CAR RENTING AND BOOKING SYSTEM

INTRODUCTION

This System manages the most important components of a car rental and booking company through a modular network, which can be adapted to any country or size.it connects management of fleet, people, and business, to make the operations as efficient as possible

REQUIREMENTS

Customer

The customer will book the vehicle for renting or for own use depending on the situation like for example if the customer prefers off-roading he prefers Renting a Jeep,if the customer wants to go on a trip moving places like different towns and villages then they will prefer SUVs and if the customer wants to be in the same city/town where there is more traffic and all then it prefers sedans, hatchback such type of small vehicles which is easy to park and roam around different places.

Driver(Optional)

If the customer is renting then he will be given the driver's details.

Vehicle

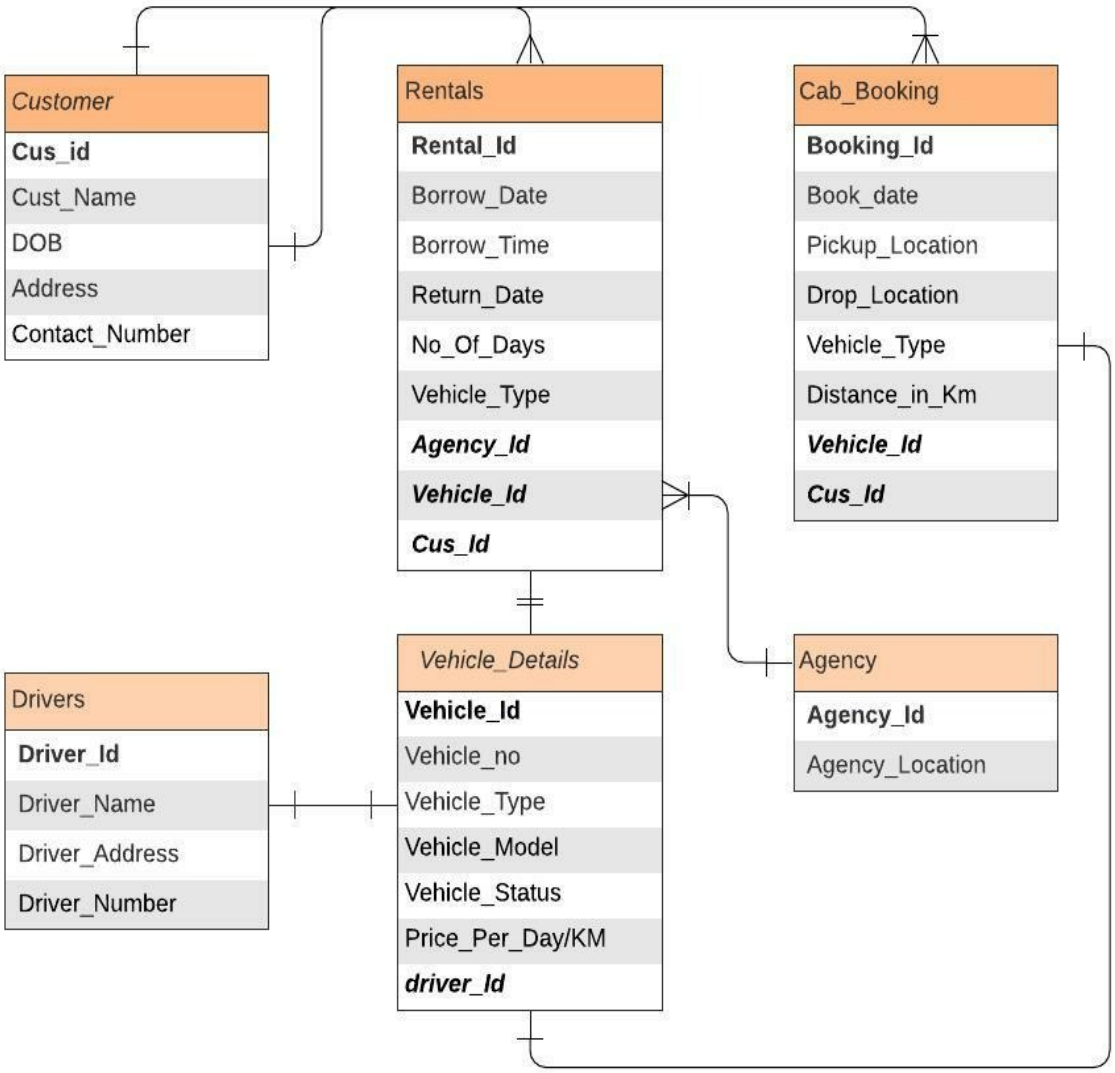
There are different types of vehicles like sedans, hatchbacks, SUV's, Jeep so based on the scenario they will choose the vehicles.
Scenarios such as Going for Off Riding, Travelling like going on a long drive to different places etc.

