

CAR RENTAL MANAGEMENT

SYSTEM

Module code : B9IS100

Module Leader : Dr Shazia A Afzal

By Akshaya Chhaban Tonde(student id : 10604976) Muhammet Enes Ilhan (Student id : 10607947) Kunal Nagesh Kini (Student id : 10601299)

Contents

[Background of Development of the project 2](#_bookmark0)

[Scope 2](#_bookmark1)

[Business Rules 8](#_bookmark2)

[Relational Schema in 3NF 10](#_bookmark3)

[XML in schema 14](#_bookmark4)

[Data Diagram 15](#_bookmark5)

[Entity Relationship Diagram 16](#_bookmark6)

[Implementation in SQL server 16](#_bookmark7)

[Tables 16](#_bookmark8)

[Stored procedures 16](#_bookmark9)

[Triggers 20](#_bookmark10)

[Views 22](#_bookmark11)

[Conclusion 24](#_bookmark12)

[Bibliography 24](#_bookmark13)

[Appendix A: Create Table queries 25](#_bookmark14)

[Appendix B: Insert into queries 27](#_bookmark15)

[Innovation 29](#_bookmark16)

[Individual Contribution 29](#_bookmark17)

Background of Development of the project Car Rental Management system -

Car Rental management is specializing in renting cars to customers. It is an online system through which customers can view available cars, register, view profile and book car. Car Rental management is not only about managing the vehicles, but also about managing the people. Car Rental Management integrates all functions for management of tasks of car rental agency and its employees. The system has all the information needed for their management. It can plan the entire day of your employees in accordance with reservations.

Scope

This project traverses a lot of areas ranging from business concept to computing field and required to perform several research to be able to achieve the project objectives. The area covers include

Car rental industry: This includes study on how the car rental business is being done, process involved and opportunity that exist for improvement.

Technology used for the development of the application. General customers as well as the

company’s staff will be able to use the system effectively.

Web-platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal.

1.Car

|  |  |  |
| --- | --- | --- |
| Attribute Name | Data Type | Description |
| car\_id | INT(10) | Car Id used to identify every  car uniquely |
| car\_name | VARCHAR(50) | Car's name |
| car\_engine\_no | VARCHAR(25) | Car's engine number |
| car\_chassis\_no | VARCHAR(25) | Car's chassis number |
| car\_model\_no | VARCHAR(25) | Car's model number |
| car\_price | INT(10) | Car's renting price |
| car\_fuel\_type | VARCHAR(10) | Car's fuel  type(Gasoline/Petrol/Diesel) |
| car\_mileage | INT(10) | Car's mileage(per KM) |
| car\_fuel\_tank\_capacity | INT(10) | Car's fuel tank's capacity |
| car\_seating\_capacity | INT(5) | Car's seating capacity(4/5/6  etc.) |
| car\_registration\_date | VARCHAR(20) | Car's registration date |

|  |  |  |
| --- | --- | --- |
| car\_rating | INT(5) | Car's user rating |
| car\_policy | TEXT | Car's policy |
| created\_at | TIMESTAMP | Date on which car details  added to DB |

Primary Key: car\_id

Required Attrivutes: car\_id, car\_name, car\_engine\_no, car\_chassis\_no, car\_model\_no, car\_price, car\_fuel\_type, car\_seating\_capacity, car\_registration\_date, car\_policy, created\_at

Optional Attributes: car\_mileage, car\_fuel\_tank\_capacity, car\_rating

1. Client

|  |  |  |
| --- | --- | --- |
| Attribute Name | Data type | Description |
| client\_id | INT(10) | Client Id used to identify  user uniquely |
| client\_first\_name | VARCHAR(50) | Client's first name |
| client\_last\_name | VARCHAR(50) | Client's last name |
| client\_adress | TEXT | Client's address |
| client\_mobile\_number | VARCHAR(20) | Client's mobile number |
| client\_role | VARCHAR(10) | Client's role  (User/Admin/SuperAdmin) |
| client\_dob | VARCHAR(15) | Client's date of birth |
| client\_email | VARCHAR(25) | Client's email |
| client\_password | VARCHAR(20) | Client's password |
| client\_licecense | VARCHAR(20) | Client's license number |
| created\_at | TIMESTAMP | Date on which user  registered in DB |

Primary Key: client\_id

Required Attrivutes: client\_first\_name, client\_mobile\_number, client\_dob, client\_email, client\_password, client\_license, client\_address, created\_at

Optional Attributes: client\_last\_name

1. Car Type

|  |  |  |
| --- | --- | --- |
| Attribute Name | Data Type | Description |

|  |  |  |
| --- | --- | --- |
| car\_type\_id | INT(10) | Car type Id identify every  car's feature |
| car\_id | INT(10) | Car Id used to connect with  Car Table |
| car\_automated | INT | Auto pilot mode support  (Yes/No) |
| car\_power\_steering | INT | Is car having power steering  (Yes/No) |
| car\_air\_conditioner | INT | Is air conditioner available  in car (Yes/No) |
| car\_passenger\_airbag | INT | Is passenger airbag available  in car (Yes/No) |
| car\_driver\_airbag | INT | Is driver airbag available in  car (Yes/No) |
| car\_sun\_roof | INT | Is any sunroof available in  car (Yes/No) |
| car\_driver\_seat | INT(5) | Driver seat is on left or right  side |
| car\_fog\_lights | INT | Is car having the fog lights  (Yes/No) |
| car\_type | VARCHAR(20) | Car type(SUV, Sports etc.) |

Primary Key: car\_type\_id Foreign Key: car\_id

Required Attrivutes: car\_automated, car\_power\_steering, car\_air\_conditioner, car\_passenger\_airbag, car\_driver\_airbag, car\_driver\_seat, car\_type, ccreated\_at

Optional Attributes: car\_sun\_roof, car\_fog\_lights

1. Payment

|  |  |  |
| --- | --- | --- |
| Attribute Name | Data type | Description |
| Payment\_id | INT(10) | Payment Id describe every  payment uniquely |
| Booking\_id | INT(10) | Booking Id will provide  booking details |
| Payment\_mode | VARCHAR(20) | Payment Mode (wallet, CC,  DC, Cash) |
| Transaction\_amount | INT(20) | Total amount of booked car |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Promo\_code | VARCHAR(20) | Promo code for special  discount |
| Add\_charges | VARCHAR(10) | Addition charges (taxes) |
| Transaction\_status | VARCHAR(10) | Transaction status of the  payment |
| Payment\_currency | VARCHAR(20) | In which currency you  paying the amount |
| created\_at | TIMESTAMP | Created Date of Payment |
| updated\_at | TIMESTAMP | Updated Date of Payment |

Note: - CC => Credit Card, DC => Debit Card. Primary Key: - payment\_id

Foreign Key: - booking\_id

Required Attributes: - payment\_mode, transaction\_amount, payment\_currency, add\_charges

Optional Attributes: - promo\_code, special\_discount, created\_at, updated\_at

5)Rental Type table

rental\_type\_id, rental\_trip\_selection, rental\_seaters, rental\_fuel\_type, rental\_car\_type

|  |  |  |
| --- | --- | --- |
| Attribute Name | Data Type | Description |
| retal\_type\_id | INT(10) | uniquely identify every  rental details |
| rental\_trip\_selection | VARCHAR(20) | Rental Trip Selection used  to determine is it Roundtrip or Oneway Booking |
| rental\_seaters | INT(10) | Rental Seaters provide details of total number of  people will seat in car |
| rental\_fuel\_type | VARCHAR(10) | Rental Fuel Type describe  about fuel type of car.+6 |

|  |  |  |
| --- | --- | --- |
| rental\_car\_type | VARCHAR(20) | Rental Car Type define car’s  type |
| rental\_auto\_pilot | INT | Rental Car Auto Pilot option  for auto driving option |
| rental\_status | VARCHAR(10) | Check the particular car  available at selected date |
| rental\_start\_date | VARCHAR(20) | Start date When Client want  to Start Journey |
| rental\_end\_date | VARCHAR(20) | End date When Client want  to End Journey |

1. Booking

|  |  |  |
| --- | --- | --- |
| Attribute Name | Data Type | Description |
| booking\_id | INT(10) | Booking Id used to identify  every unique booking |
| car\_id | INT(10) | Car Id used to show car  details |
| client\_id | INT(10) | Client Id used to show client  details |
| rental\_type\_id | INT(10) | Rental Type Id provide  rental details |
| booking\_start\_location | VARCHAR(25) | Booking’s Start Location |
| booking\_end\_location | VARCHAR(25) | Booking’s end Location |
| booking\_duration | INT(5) | Booking’s duration |
| booking\_trip\_amount | INT(20) | Total Amount chargeable of  booking |
| car\_km\_start | INT(20) | Car’s Kilometres at the start  of Booking |
| car\_km\_end | INT(20) | Car’s Kilometres at the end  of Booking |
| car\_fuel\_start | INT(10) | Car’s Fuel Percentage at the  start of booking |

|  |  |  |
| --- | --- | --- |
| car\_fuel\_end | INT(10) | Car’s Fuel Percentage at the  end of booking |
| booking\_status | VARCHAR(20) | Booking’s status will define  booking |
| created\_at | TIMESTAMP | Booking’s creation date |
| updated\_at | TIMESTAMP | Booking’s updating date |

Primary Key: - booking\_id

Foreign Key: - car\_id, client\_id, rental\_id

Required Attributes: - booking\_start\_location, booking\_end\_location, booking\_duration, booking\_trip\_amount, created\_at, updated\_at, booking\_status

Optional Attributes: - car\_km\_start, car\_km\_end, car\_fuel\_start, car\_fuel\_end

