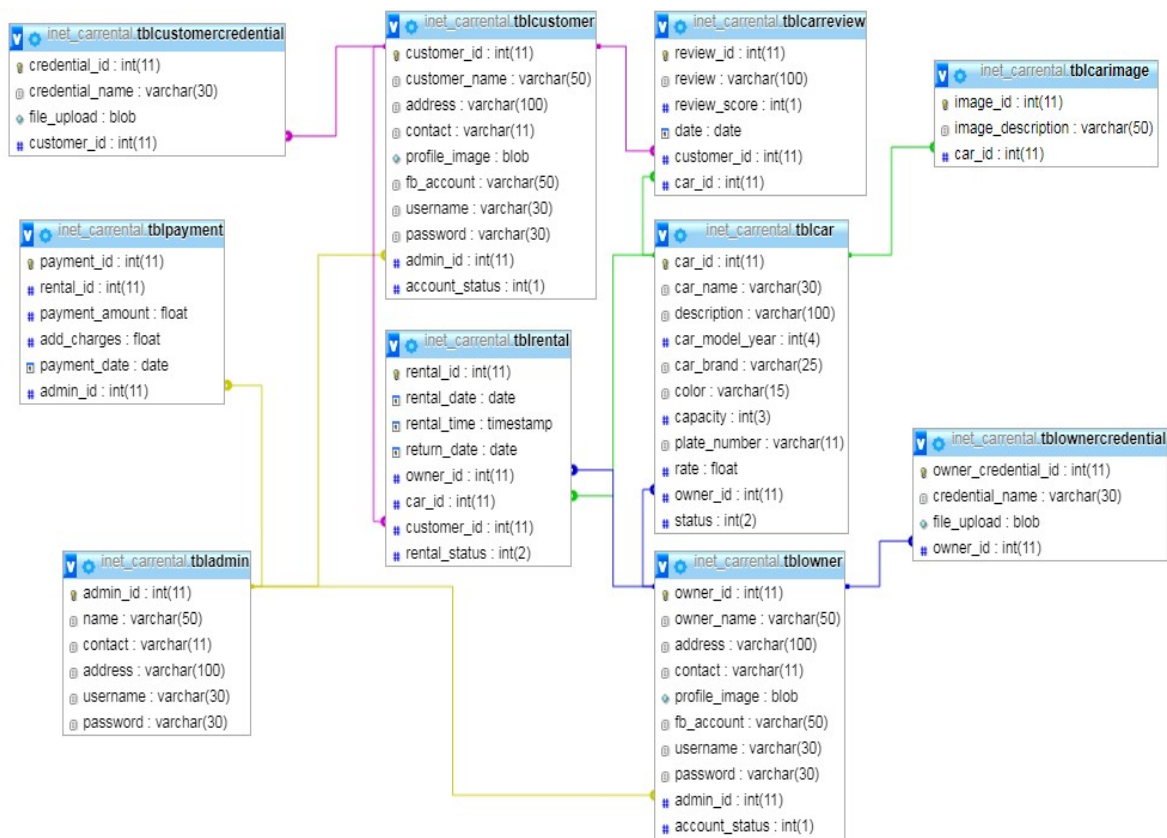
tblcustomercredential (credential_id, credential_name, file_upload, customer_id)

Description: the customers will need first to upload the necessary documents to verify its identity. The credentials uploaded by the customers will be stored in the tblcustomercredential.

Column	Type	null
Credential_id	Int (11)	Not
Credential_name	Varchar (30)	Not
File_upioad	Blob	Not
Customer_id	Int (11)	Not

Description: there are two ways in order for the car owners to be part of the platform; (1) is to register the information through the registration module of the system or (2) fill-up the hardcopy form and submit it to the administrators for encoding.

Entity Relationship Diagram



Different Functions

SELECT column1, column2, ...
FROM table_name
WHERE condition;

SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;



```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;  
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

Sub Query

```
UPDATE table SET column_name = new_value  
[ WHERE OPERATOR [ VALUE ]
```



```
(SELECT COLUMN_NAME  
FROM TABLE_NAME)  
[ WHERE ]
```

```
DELETE FROM TABLE_NAME  
[ WHERE OPERATOR [ VALUE ]  
(SELECT COLUMN_NAME  
FROM TABLE_NAME)  
[ WHERE ]
```

```
INSERT INTO table_name [ (column1 [, column2 ]) ]  
SELECT [ *|column1 [, column2 ]  
FROM table1 [, table2 ]  
[ WHERE VALUE OPERATOR ];
```

Count and Group Functions

```
1SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country;  
  
2SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

Different Joins

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;  
  
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2  
ON table1.column_name = table2.column_name;
```



```
SELECT column_name(s)
FROM table1
RIGHT 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;
```

Insert Statement

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

```
INSERT INTO Customers (CustomerName, City, Country)
VALUES ('Cardinal', 'Stavanger', 'Norway');
```

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Update Statement

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```



UPDATE Customers

SET ContactName='Juan'

WHERE Country='Mexico';

UPDATE Customers

SET ContactName='Juan'



DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

DELETE FROM Customers;

DELETE FROM CAR;



References

1. w3schools
2. Beynon-Davies, Paul (2003). *Database Systems (3rd ed.)*. Palgrave Macmillan. ISBN 978-1403916013.
3. Egyptian knowledge bank