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

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

2.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

3.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

4. 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

6.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

7. 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

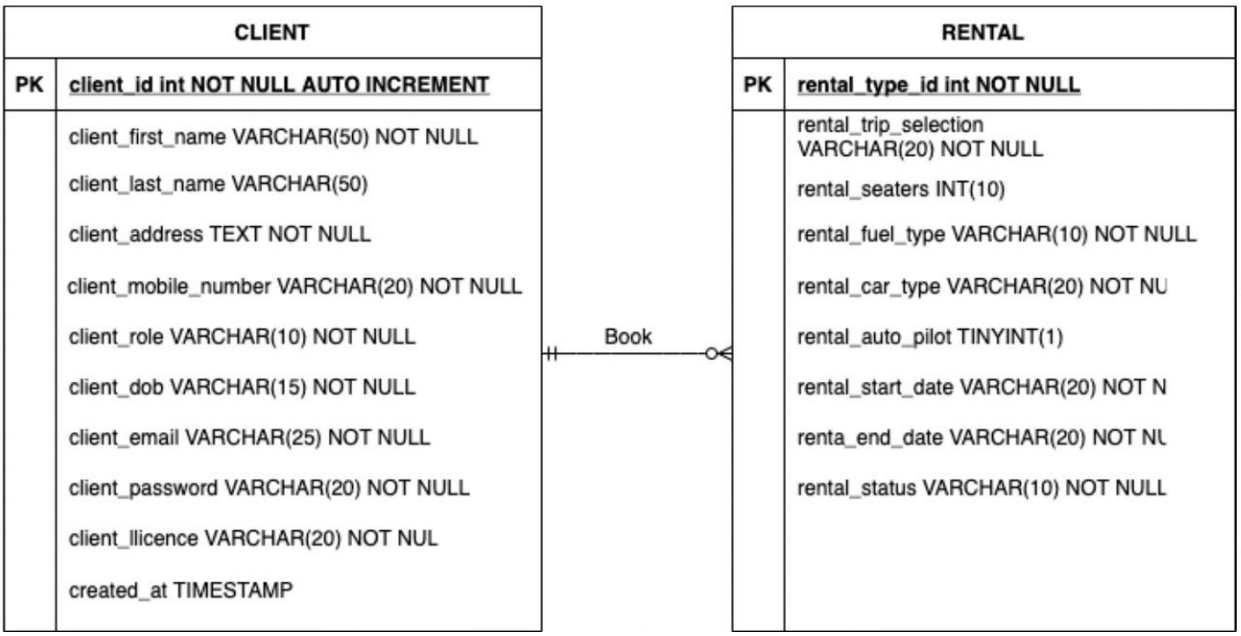
Foreign Key: - booking_id

Required Attributes: - refund_status, refund_amount, refund_account, created_at

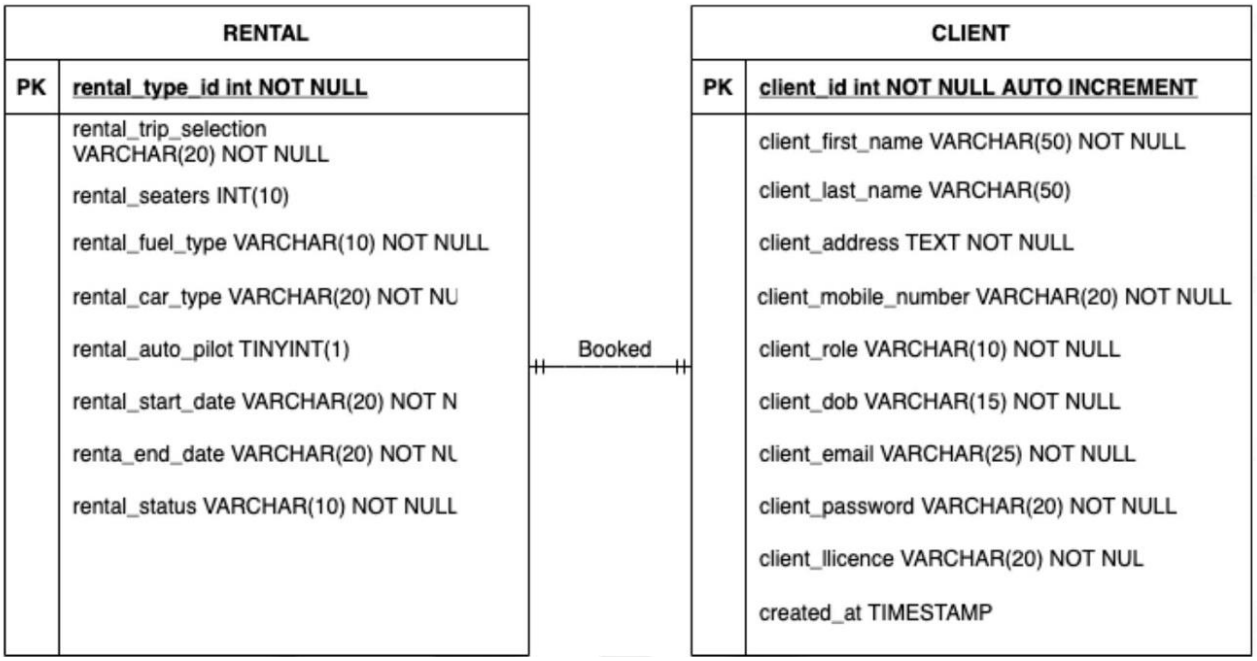
Optional Attributes: - deduction_amount

Business Rules

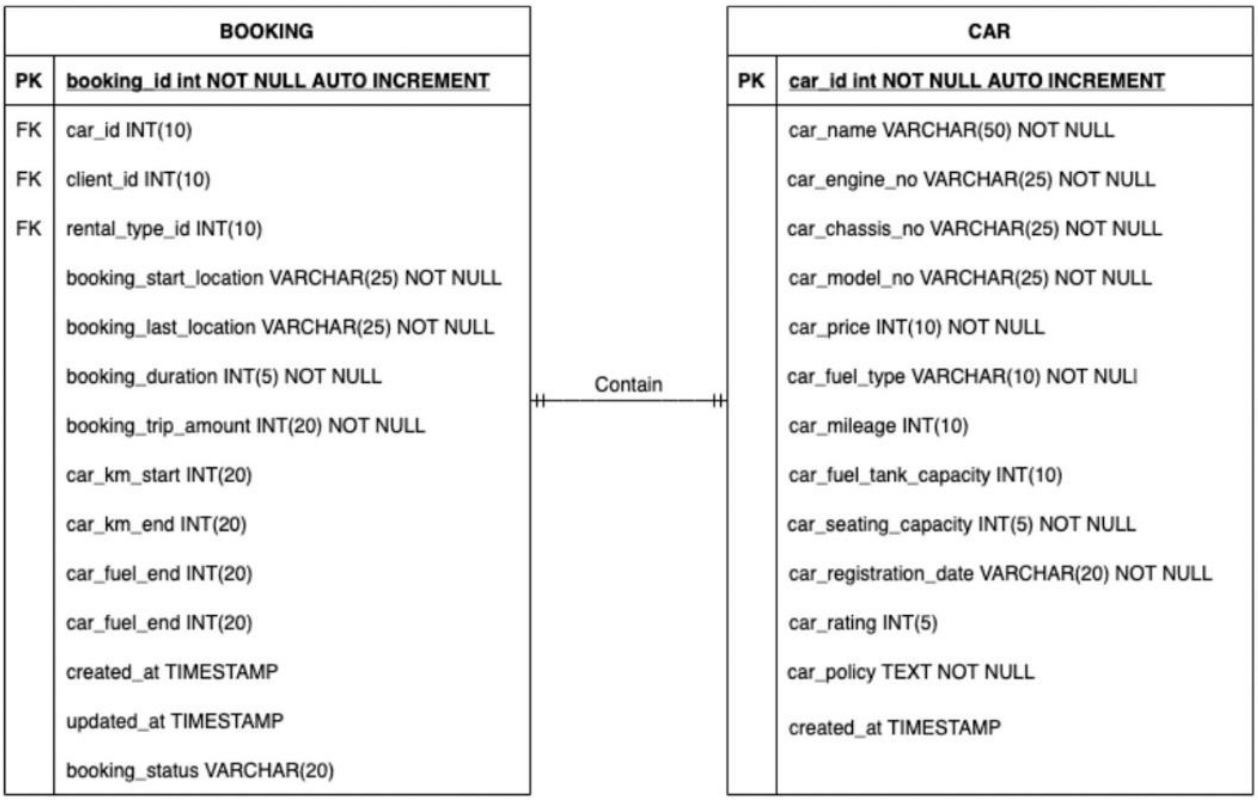
1. One client can book many rentals



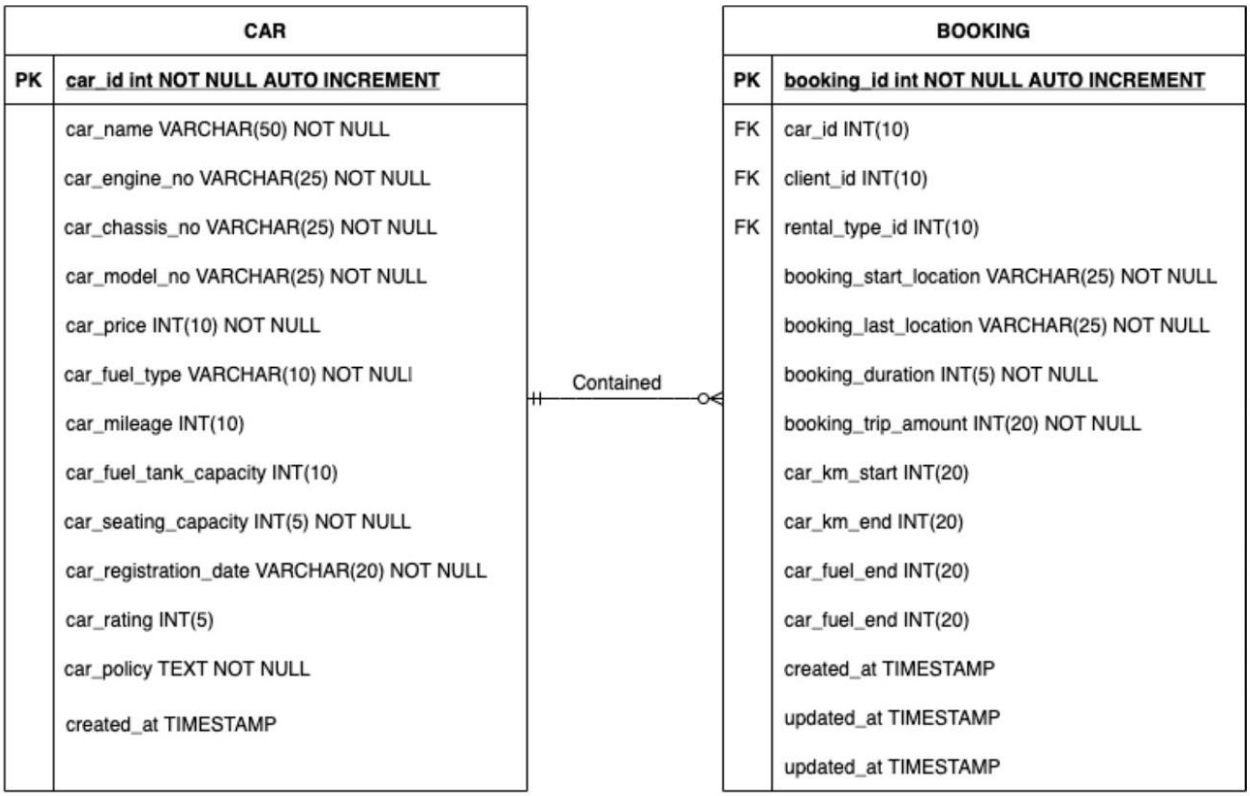
2. One rental can be booked by one client