1. Refund table

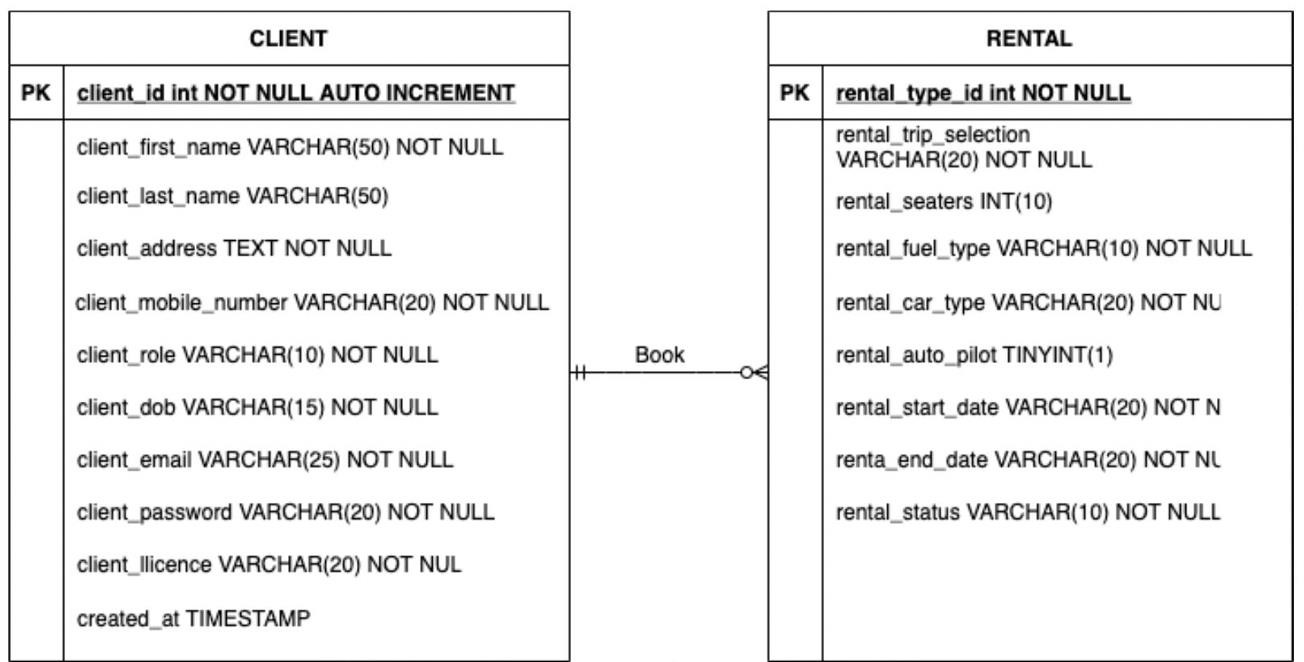
|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data Type** | **Description** |
| refunds refund\_id | INT(10) | Refund Id used to identify  every unique refunds |
| booking\_id | INT(10) | Booking\_id is referred to  the booking for which it is refunded |
| deduction\_amount | INT(10) | Deduct any amount from  the transaction amount |
| refund\_status | VARCHAR(20) | Refund Status(Initiation,  Processing, etc.) |
| refund\_amount | INT(20) | Total Refund amount |
| refund\_account | VARCHAR(20) | The account in which  amount will be refunded |
| created\_at | TIMESTAMP | Refund’s Initiation date |

Primary Key: - refund\_id Foreign Key: - booking\_id

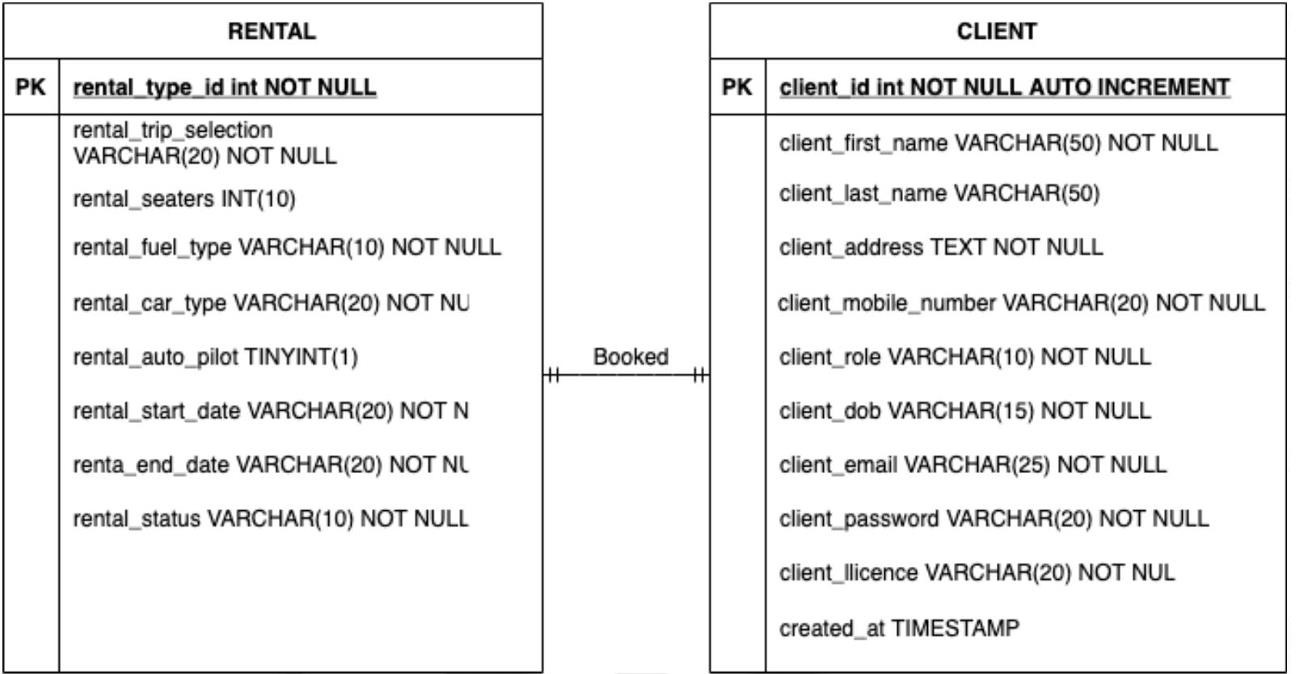
Required Attributes: - refund\_status, refund\_amount, refund\_account, created\_at Optional Attributes: - deduction\_amount