ERD :

Seema Naik

CAB_Booking/Rental System

October 28, 2021



Tables:

	TABLE_NAME
1	DRIVERS
2	BOOK_PAYMENT
3	RENT_PAYMENT
4	BOOK_PRICE
5	VEHICLE_DETAILS_BOOKING
6	VEHICLE_DETAILS_RENT
7	CAB_BOOKING
8	RENTALS
9	RENT_PRICE
10	AGENCY
11	CUSTOMERS

1. Customers Table

Create table customers (Cust_id int CONSTRAINT pk_cust_id PRIMARY KEY, Cust_Name varchar(10),dob date ,Address varchar(20),Contact_Number int);

```
Name            Null?    Type
--------------- -------- ------------
CUST_ID         NOT NULL NUMBER(38)
CUST_NAME                VARCHAR2(10)
DOB                      DATE
ADDRESS                  VARCHAR2(20)
CONTACT_NUMBER           NUMBER(38)
```

2. Drivers Table

Create table Drivers (Driver_id int constraint pk_driv_id primary key ,Driver_Name varchar(10), Driver_Address varchar(50), Driver_Number int);

```
Name            Null?    Type
--------------- -------- ------------
DRIVER_ID       NOT NULL NUMBER(38)
DRIVER_NAME              VARCHAR2(10)
DRIVER_ADDRESS           VARCHAR2(50)
DRIVER_NUMBER            NUMBER(38)
```

3.Vehicle_Details for Booking

Create table Vehicle_Details_Booking(Vehicle_id int constraint pk_vchl_no primary key, Vehicle_No varchar(20), Vehicle_Type varchar(20),Vehicle_Model varchar(50), Vehicle_Status varchar(50),price_per_km int, Driver_id int constraint fk_drivr_id REFERENCES Drivers(Driver_id));

```
Name           Null?    Type
-------------- -------- ------------
VEHICLE_ID     NOT NULL NUMBER(38)
VEHICLE_NO              VARCHAR2(20)
VEHICLE_TYPE            VARCHAR2(20)
VEHICLE_MODEL           VARCHAR2(50)
VEHICLE_STATUS          VARCHAR2(50)
PRICE_PER_KM            NUMBER(38)
DRIVER_ID               NUMBER(38)
```

4. Vehicle_Details for Rentals

Create table Vehicle_Details_Rent(Vehicle_id int constraint pk_rent_vchl_no primary key, Vehicle_No varchar(20), Vehicle_Type varchar(20),Vehicle_Model varchar(50), Vehicle_Status varchar(50),price_per_day int, Driver_id int constraint fk_rent_drivr_id REFERENCES Drivers(Driver_id));

```
Name           Null?    Type
-------------- -------- ------------
VEHICLE_ID     NOT NULL NUMBER(38)
VEHICLE_NO              VARCHAR2(20)
VEHICLE_TYPE            VARCHAR2(20)
VEHICLE_MODEL           VARCHAR2(50)
VEHICLE_STATUS          VARCHAR2(50)
PRICE_PER_DAY           NUMBER(38)
DRIVER_ID               NUMBER(38)
```

5. Agency Table

Create table Agency(Agency_id int constraint pk_agncyid primary key ,Agency_location varchar(50));