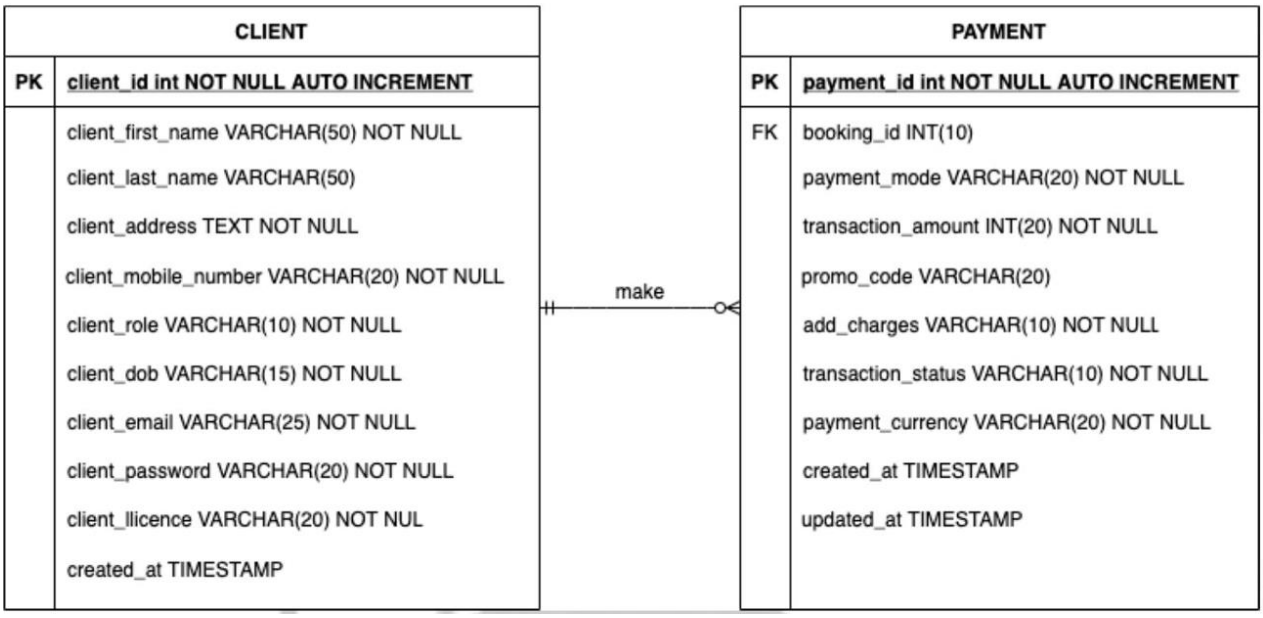
3. One booking can contain one car



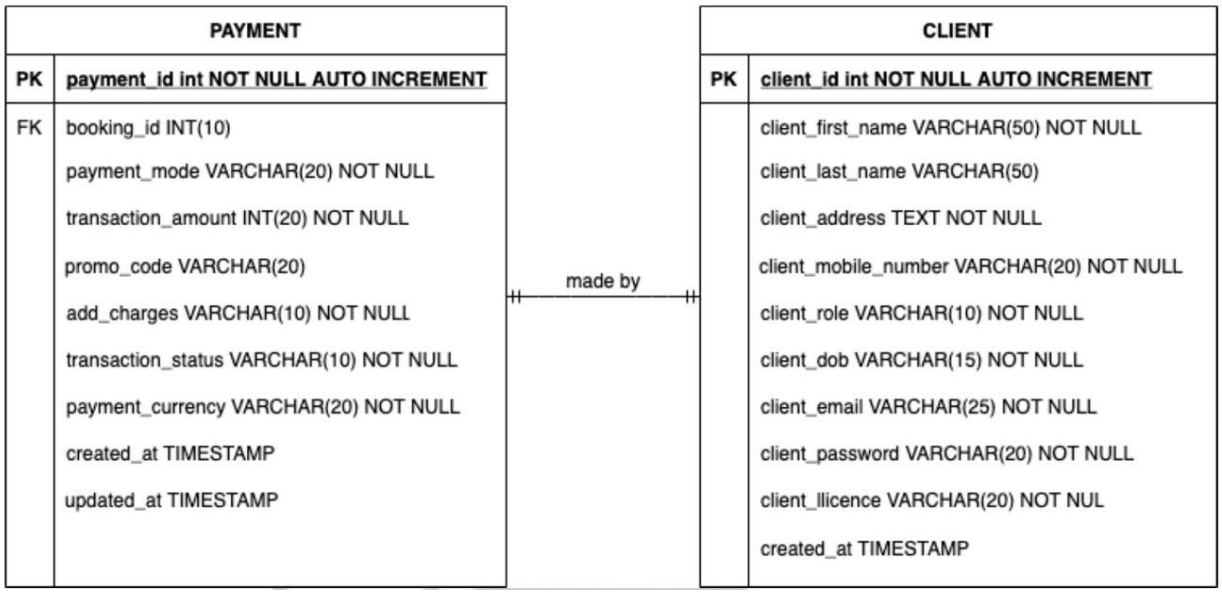
4. One car can be contained in many bookings



5. One client can make many payment



6. 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:

a. It cannot hold the multiple values.

b. All the values must be atomic

2. Second Normal Form (2NF) The entity/table is in the Second Normal Form only if:

a. It must be in the 1NF.

b. 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:

a. It must be in the 2NF.

b. It should not contain any transitive dependency.

1. Car Table

a. Primary Key is defined(car_id).

b. Every column has an atomic value.

c. No repeating groups.

Hence, the table is in the First Normal Form.

a. The table is in the First Normal Form

b. Every non-key attribute is dependent on the primary key.

Hence, the table is in the Second Normal Form.

a. The table is in the Second Normal Form.

b. It does not contain any transitive dependency.

Hence The table is in the Third Normal Form.

2. Client Table

a. Primary Key is defined(client_id).

b. Every column has an atomic value.

c. No repeating groups. Hence,

The table is in the First Normal Form.

a. The table is in the First Normal Form

b. Every non-key attribute is dependent on the primary key.

Hence, the table is in the Second Normal Form.

a. The table is in the Second Normal Form.

b. It does not contain any transitive dependency.

Hence The table is in the Third Normal Form.

3. Car Type Table

a. Primary Key is defined(car_type_id).

b. Every column has an atomic value.

c. 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.

4. Payment Table

a. Primary Key is defined(payment_id).

b. Every column has an atomic value.

c. 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

5. Rental Type Table

a. Primary Key is defined(rental_type_id).

b. Every column has an atomic value.

c. No repeating groups.

Hence, The table is in the First Normal Form.

d. 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.

6. Booking Table

a. Primary Key is defined(booking_id).

b. Every column has an atomic value.

c. No repeating groups.

d. Hence, The table is in the First Normal Form.

e. The table is in the First Normal Form

f. 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

7. Refund Table

a. Primary Key is defined(refund_id).

b.Every column has an atomic value.

c. No repeating groups.