Business Rules

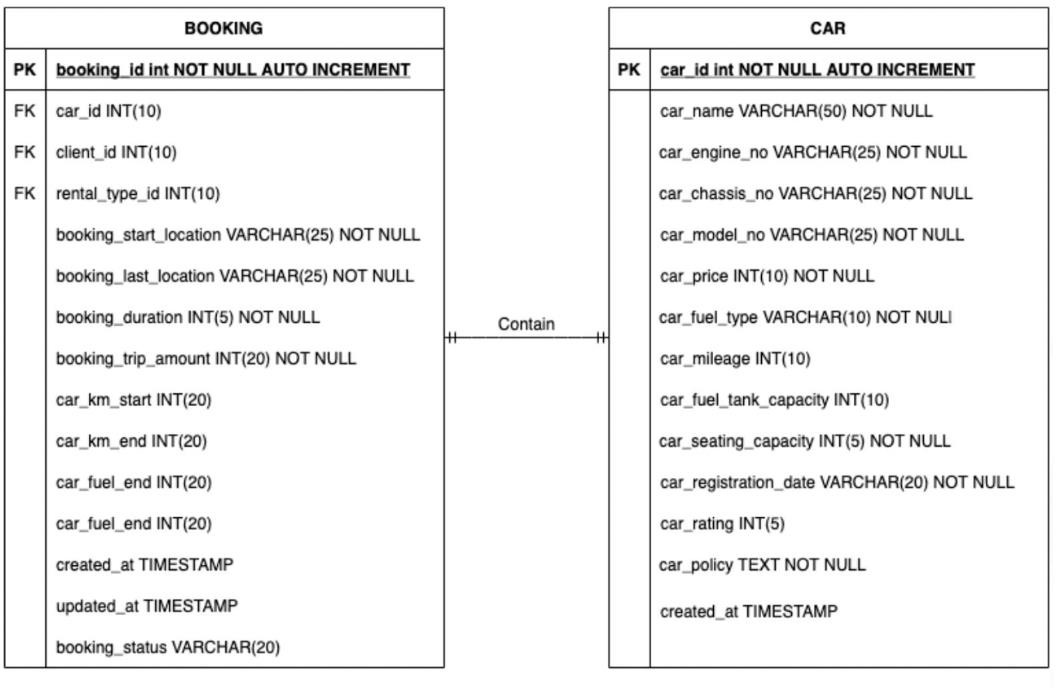
1. One client can book many rentals



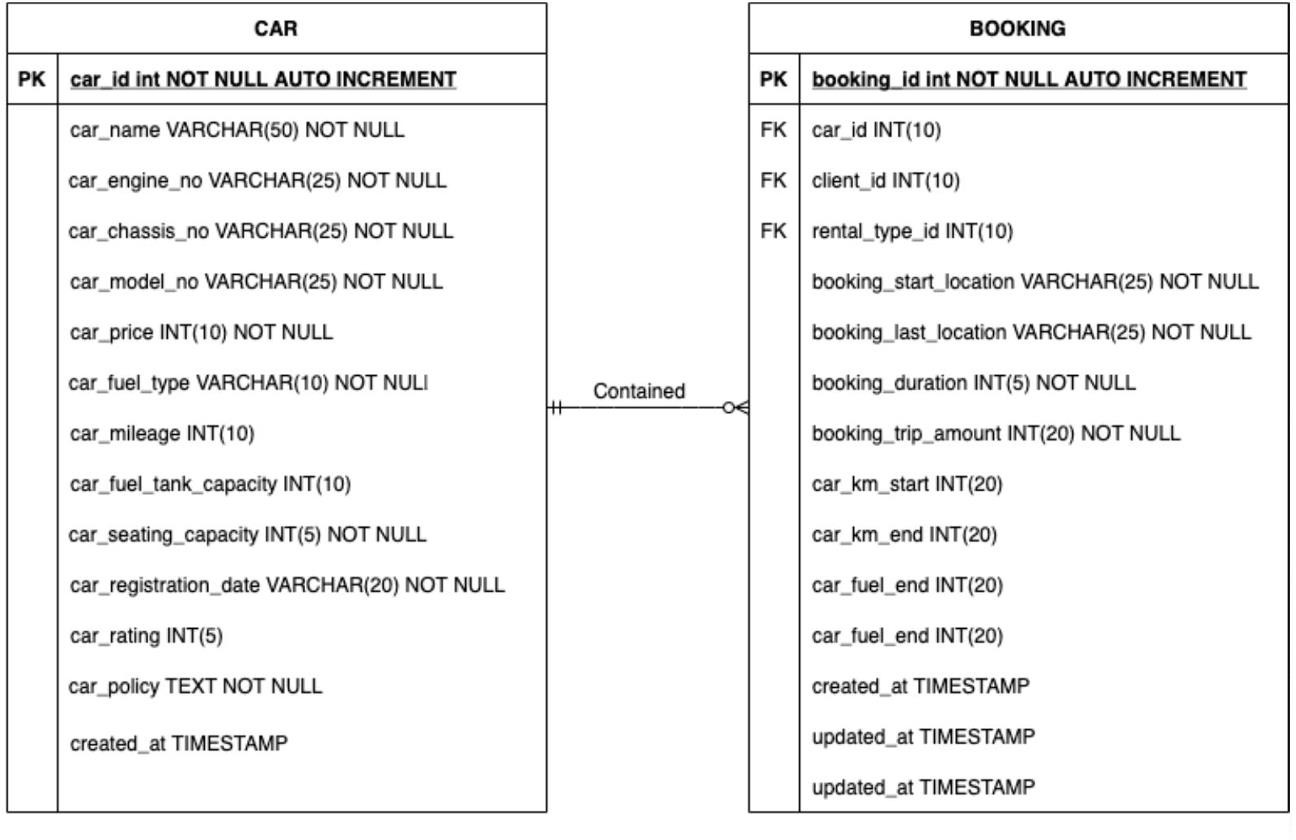
1. One rental can be booked by one client



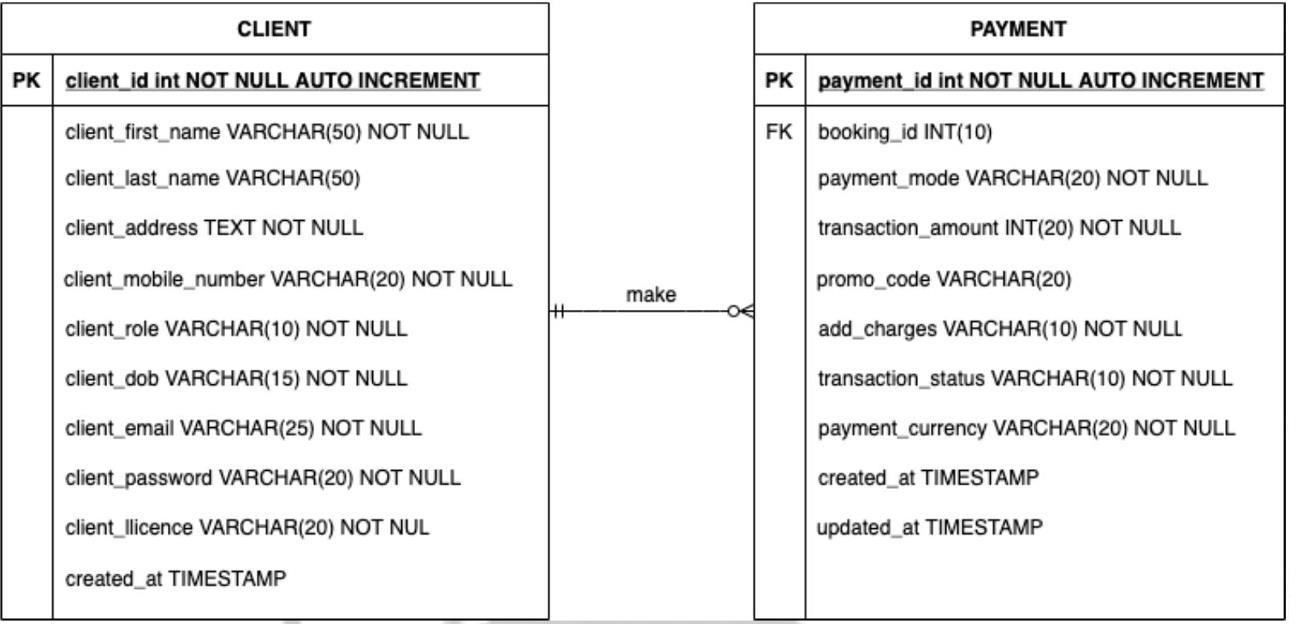
1. One booking can contain one car



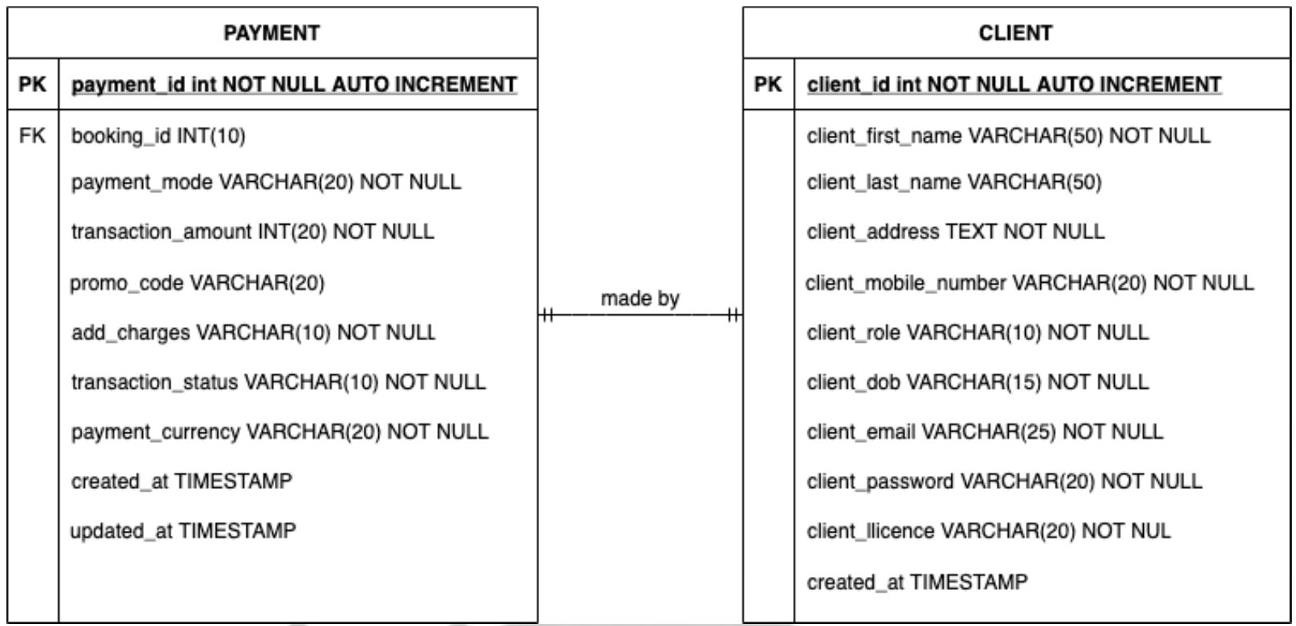
1. One car can be contained in many bookings



1. One client can make many payment



1. One payment can be made by one client



Relational Schema in 3NF

Normalization is a technique to organize data in a database. Normalization is used to lessen the redundancy from a relation or set of relations. It also helps to counteract the inconvenient characteristics like Insertion, Update, and Deletion Anomalies. It divides the larger table into the smaller table and links them using relationship.

Here are the most used normal forms:

1. First Normal Form (1NF) The entity/table is in First Normal Form only if:
   1. It cannot hold the multiple values.
   2. All the values must be atomic
2. Second Normal Form (2NF) The entity/table is in the Second Normal Form only if:
   1. It must be in the 1NF.
   2. All non-key attributes must be dependent on primary key. (No partial dependencies)
3. Third Normal Form (3NF) The entity/table is in the Third Normal Form if and only if:
   1. It must be in the 2NF.
   2. It should not contain any transitive dependency.

# Car Table

* + - 1. Primary Key is defined(car\_id).
      2. Every column has an atomic value.
      3. No repeating groups.

Hence, the table is in the First Normal Form.

1. The table is in the First Normal Form
2. Every non-key attribute is dependent on the primary key. Hence, the table is in the Second Normal Form.
3. The table is in the Second Normal Form.
4. It does not contain any transitive dependency. Hence The table is in the Third Normal Form.

# Client Table

* + - 1. Primary Key is defined(client\_id).
      2. Every column has an atomic value.
      3. No repeating groups. Hence,

The table is in the First Normal Form.

1. The table is in the First Normal Form
2. Every non-key attribute is dependent on the primary key. Hence, the table is in the Second Normal Form.
3. The table is in the Second Normal Form.
4. It does not contain any transitive dependency. Hence The table is in the Third Normal Form.