Name	Null?	Type
AGENCY_ID	NOT NULL	NUMBER(38)
AGENCY_LOCATION		VARCHAR2(50)

6. Cab_Booking Table

Create table Cab_Booking (Booking_id int constraint pk_bookid primary key ,Book_date date, Pickup_location varchar(50), Drop_location varchar(50), Vehicle_Type varchar(20),Distance_in_KM int,Vehicle_id int constraint fk_vhclno REFERENCES Vehicle_Details_Booking(Vehicle_id), Cust_id int constraint fk_cust_id REFERENCES customers(cust_id));

Name	Null?	Type
BOOKING_ID	NOT NULL	NUMBER(38)
BOOK_DATE		DATE
PICKUP_LOCATION		VARCHAR2(50)
DROP_LOCATION		VARCHAR2(50)
VEHICLE_TYPE		VARCHAR2(20)
DISTANCE_IN_KM		NUMBER(38)
VEHICLE_ID		NUMBER(38)
CUST_ID		NUMBER(38)

7. Booking Price Table

Create table book_price (id int Constraint pk_price_id Primary Key, Price int Default NULL, Book_id int CONSTRAINT fk_price_bookid REFERENCES cab_booking(booking_id));

```
Name    Null?    Type
------- -------- ----------
ID      NOT NULL NUMBER(38)
PRICE            NUMBER(38)
BOOK_ID          NUMBER(38)
```

8. Booking Payment Table

Create table book_payment (pay_id int, pay_mode varchar(20), pay_status varchar(20), book_id int CONSTRAINT fk_bookid REFERENCES cab_booking(booking_id));

```
Name       Null? Type
---------- ----- ------------
PAY_ID           NUMBER(38)
PAY_MODE         VARCHAR2(20)
PAY_STATUS       VARCHAR2(20)
BOOK_ID          NUMBER(38)
```

9. Rentals Table

Create table Rentals
(Rental_id int constraint pk_rentid primary key, Borrowing_Date date, Borrowing_Time varchar(10), Return_Date date, No_of_Days int, Vehicle_type varchar(20),
agency_id int constraint fk_agncyid REFERENCES agency(agency_id), Vehicle_id int constraint fk_vehino REFERENCES Vehicle_Details_Rent(Vehicle_id),
Cust_id int constraint fk_custm_id REFERENCES customers(Cust_id));

```
Name           Null?    Type
-------------- -------- ------------
RENTAL_ID      NOT NULL NUMBER(38)
BORROWING_DATE          DATE
BORROWING_TIME          VARCHAR2(10)
RETURN_DATE             DATE
NO_OF_DAYS              NUMBER(38)
VEHICLE_TYPE            VARCHAR2(20)
AGENCY_ID               NUMBER(38)
VEHICLE_ID              NUMBER(38)
CUST_ID                 NUMBER(38)
```

10. Rent Price Table

Create table rent_price (id int Constraint pk_priceid Primary Key, Price int Default NULL, Rent_id int CONSTRAINT fk_price_rentid REFERENCES Rentals(Rental_id));

```
Name     Null?    Type
-------- -------- ----------
ID       NOT NULL NUMBER(38)
PRICE             NUMBER(38)
RENT_ID           NUMBER(38)
```

11. Rent Payment Table

Create table rent_payment (pay_id int, pay_mode varchar(20), pay_status varchar(20), Rental_id int CONSTRAINT fk_rentid REFERENCES Rentals(Rental_id));

```
Name        Null? Type
----------- ----- ------------
PAY_ID            NUMBER(38)
PAY_MODE          VARCHAR2(20)
PAY_STATUS        VARCHAR2(20)
RENT_ID           NUMBER(38)
```

REPORTS:

Report 1 : Display Customers Details and their NearBy Agency Location

Select c.cust_id, c.cust_name, c.address,
a.agency_id, a.agency_location from customers c left join agency a on
c.address = a.agency_location ORDER BY c.cust_id asc;