Hence, The table is in the First Normal Form.

d. 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
<?xml version="1.0"?>
<refunds>

    <refund>
        <refund_id>1</refund_id>
        <booking_id>1</booking_id>
        <refund_account>545411</refund_account>
        <refund_amount>10</refund_amount>
        <deduction_amount>0</deduction_amount>
        <refund_status>success</refund_status>
        <created_at>2022-04-21 17:42:33</created_at>
    </refund>

    <refund>
        <refund_id>2</refund_id>
        <booking_id>3</booking_id>
        <refund_account>545454</refund_account>
        <refund_amount>16</refund_amount>
        <deduction_amount>0</deduction_amount>
        <refund_status>success</refund_status>
        <created_at>2022-04-21 17:42:33</created_at>
    </refund>

    <refund>
        <refund_id>3</refund_id>
        <booking_id>5</booking_id>
        <refund_account>123456</refund_account>
        <refund_amount>57</refund_amount>
        <deduction_amount>6</deduction_amount>
        <refund_status>initiated</refund_status>
        <created_at>2022-04-21 17:42:33</created_at>
    </refund>
</refunds>
```

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.

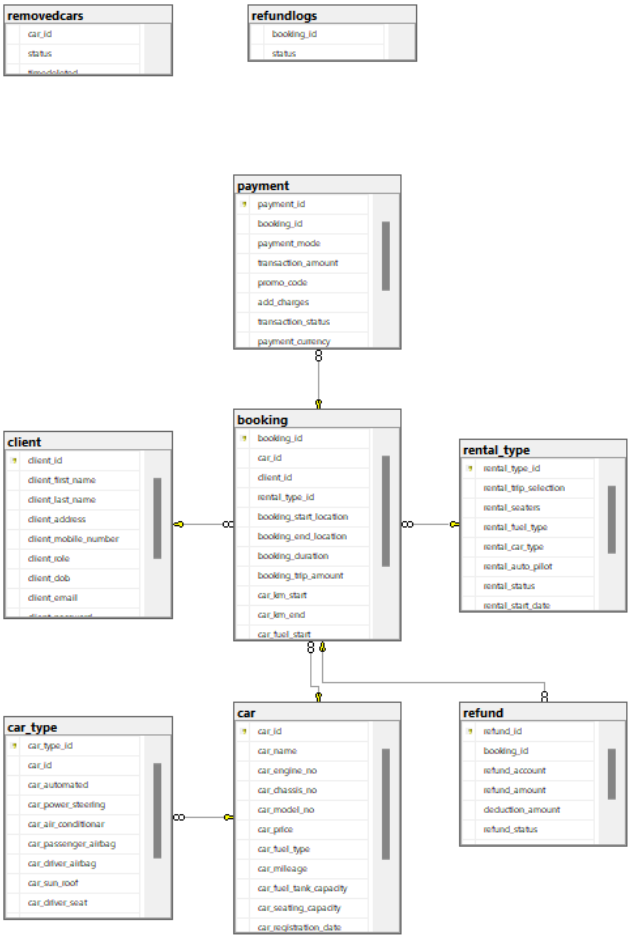
```sql
INSERT INTO refund ( booking_id, refund_account, refund_amount, deduction_amount, refund_status)
SELECT
    MY_XML.refund.query('booking_id').value('.', 'VARCHAR(20)'),
    MY_XML.refund.query('refund_account').value('.', 'VARCHAR(20)'),
    MY_XML.refund.query('refund_amount').value('.', 'VARCHAR(20)'),
    MY_XML.refund.query('deduction_amount').value('.', 'VARCHAR(20)'),
    MY_XML.refund.query('refund_status').value('.', 'VARCHAR(20)')
FROM (SELECT CAST(MY_XML AS xml)
      FROM OPENROWSET(BULK 'C:\Users\tonde\Downloads\car rental\CAR_RENTAL\refund.xml', SINGLE_BLOB) AS T(MY_XML)) AS T(MY_XML)
      CROSS APPLY MY_XML.nodes('refunds/refund') AS MY_XML (refund);