# Car Type Table

* + - 1. Primary Key is defined(car\_type\_id).
      2. Every column has an atomic value.
      3. No repeating groups.

Hence, the table is in the First Normal Form.

c. The table is in the First Normal Form

d. Every non-key attribute is dependent on the primary key. Hence, the table is in the Second Normal Form.

c. The table is in the Second Normal Form.

d. It does not contain any transitive dependency. Hence The table is in the Third Normal Form.

# Payment Table

* + - 1. Primary Key is defined(payment\_id).
      2. Every column has an atomic value.
      3. No repeating groups.

Hence, the table is in the First Normal Form.

c. The table is in the First Normal Form

d. Every non-key attribute is dependent on the primary key. Hence, the table is in the Second Normal Form.

c. The table is in the Second Normal Form.

d. It does not contain any transitive dependency. Hence The table is in the Third Normal Form

# Rental Type Table

* + - 1. Primary Key is defined(rental\_type\_id).
      2. Every column has an atomic value.
      3. No repeating groups.

Hence, The table is in the First Normal Form.

* + - 1. The table is in the First Normal Form

f. Every non-key attribute is dependent on the primary key. Hence, The table is in the Second Normal Form.

e. The table is in the Second Normal Form.

f. It does not contain any transitive dependency. Hence The table is in the Third Normal Form.

# Booking Table

* + - 1. Primary Key is defined(booking\_id).
      2. Every column has an atomic value.
      3. No repeating groups.
      4. Hence, The table is in the First Normal Form.
      5. The table is in the First Normal Form
      6. Every non-key attributes is dependent on the primary key. Hence, The table is in the Second Normal Form.

e. The table is in the Second Normal Form.

f. It does not contain any transitive dependency. Hence The table is in the Third Normal Form

# Refund Table

* + - 1. Primary Key is defined(refund\_id).
      2. Every column has an atomic value.
      3. No repeating groups.

Hence, The table is in the First Normal Form.

* + - 1. The table is in the First Normal Form

h. Every non-key attribute is dependent on the primary key. Hence, The table is in the Second Normal Form.

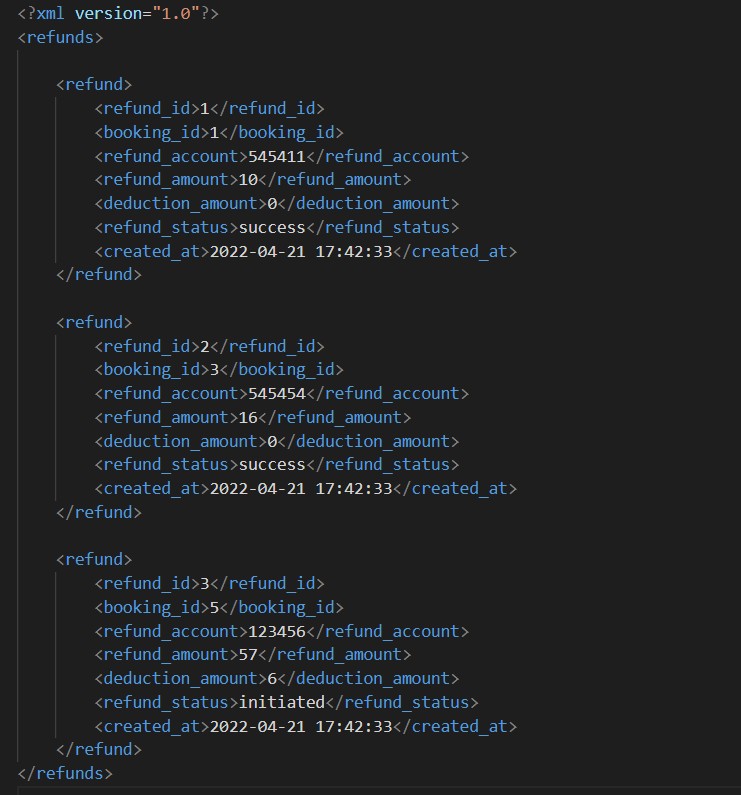
e. The table is in the Second Normal Form.

f. It does not contain any transitive dependency.

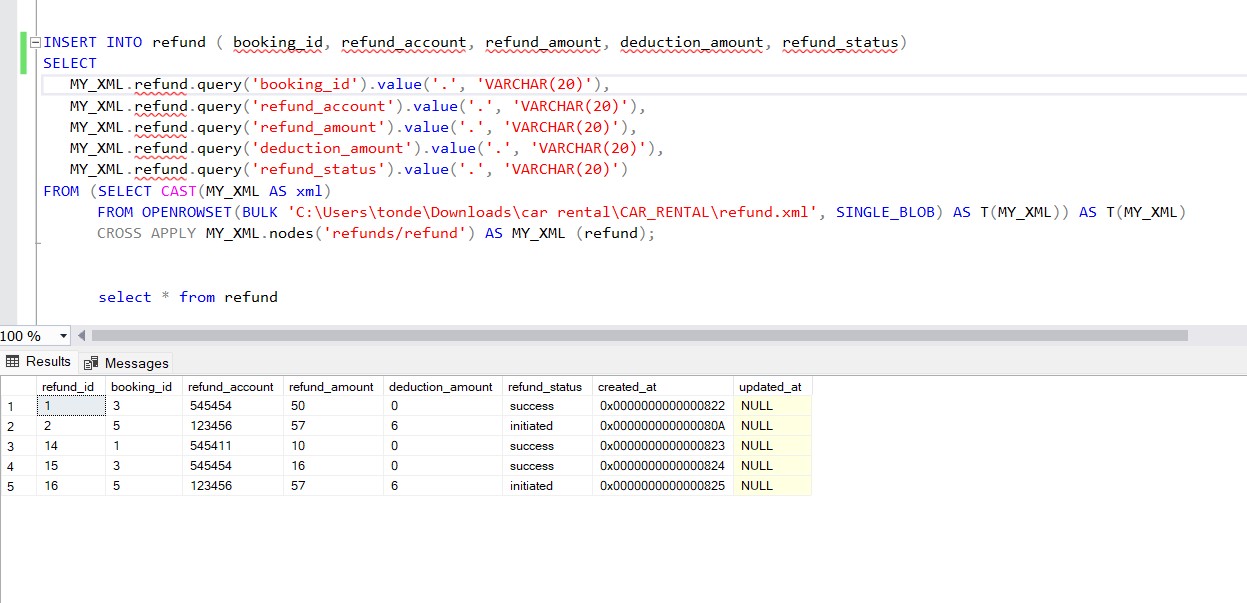
Hence The table is in the Third Normal Form. XML in schema

Here the external system is taking into consideration which processes the refunds and sends the response in terms of the XML file format.

To make our database hybrid we have used the insert queries which read the below xml file at location and reads it and inserts these values into table.

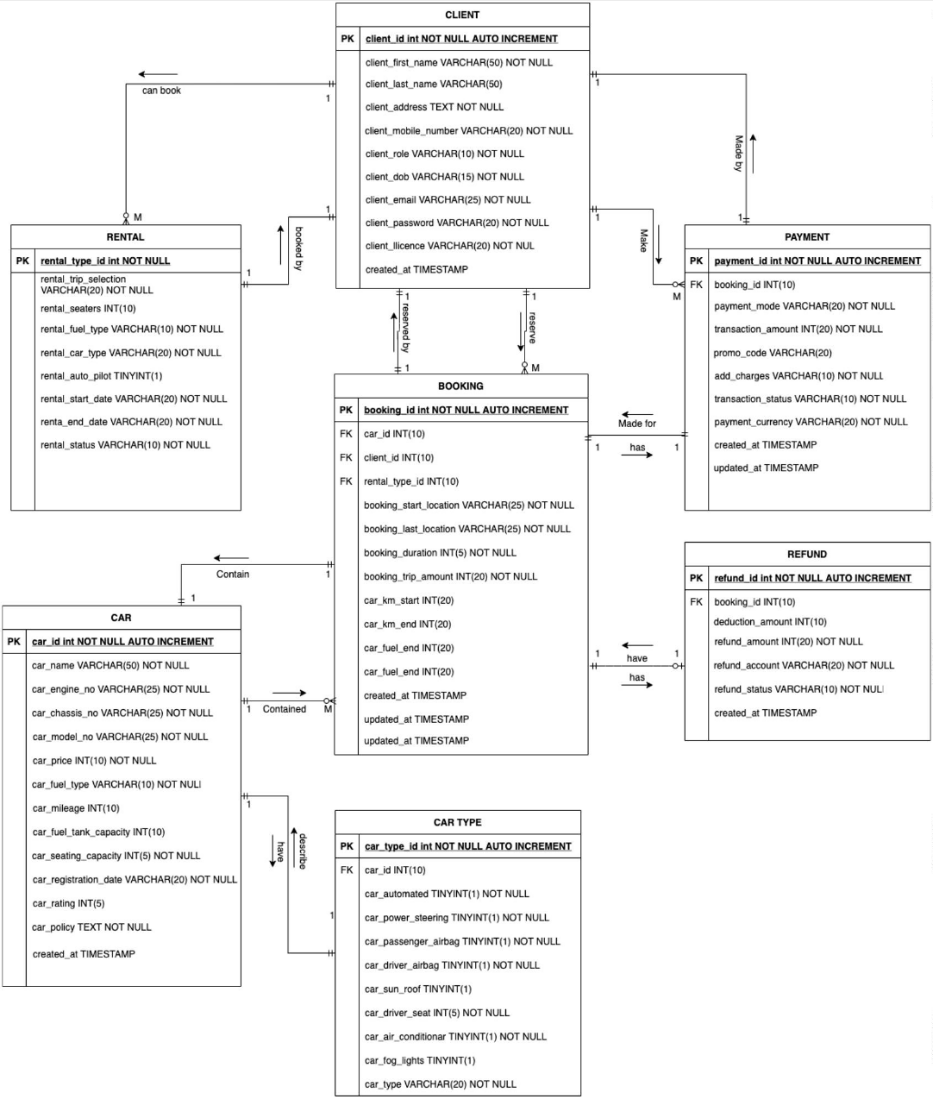


XML schema defines the shape, or structure, of an XML document, along with rules for data content and semantics such as what fields an element can contain, which sub elements it can contain and how many items can be present. It can also describe the type and values that can be placed into each element or attribute. The XML data constraints are called facets and include rules such as min and max length.