	CUST_ID	CUST_NAME	ADDRESS	AGENCY_ID	AGENCY_LOCATION
1	1	Vaibhav	Vasco	3	Vasco
2	2	shubhav	Ponda	2	Ponda
3	3	vedika	Panjim	1	Panjim
4	4	anuksha	Mapusa	4	Mapusa
5	5	faizal	Margao	6	Margao
6	6	sidhi	Old Goa	5	Old Goa

Report 2 : Display Vehicles and their Rates Of Cab Booking

select vehicle_model, vehicle_type, price_per_km from vehicle_details_booking;

	VEHICLE_MODEL	VEHICLE_TYPE	PRICE_PER_KM
1	Ertiga	MVP	200
2	BMW	XUV	800
3	Audi	SEDAN	600

Report 3 : Display Customers Details and their Booking Details who have booked a CAB.

select c.cust_id, c.cust_name, c.contact_number,
cb.booking_id, cb.pickup_location, cb.drop_location, cb.distance_in_km,
v.vehicle_no, v.vehicle_model,v.vehicle_type, v.price_per_km
from cab_booking cb left join customers c on cb.cust_id =
c.cust_id left JOIN vehicle_details_booking v on cb.vehicle_id = v.vehicle_id;

	CUST_ID	CUST_NAME	CONTACT_NUMBER	BOOKING_ID	PICKUP_LOCATION	DROP_LOCATION	DISTANCE_IN_KM	VEHICLE_NO	VEHICLE_MODEL	VEHICLE_TYPE	PRICE_PER_KM
1	1	Vaibhav	7798866095	1	pamjim	phonda	8	Ga-08-F-0862	Ertiga	MVP	200
2	2	shubhav	5687468215	2	phonda	Vasco	10	Ga-04-A-0527	BMW	XUV	800
3	3	vedika	5247813675	3	mapusa	Old Goa	9	Ga-01-CD-7524	Audi	SEDAN	600

Report 4 : Display Vehicles and their Rates Of Renting

select vehicle_model, vehicle_type, price_per_day from vehicle_details_rent;

	VEHICLE_MODEL	VEHICLE_TYPE	PRICE_PER_DAY
1	Renult	HatchBack	800
2	Kia Seltos	SUV	1400
3	Chevolate	MVP	1000

Report 5 : Display Customer Details ,Rental Details and Vehicle Details of Customer Who have Rented a Car.

select c.cust_id, c.cust_name, c.contact_number,
r.rental_id, r.borrowing_date, r.return_date, r.no_of_days,
v.vehicle_no, v.vehicle_model, v.vehicle_type, v.price_per_day
from rentals r left join customers c on r.cust_id =
c.cust_id left JOIN vehicle_details_rent v on r.vehicle_id = v.vehicle_id;

	CUST_ID	CUST_NAME	CONTACT_NUMBER	RENTAL_ID	BORROWING_DATE	RETURN_DATE	NO_OF_DAYS	VEHICLE_NO	VEHICLE_MODEL	VEHICLE_TYPE	PRICE_PER_DAY
1	4	anuksha	9615731476	1	07-10-21	10-10-21	3	Ga-09-AD-0828	Renult	HatchBack	800
2	5	faizal	9534781205	2	07-10-21	12-10-21	5	Ga-06-BS-5461	Kia Seltos	SUV	1400
3	6	sidhi	7634158501	3	07-10-21	14-10-21	7	Ga-02-Q-2369	Chevolate	MVP	1000

Report 6 : Display Rented Vehicle on Specific Dates

select * from vehicle_details_rent where vehicle_id =
any(select vehicle_id from rentals where borrowing_date = '07-10-21');

	VEHICLE_ID	VEHICLE_NO	VEHICLE_TYPE	VEHICLE_MODEL	VEHICLE_STATUS	PRICE_PER_DAY	DRIVER_ID
1	1	Ga-09-AD-0828	HatchBack	Renult	UN-Occupied	800	2
2	2	Ga-06-BS-5461	SUV	Kia Seltos	Occupied	1400	3
3	3	Ga-02-Q-2369	MVP	Chevolate	Occupied	1000	6