      select * from refund
```

100 %

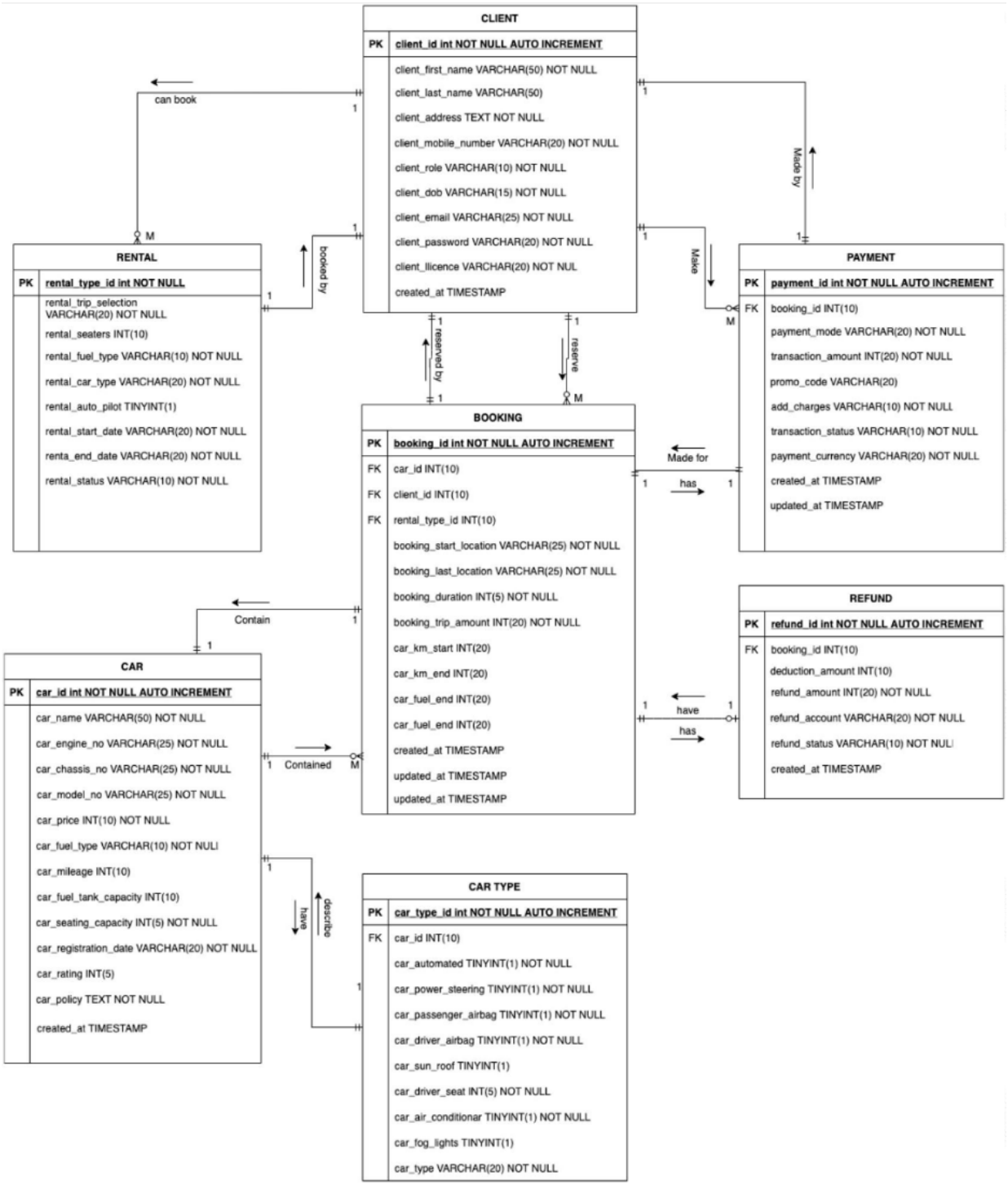
Results | Messages

	refund_id	booking_id	refund_account	refund_amount	deduction_amount	refund_status	created_at	updated_at
1	1	3	545454	50	0	success	0x0000000000000822	NULL
2	2	5	123456	57	6	initiated	0x000000000000080A	NULL
3	14	1	545411	10	0	success	0x0000000000000823	NULL