Data Diagram



Entity Relationship Diagram

Implementation in SQL server Tables

Stored procedures

1. Create a procedure so that user can search the available cars as per their desired ratings

Here the input is taken from the user and available cars more than that rating are shown to the user :

CREATE PROCEDURE search\_by\_rating

-- Add the parameters for the stored procedure here @rating int = 1

AS BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements. SET NOCOUNT ON;

-- Insert statements for procedure here

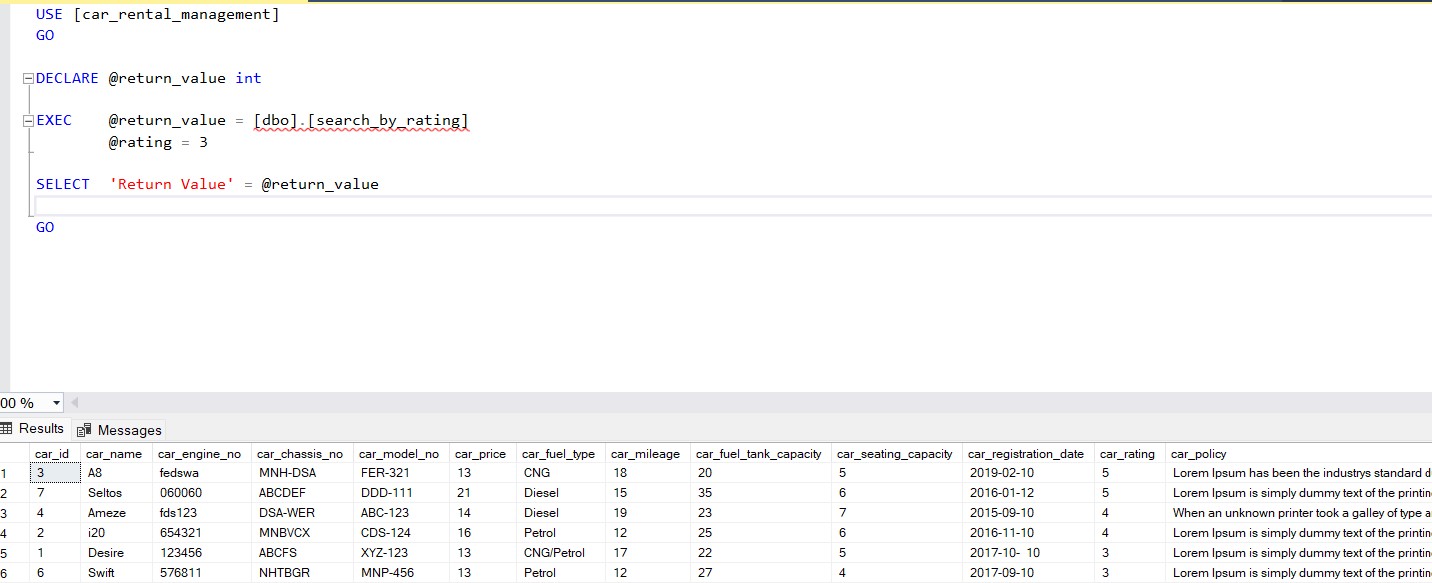
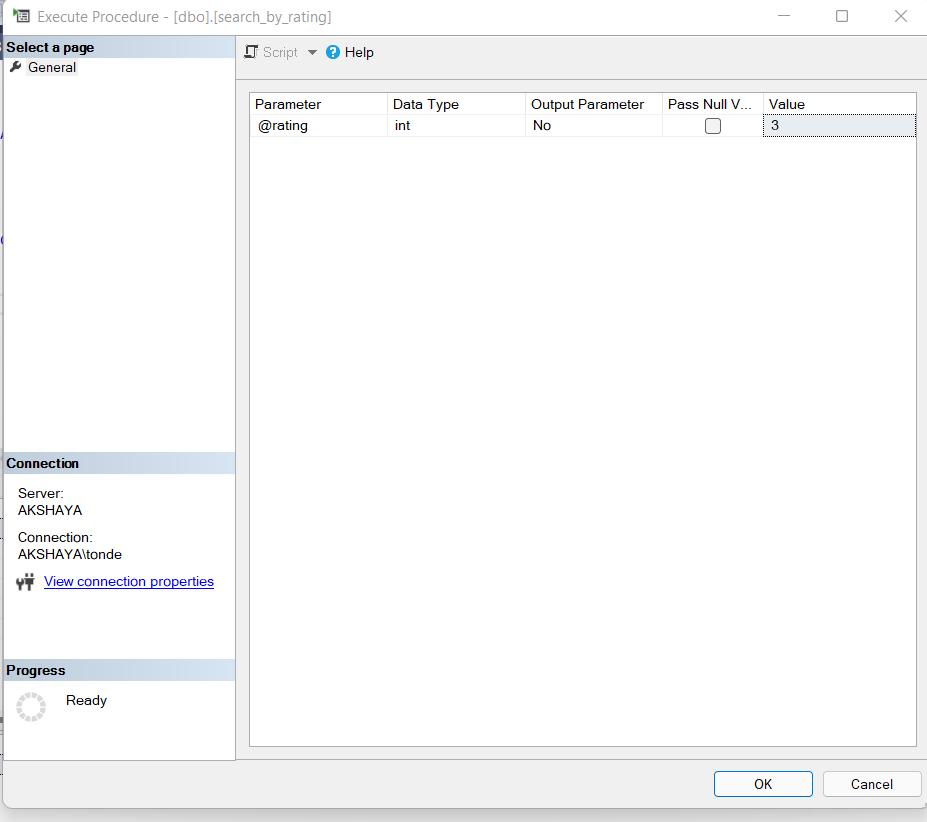
select \* from car

where car\_rating >= @rating order by car\_rating desc

END

GO

Output:



1. This procedure helps user to find the count of cars as per their type

CREATE PROCEDURE cartypes\_count AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

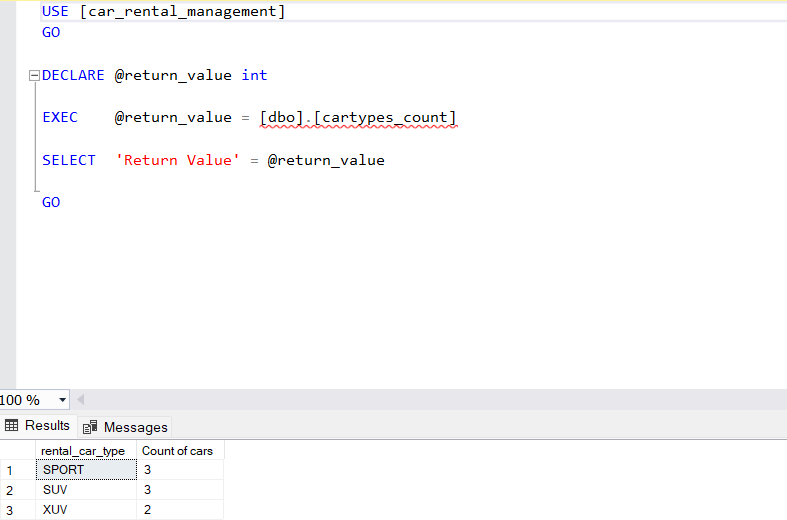
-- interfering with SELECT statements. SET NOCOUNT ON;

select rental\_car\_type, count(\*) as 'Count of cars' from rental\_type group by rental\_car\_type

having count(\*) >0

--having rental\_auto\_pilot = 0

END GO



1. Procedure to view the transaction details in a go. Here the payment and refund tables are joined so that the entire details can be viewed in a go.

CREATE PROCEDURE transaction\_details AS

BEGIN

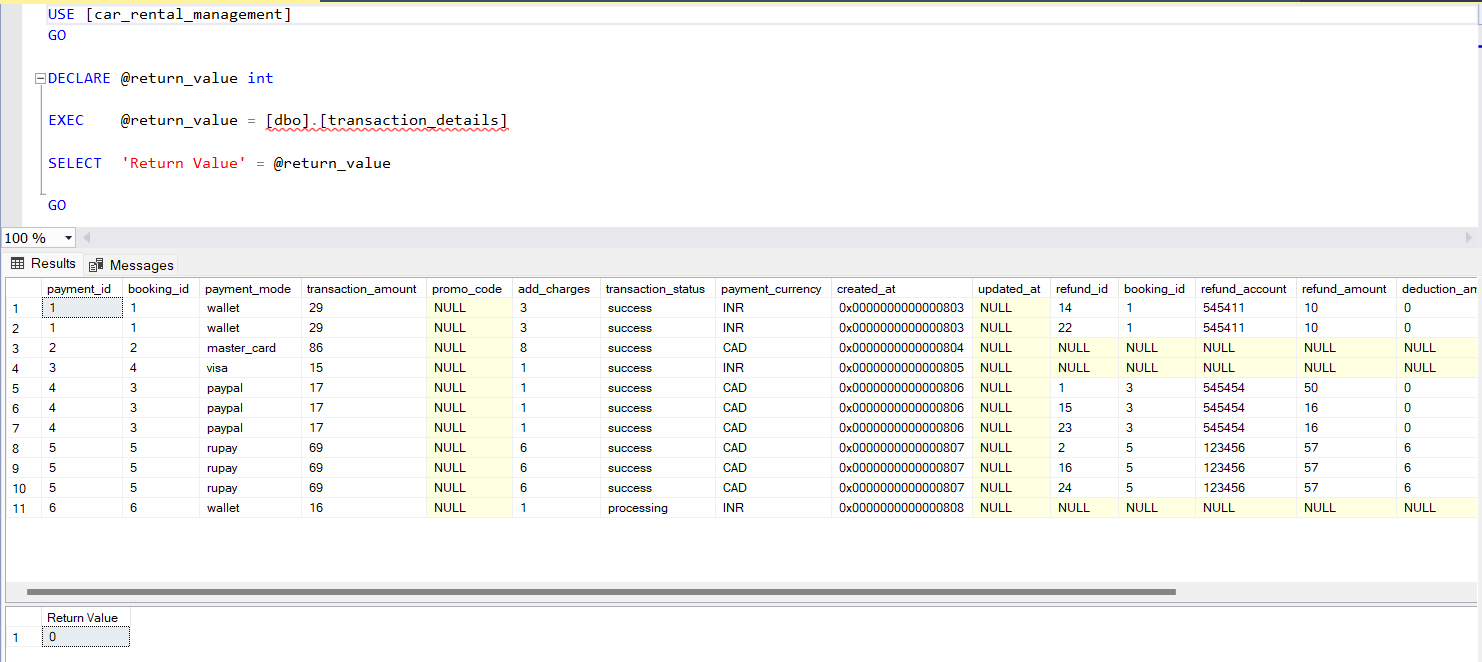
-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements. SET NOCOUNT ON;

select \* from payment full outer join refund on payment.booking\_id = refund.booking\_id

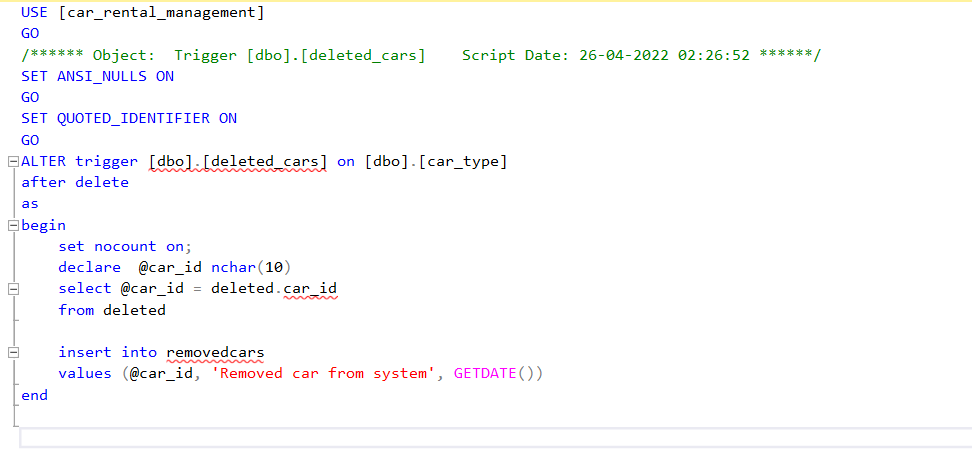
END GO

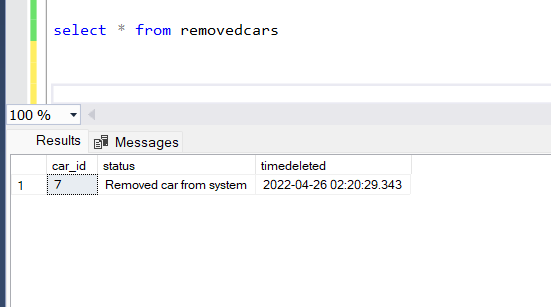
Output:



Triggers

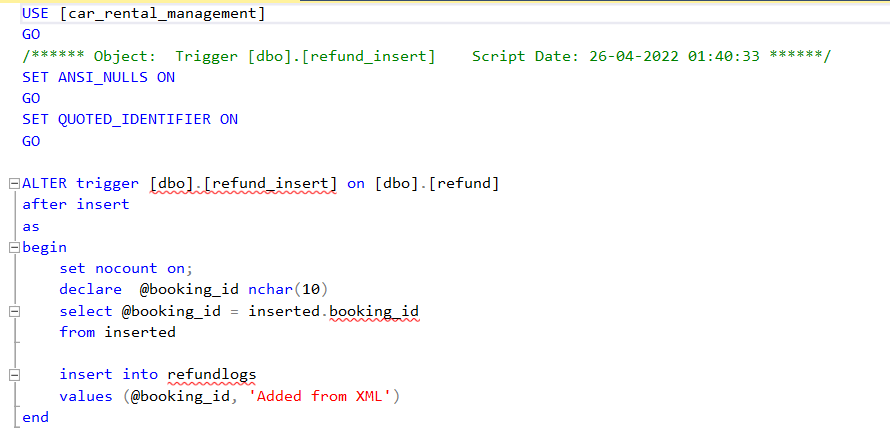
1. When a car type is no longer supported or removed from the system there should be a log of such vehicles and their IDs hence this trigger is designed to track such deleted /no longer supported car types



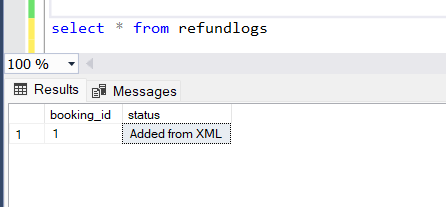


1. As data in refund table is inserted through the XML we needed a trigger to track the information on insert for the refund table.

The logs will be maintained into the refundlogs table whenever there is an insert into the table



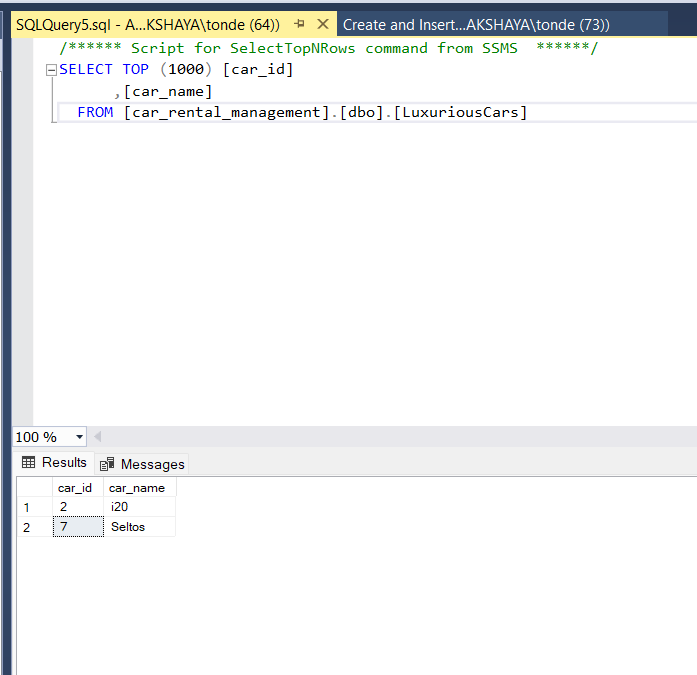
Output when there is an insert into the table



Views 1)

CREATE VIEW LuxuriousCars AS SELECT car.car\_id, car.car\_name FROM car inner join car\_type

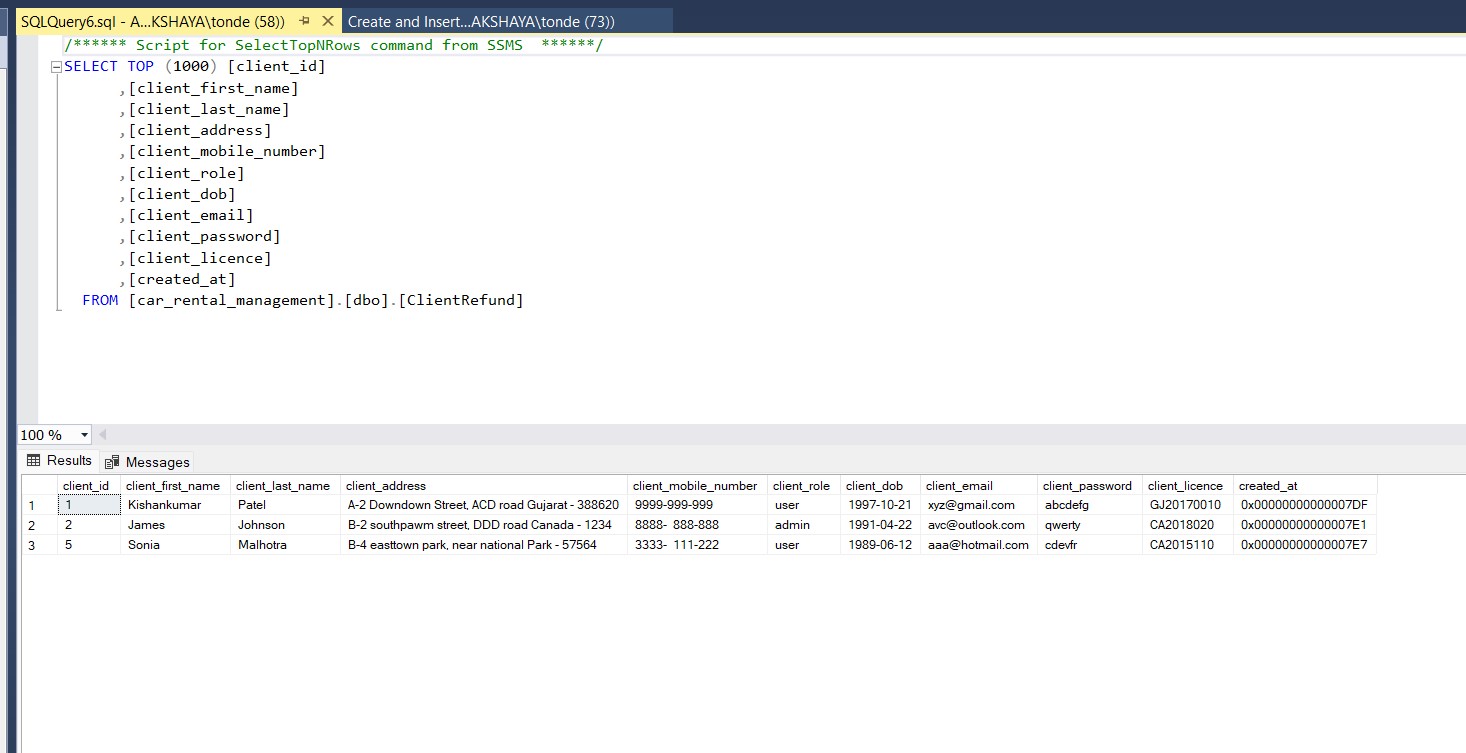
on car.car\_id = car\_type.car\_id and car.car\_seating\_capacity > 5 and car\_type.car\_automated = 1;