4	15	3	545454	16	0	success	0x0000000000000824	NULL
5	16	5	123456	57	6	initiated	0x0000000000000825	NULL

Data Diagram



Entity Relationship Diagram



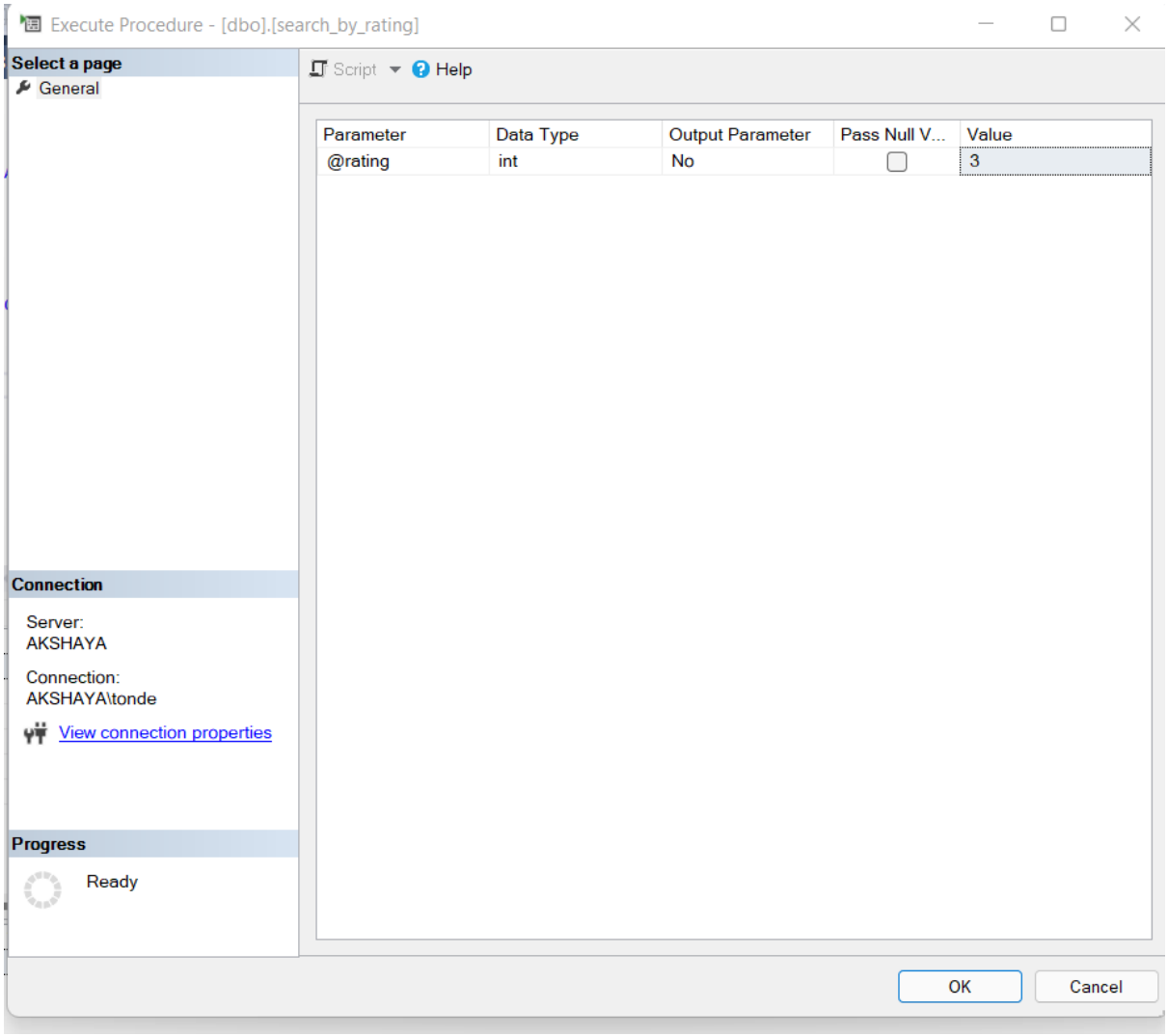
Implementation in SQL server

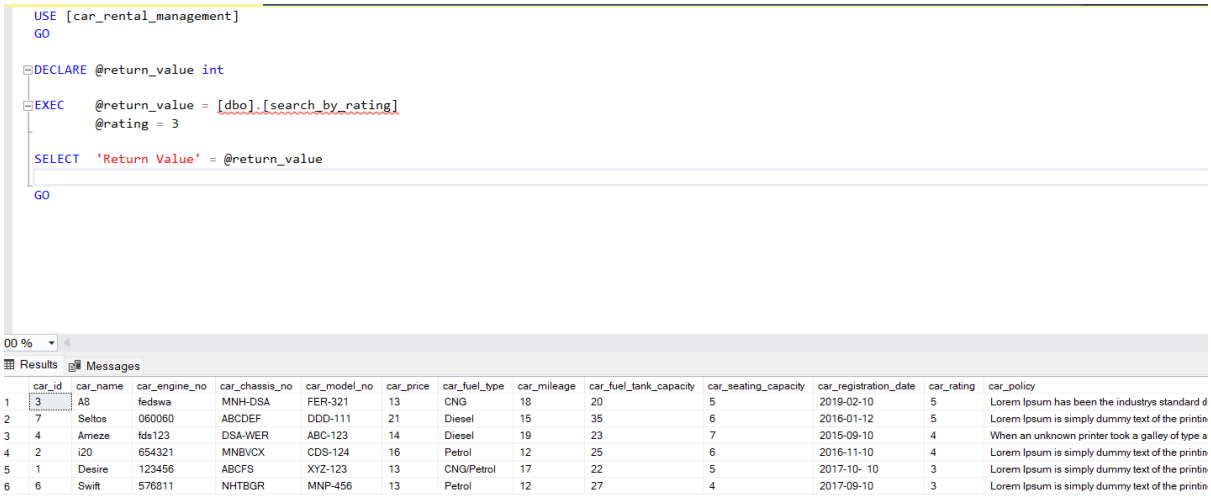
Tables

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:
```