2)

CREATE VIEW ClientRefund AS SELECT \*FROM client

WHERE client\_id in (SELECT booking.client\_id FROM booking INNER JOIN refund ON booking.booking\_id = refund.booking\_id);



Conclusion

Car rental business has emerged with a new goody compared to the experience where every activity concerning car rental business is limited to a physical location only. Even though the physical location has not been totally eradicated the nature of functions and how these functions are achieved has been reshaped by the power of internet.

Nowadays, customers can reserve cars online, rent car online, and have the car brought to their doorstep once the customer is a registered member or go to the office to pick the car. The web-based car rental system has offered an advantage to both customers as well as Car Rental Website to manage the business and satisfies customers need at the click of a button efficiently and effectively

Bibliography

1. Ian Sommerville, “Software Engineering”, Addison Wesley, 7th edition, 2004
2. MikeO‟Docherty, "Object-Oriented Analysis and Design Understanding System Development with UML 2.0", John Wiley & Sons Ltd, England, 2005
3. Abhishek Shukla, Rahul S.Modeling of car rental management system using unified modelling language, Journal of advanced research in modelling and simulation Volume 1 Number 2 2014
4. Nabil Mohammed, Dr. A. Govardhan, Comparison between Traditional Approach and Object-Oriented Approach in Software Engineering Development, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 6, 2011

5] T. Prince, M.Jenifer - Central Credit Based Billing System for Personal Bills, International Journal of Engineering Trends and Technology Volume 32 Number 3 Feb 2016, PP 129-131

1. R. Ramani, S. Valarmathy - Vehicle Tracking and Locking System based on GSM and GPS, Modern Education and Computer Science Press, August 2013, PP 86-93
2. Asaad M. J. Al-Hindawi, IbraheemTalib,“Experimentally Evaluation of GPS/GSM Based System Design”, Journal of Electronic System Volume 2 Number 2 June 2012
3. KunalMaurya ,Mandeep Singh, Neelu Jain, “Real Time Vehicle Tracking System using GSM and GPS Technology- An Anti-theft Tracking System,” International Journal of Electronics and Computer Science Engineering. ISSN 2277-1956/V1N3-1103-1107
4. VikramKulkarni&ViswaprakashBabu, “embedded smart car security system on face detection’, special issue of IJCCT,ISSN(Online):2231-0371, ISSN(Print):0975 7449,volume- 3,issue-1
5. Karma TshetenDorjee, Deepak Rasaily, BishalCintury"RFID-Based Automatic Vehicle Parking System using Microcontroller", International Journal of Engineering Trends and Technology (IJETT), V32(4),191-194 February 2016. ISSN:2231-5381.

Appendix A: Create Table queries

Create Tables 1)

Car CREATE TABLE `car` ( `car\_id` int NOT NULL AUTO\_INCREMENT, `car\_name` varchar(50) NOT NULL, `car\_engine\_no` varchar(25) NOT NULL, `car\_chassis\_no` varchar(25) NOT NULL,

`car\_model\_no` varchar(25) NOT NULL, `car\_price` int NOT NULL, `car\_fuel\_type` varchar(10) NOT NULL, `car\_mileage` int DEFAULT NULL, `car\_fuel\_tank\_capacity` int DEFAULT NULL, `car\_seating\_capacity` int DEFAULT NULL, `car\_registration\_date` varchar(20) DEFAULT NULL, `car\_rating` int DEFAULT NULL, `car\_policy` text, `created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`car\_id`), )

2)

CREATE TABLE `client` ( `client\_id` int NOT NULL AUTO\_INCREMENT, `client\_first\_name` varchar(50) NOT NULL, `client\_last\_name` varchar(50) DEFAULT NULL, `client\_address` text NOT NULL, `client\_mobile\_number` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT '', `client\_role` varchar(10) NOT NULL, `client\_dob` varchar(15) NOT NULL, `client\_email` varchar(25) NOT NULL, `client\_password` varchar(20) NOT NULL, `client\_licence` varchar(20) NOT NULL, `created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`client\_id`), )

3)

CREATE TABLE `car\_type` ( `car\_type\_id` int NOT NULL AUTO\_INCREMENT, `car\_id` int DEFAULT NULL, `car\_automated` int NOT NULL, `car\_power\_steering` int NOT NULL,

`car\_air\_conditionar` int NOT NULL, `car\_passenger\_airbag` int NOT NULL,

`car\_driver\_airbag` int NOT NULL, `car\_sun\_roof` int DEFAULT NULL, `car\_driver\_seat` varchar(20) NOT NULL DEFAULT '', `car\_fog\_lights` int DEFAULT NULL, `car\_type` varchar(20) DEFAULT NULL, `created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`car\_type\_id`), KEY `fk\_car\_id` (`car\_id`), CONSTRAINT `fk\_car\_id` FOREIGN KEY (`car\_id`) REFERENCES `car` (`car\_id`) )

4)

CREATE TABLE `payment` ( `payment\_id` int NOT NULL AUTO\_INCREMENT, `booking\_id` int NOT NULL, `payment\_mode` varchar(20) NOT NULL, `transaction\_amount` int NOT NULL,

`promo\_code` varchar(20) DEFAULT NULL, `add\_charges` varchar(20) NOT NULL,

`transaction\_status` varchar(10) NOT NULL, `payment\_currency` varchar(20) NOT NULL,

`created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, `updated\_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`payment\_id`), KEY `booking\_id` (`booking\_id`), CONSTRAINT `payment\_ibfk\_1` FOREIGN KEY (`booking\_id`) REFERENCES

`booking` (`booking\_id`) ) 5)

CREATE TABLE `rental\_type` ( `rental\_type\_id` int NOT NULL AUTO\_INCREMENT,

`rental\_trip\_selection` varchar(20) NOT NULL, `rental\_seaters` int DEFAULT NULL,

`rental\_fuel\_type` varchar(10) NOT NULL, `rental\_car\_type` varchar(20) NOT NULL,

`rental\_auto\_pilot` int DEFAULT NULL, `rental\_status` varchar(10) NOT NULL,

`rental\_start\_date` varchar(20) NOT NULL, `rental\_end\_date` varchar(20) NOT NULL,

`created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`rental\_type\_id`) )

6)

CREATE TABLE `booking` ( `booking\_id` int NOT NULL AUTO\_INCREMENT, `car\_id` int NOT NULL, `client\_id` int NOT NULL, `rental\_type\_id` int NOT NULL, `booking\_start\_location` varchar(25) NOT NULL, `booking\_end\_location` varchar(25) NOT NULL, `booking\_duration` int NOT NULL, `booking\_trip\_amount` int NOT NULL, `car\_km\_start` int DEFAULT NULL,

`car\_km\_end` int DEFAULT NULL, `car\_fuel\_start` int DEFAULT NULL, `car\_fuel\_end` int DEFAULT NULL, `created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`updated\_at` timestamp NULL DEFAULT NULL, `booking\_status` varchar(20) DEFAULT NULL, PRIMARY KEY (`booking\_id`), KEY `car\_id` (`car\_id`), KEY `client\_id` (`client\_id`), KEY

`rental\_type\_id` (`rental\_type\_id`), CONSTRAINT `booking\_ibfk\_1` FOREIGN KEY (`car\_id`) REFERENCES `car` (`car\_id`), CONSTRAINT `booking\_ibfk\_2` FOREIGN KEY (`client\_id`) REFERENCES `client` (`client\_id`), CONSTRAINT `booking\_ibfk\_3` FOREIGN KEY (`rental\_type\_id`) REFERENCES `rental\_type` (`rental\_type\_id`) )

7)

CREATE TABLE `refund` ( `refund\_id` int NOT NULL AUTO\_INCREMENT, `booking\_id` int NOT NULL, `refund\_account` varchar(20) NOT NULL, `refund\_amount` int NOT NULL,

`deduction\_amount` int DEFAULT '0', `refund\_status` varchar(20) NOT NULL, `created\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`refund\_id`), KEY

`booking\_id` (`booking\_id`), CONSTRAINT `refund\_ibfk\_1` FOREIGN KEY (`booking\_id`) REFERENCES `booking` (`booking\_id`) )

Appendix B: Insert into queries

* 1. Car INSERT INTO `car` (`car\_name`, `car\_engine\_no`, `car\_chassis\_no`,

`car\_model\_no`, `car\_price`, `car\_fuel\_type`, `car\_mileage`,

`car\_fuel\_tank\_capacity`, `car\_seating\_capacity`, `car\_registration\_date`,

`car\_rating`, `car\_policy`) VALUES ('Desire', '123456', 'ABCFS', 'XYZ-123', 13, 'CNG/Petrol', 17, 22, 5, '2017-10- 10', 3, 'Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industrys standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.'), ('i20', '654321', 'MNBVCX', 'CDS- 124', 16, 'Petrol', 12, 25, 6, '2016-11-10', 4, 'Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industrys standard dummy text ever since the 1500s.'), ('A8', 'fedswa', 'MNH-DSA', 'FER-321', 13, 'CNG', 18, 20, 5, '2019-02-10', 5, 'Lorem Ipsum has been the industrys standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.'), ('Ameze', 'fds123', 'DSA-WER', 'ABC-123', 14, 'Diesel', 19, 23, 7, '2015-09-10', 4, 'When an unknown printer took a galley of type and scrambled it to make a type specimen book.'), ('Verna', 'FED-SED', 'SWE-EWS', 'FGH-456', 13, 'CNG/Petrol', 15, 17, 5, '2017- 01-12', 2, 'Lorem Ipsum is simply dummy text of the printing and typesetting industry. When an unknown printer took a galley of type and scrambled it to make a type specimen book.'), ('Swift', '576811', 'NHTBGR', 'MNP-456', 13, 'Petrol', 12, 27, 4, '2017-09-10', 3, 'Lorem Ipsum is simply

dummy text of the printing and typesetting industry. When an unknown printer took a galley of type and scrambled it to make a type specimen book.'), ('Seltos', '060060', 'ABCDEF', 'DDD-111', 21, 'Diesel', 15, 35, 6, '2016-01-12', 5, 'Lorem Ipsum is simply

dummy text of the printing and typesetting industry. Lorem Ipsum has been the industrys standard dummy text ever since the 1500s.');

* 1. Car Type INSERT INTO `car\_type` (`car\_id`, `car\_automated`, `car\_power\_steering`,

`car\_air\_conditionar`, `car\_passenger\_airbag`, `car\_driver\_airbag`, `car\_sun\_roof`,

`car\_driver\_seat`, `car\_fog\_lights`, `car\_type`) VALUES (1, 0, 1, 1, 0, 1, 0, 'Left', 0,

'SUV'), (2, 1, 0, 1, 0, 1, 0, 'Right', 1, 'XUV'), (3, 0, 1, 0, 1, 0, 1, 'Left', 0, 'SPORT'), (4, 0, 0,

0, 1, 1, 1, 'Left', 1, 'SUV'), (5, 1, 1, 1, 0, 0, 0, 'Right', 1, 'SPORT'), (6, 0, 0, 1, 1, 0, 0,

'Right', 0, 'SUV'), (7, 1, 1, 0, 0, 1, 1, 'Left', 1, 'XUV');

* 1. Client INSERT INTO `client` (`client\_first\_name`, `client\_last\_name`, `client\_address`,

`client\_mobile\_number`, `client\_role`, `client\_dob`, `client\_email`, `client\_password`,

`client\_licence`) VALUES ('Kishankumar', 'Patel', 'A-2 Downdown Street, ACD road Gujarat - 388620', '9999-999-999', 'user', '1997-10-21', 'xyz@gmail.com', 'abcdefg', 'GJ20170010'), ('James', 'Johnson', 'B-2 southpawm street, DDD road Canada - 1234', '8888- 888-888', 'admin', '1991-04-22', 'avc@outlook.com', 'qwerty', 'CA2018020'), ('Vin', 'Paul', 'C-1 westpawm street, CAS road India - 4564', '7777-888-666', 'user', '1992-08-27', 'bvc@outlook.com', 'rrrrrr', 'GJ2014090'), ('Bond', 'Smith', 'B-4 northtown park, near social Park - 76756', '1111-222- 333', 'superadmin', '1985-01- 31', 'ddd@hotmail.com', 'mnbvcx', 'CA2019210'), ('Sonia', 'Malhotra', 'B-4 easttown park, near national Park - 57564', '3333- 111-222', 'user', '1989-06-12', 'aaa@hotmail.com', 'cdevfr', 'CA2015110'), ('Christ', 'Morris', 'A-4 newtown park, near down street - 78787', '2222-222- 121', 'user', '1994-01-13', 'eee@gmail.com', 'plmokn', 'CA2020000'), ('Riyan', 'Parag', 'Z-4 nothern street, marriot road kerala - 34343', '7878-787- 878', 'user', '1995-02-28', 'cfc@hotmail.com', 'vfrtgb', 'CA2013100');

* 1. INSERT INTO `rental\_type` (`rental\_trip\_selection`, `rental\_seaters`,

`rental\_fuel\_type`, `rental\_car\_type`, `rental\_auto\_pilot`, `rental\_status`,

`rental\_start\_date`, `rental\_end\_date`) VALUES ('roundtrip', 4, 'CNG', 'SPORT', 0, 'Available', '2020-01-21'), ('oneway', 5, 'Petrol', 'SUV', 0, 'Available', '2020-02-02',

'2020-02-02'), ('oneway', 4, 'Diesel', 'SUV', 1, 'Avilable', '2020-01-19', '2020-01-19'),

('roundtrip', 7, 'CNG/Petrol', 'XUV', 0, 'NA', '2020-06-18', '2020-06-23'), ('roundtrip',

6, 'Petrol', 'SPORT', 0, 'Available', '2020-10-21', '2020-10-23'), ('oneway', 4, 'Diesel',

'SUV', 0, 'Available', '2020-11-01', '2020-11-01'), ('oneway', 5, 'CNG', 'XUV', 0, 'NA',

'2020-09-10', '2020-09-10'), ('roundtrip', 7, 'Petrol', 'SPORT', 1, 'Available', '2020-12-

20', '2020-12-23');

* 1. booking INSERT INTO `booking` (`car\_id`, `client\_id`, `rental\_type\_id`,

`booking\_start\_location`, `booking\_end\_location`, `booking\_duration`,

`booking\_trip\_amount`, `car\_km\_start`, `car\_km\_end`, `car\_fuel\_start`,

`car\_fuel\_end`, `booking\_status`) VALUES (1, 1, 1, '71.77', '21.22', 2, 26, 3002, 3100,

NULL, NULL, 'finished'), (3, 3, 4, '72.44', '22.45', 6, 78, NULL, NULL, NULL, NULL,

'failed'), (2, 5, 6, '81.11', '11.22', 1, 16, 28382, NULL, NULL, NULL, 'cancelled'), (4, 1, 3,

'71.77', '26.22', 1, 14, 50321, 50372, 78, NULL, 'finished'), (7, 2, 8, '77.09', '24.22', 3,

63, NULL, NULL, 98, 10, 'cancelled'), (5, 4, 2, '90.22', '11.07', 1, 15, 30919, NULL, 89,

NULL, 'in\_progress');

* 1. Payment INSERT INTO `payment` (`booking\_id`, `payment\_mode`,

`transaction\_amount`, `promo\_code`, `add\_charges`, `transaction\_status`,

`payment\_currency`, `created\_at`, `updated\_at`) VALUES (1, 'wallet', 29, NULL, '3',

'success', 'INR'), (2, 'master\_card', 86, NULL, '8', 'success', 'CAD'), (4, 'visa', 15, NULL,

'1', 'success', 'INR'), (3, 'paypal', 17, NULL, '1', 'success'), (5, 'rupay', 69, NULL, '6',

'success'), (6, 'wallet', 16, FIRST, NULL, '1', 'processing', 'INR');

* 1. Refund INSERT INTO `refund` (`booking\_id`, `refund\_account`, `refund\_amount`,

`deduction\_amount`, `refund\_status`, `created\_at`) VALUES (3, '545454', 16, 0,

'success'), (5, '123456', 57, 6, 'initiated');

Innovation:

* Created stored procedures such that it helps the business owners to get the frequently needed tables.
* Apart from normal data, we are handling connected tables while deleting and inserting new data through Primary key and foreign key Constraints.
* Also created Views such that it helps the business owners to get the frequently needed tables.
* Since we have implemented the data in 3rd normal form, we do not have deletion and updating anomaly, so we have created stored procedures for it.
* Since the booking and payment details are kept in the separate tables so it makes vital information more secure and abstract

Individual Contribution:

# Akshaya Chhaban Tonde

As a group we worked on creating a hybrid database for this assignment. As a starting point we have arranged a meeting to discuss what we are going to do about the project in general. Initially, we have decided the topic of our project before making any progress. We divided our work as per our assignments I took the Xml Scheme part as I did a lot of research in SQL studio, XML schema defines the shape, or structure, of an XML document, along with rules for data content and semantics such as what fields an element can contain, which sub elements it can contain and how many items can be present. I have created an external system and is taking into consideration which processes the refunds and sends the response in terms of the XML file format. To make our database hybrid we have used the insert queries which read the below xml file at location and reads it and inserts these values into table. Later I worked on Implementation in SQL Server for Stored Procedures which are stored in a relational database management system (RDBMS) as a group we have created a stored procedure with database car\_rental and later Triggers. For starting point it was hard

for me to work on this project since my knowledge in this area is not so good, but after I made some research and joining classes I have overcome the difficulties that I face and successfully finish my part.

In conclusion, while implementing the project I can say that I learned a lot about databases. Not only that, I also learned how to work together as a group on the project and scheduling times for the project. And luckily, I had the best members in our group helping each other every time.