	car_id	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
1	3	A8	fedswa	MNH-DSA	FER-321	13	CNG	18	20	5	2019-02-10	5	Lorem Ipsum has been the industrys standard d
2	7	Seltos	060060	ABCDEF	DDD-111	21	Diesel	15	35	6	2016-01-12	5	Lorem Ipsum is simply dummy text of the printin
3	4	Ameze	fds123	DSA-WER	ABC-123	14	Diesel	19	23	7	2015-09-10	4	When an unknown printer took a galley of type a
4	2	i20	654321	MNBVCX	CDS-124	16	Petrol	12	25	6	2016-11-10	4	Lorem Ipsum is simply dummy text of the printin
5	1	Desire	123456	ABCFS	XYZ-123	13	CNG/Petrol	17	22	5	2017-10- 10	3	Lorem Ipsum is simply dummy text of the printin
6	6	Swift	576811	NHTBGR	MNP-456	13	Petrol	12	27	4	2017-09-10	3	Lorem Ipsum is simply dummy text of the printin

2) 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
```

```
USE [car_rental_management]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[cartypes_count]

SELECT  'Return Value' = @return_value

GO
```

100 %

Results | Messages

	rental_car_type	Count of cars
1	SPORT	3
2	SUV	3
3	XUV	2

3) 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:

```
USE [car_rental_management]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[transaction_details]

SELECT  'Return Value' = @return_value

GO
```

100 %

Results | Messages

	payment_id	booking_id	payment_mode	transaction_amount	promo_code	add_charges	transaction_status	payment_currency	created_at	updated_at	refund_id	booking_id	refund_account	refund_amount	deduction_an
1	1	1	wallet	29	NULL	3	success	INR	0x0000000000000803	NULL	14	1	545411	10	0
2	1	1	wallet	29	NULL	3	success	INR	0x0000000000000803	NULL	22	1	545411	10	0
3	2	2	master_card	86	NULL	8	success	CAD	0x0000000000000804	NULL	NULL	NULL	NULL	NULL	NULL
4	3	4	visa	15	NULL	1	success	INR	0x0000000000000805	NULL	NULL	NULL	NULL	NULL	NULL
5	4	3	paypal	17	NULL	1	success	CAD	0x0000000000000806	NULL	1	3	545454	50	0
6	4	3	paypal	17	NULL	1	success	CAD	0x0000000000000806	NULL	15	3	545454	16	0
7	4	3	paypal	17	NULL	1	success	CAD	0x0000000000000806	NULL	23	3	545454	16	0
8	5	5	rupay	69	NULL	6	success	CAD	0x0000000000000807	NULL	2	5	123456	57	6
9	5	5	rupay	69	NULL	6	success	CAD	0x0000000000000807	NULL	16	5	123456	57	6
10	5	5	rupay	69	NULL	6	success	CAD	0x0000000000000807	NULL	24	5	123456	57	6
11	6	6	wallet	16	NULL	1	processing	INR	0x0000000000000808	NULL	NULL	NULL	NULL	NULL	NULL

	Return Value
1	0

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

```sql
USE [car_rental_management]
GO
/****** Object:  Trigger [dbo].[deleted_cars]    Script Date: 26-04-2022 02:26:52 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER trigger [dbo].[deleted_cars] on [dbo].[car_type]
after delete
as
begin
    set nocount on;
    declare  @car_id nchar(10)
    select @car_id = deleted.car_id
    from deleted

    insert into removedcars
    values (@car_id, 'Removed car from system', GETDATE())
end
```

```sql
select * from removedcars
```

100 %

Results | Messages

	car_id	status	timedeleted
1	7	Removed car from system	2022-04-26 02:20:29.343

2)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

```
USE [car_rental_management]
GO
/****** Object:  Trigger [dbo].[refund_insert]    Script Date: 26-04-2022 01:40:33 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER trigger [dbo].[refund_insert] on [dbo].[refund]
after insert
as
begin
    set nocount on;
    declare  @booking_id nchar(10)
    select @booking_id = inserted.booking_id
    from inserted

    insert into refundlogs
    values (@booking_id, 'Added from XML')
end
```

Output when there is an insert into the table

```
select * from refundlogs
```

100 %

Results | Messages

	booking_id	status
1	1	Added from XML

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;

SQLQuery5.sql - A...KSHAYA\tonde (64)) | Create and Insert...AKSHAYA\tonde (73))

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [car_id]
      ,[car_name]
  FROM [car_rental_management].[dbo].[LuxuriousCars]
```

100 %

Results | Messages

	car_id	car_name
1	2	i20
2	7	Seltos

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);

SQLQuery6.sql - A...KSHAYA\tonde (58)) | Create and Insert...AKSHAYA\tonde (73))

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [client_id]
      ,[client_first_name]
      ,[client_last_name]
      ,[client_address]
      ,[client_mobile_number]
      ,[client_role]
      ,[client_dob]
      ,[client_email]
      ,[client_password]
      ,[client_licence]
      ,[created_at]
  FROM [car_rental_management].[dbo].[ClientRefund]
```

100 %

Results | Messages

	client_id	client_first_name	client_last_name	client_address	client_mobile_number	client_role	client_dob	client_email	client_password	client_licence	created_at
1	1	Kishankumar	Patel	A-2 Downdown Street, ACD road Gujarat - 388620	9999-999-999	user	1997-10-21	xyz@gmail.com	abcdefg	GJ20170010	0x00000000000007DF
2	2	James	Johnson	B-2 southpawm street, DDD road Canada - 1234	8888- 888-888	admin	1991-04-22	avc@outlook.com	qwerty	CA2018020	0x00000000000007E1
3	5	Sonia	Malhotra	B-4 easttown park, near national Park - 57564	3333- 111-222	user	1989-06-12	aaa@hotmail.com	cdevfr	CA2015110	0x00000000000007E7

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

[6] 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

[7]Asaad M. J. Al-Hindawi, IbraheemTalib,"Experimentally Evaluation of GPS/GSM Based System Design", Journal of Electronic System Volume 2 Number 2 June 2012

[8]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

[9]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

[10] 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

a. 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.');

b. 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');

c. 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');

d. 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');

e. 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');

f. 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');

g. 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.