Report 7 : Display CAB Details Booked on Specific Dates

```
select * from vehicle_details_booking where vehicle_id =
any(select vehicle_id from cab_booking where book_date = '07-10-21');
```

	VEHICLE_ID	VEHICLE_NO	VEHICLE_TYPE	VEHICLE_MODEL	VEHICLE_STATUS	PRICE_PER_KM	DRIVER_ID
1	1	Ga-08-F-0862	MVP	Ertiga	Occupied	200	1

Report 8 : Trigger to Update Vehicle Status of CAB Booked

```
Create or Replace Trigger vehicle_stat
Before Insert OR Delete OR Update
OF vehicle_status
ON vehicle_details_booking
FOR EACH ROW
WHEN (old.vehicle_id > 0)
BEGIN
        dbms_output.put_line('Vechile NO: '|| :OLD.vehicle_no);
        dbms_output.put_line('OLD Vehicle Status: '|| :OLD.vehicle_status);
        dbms_output.put_line('New Vehicle Status: '|| :NEW.vehicle_status);
END;
/
```

```
Vechile NO: Ga-08-F-0862
OLD Vehicle Status: Un-Occupied
New Vehicle Status: Occupied
```

Report 9 : Trigger to Update Vehicle Status of Rented Car

```
Create or Replace Trigger rent_vehicle_stat
Before Insert OR Delete OR Update
OF vehicle_status
ON vehicle_details_rent
FOR EACH ROW
WHEN (old.vehicle_id > 0)
BEGIN
        dbms_output.put_line('Vehicle NO: '|| :OLD.vehicle_no);
        dbms_output.put_line('OLD Vehicle Status: '|| :OLD.vehicle_status);
        dbms_output.put_line('New Vehicle Status: '|| :NEW.vehicle_status);
END;
/
```

```
Vechile NO: Ga-09-AD-0828
OLD Vehicle Status: Un-Occupied
New Vehicle Status: Occupied
```

Report 10 : Trigger to Updated CAB Booking Price after Calculation

```
Create or Replace Trigger book_price_update
Before Insert OR Delete OR Update
OF price
ON book_price
FOR EACH ROW
WHEN (old.id > 0)
BEGIN
        dbms_output.put_line('Book ID: '|| :OLD.book_id);
        dbms_output.put_line('OLD price: '|| :OLD.price);
        dbms_output.put_line('Updated Booking Price: '|| :NEW.price);
        dbms_output.put_line('');
END;
/
```

```
Book ID: 1
OLD price: 0
Updated Booking Price: 1600


1 row updated.
```

Report 11 : Trigger to Updated CAR Rental Price after Calculation

```
Create or Replace Trigger rent_price_update
Before Insert OR Delete OR Update
OF price
ON rent_price
FOR EACH ROW
WHEN (old.id > 0)
BEGIN
        dbms_output.put_line('Rental ID: '|| :OLD.rent_id);
        dbms_output.put_line('OLD price: '|| :OLD.price);
        dbms_output.put_line('Updated Rental Price: '|| :NEW.price);
        dbms_output.put_line('');
END;
/
```

```
Rental ID: 1
OLD price: 0
Updated Rental Price: 2400


1 row updated.
```

Report 12 : Procedure To Calculate Cab Booking Price

```
Declare

        pay_id book_price.id%type:=1;
        price book_price.price%type;

Begin

        Loop

        Select id,price into pay_id, price from book_price where id = pay_id;

        IF pay_id > 0 THEN

        Update book_price set price = (select cb.Distance_in_km *(select price_per_km
from vehicle_details_booking where vehicle_id = cb.vehicle_id )
        as Booking_Price from cab_booking cb where booking_id = book_price.id) where
id = pay_id;

        END IF;

        pay_id := pay_id + 1;
        EXIT WHEN pay_id > 5;

        END LOOP;

        Exception
        WHEN no_data_found THEN
        dbms_output.put_line('Data Updated Successfully');
        WHEN others THEN
        dbms_output.put_line('Error!');
End;
/
```

```
Book ID: 1
OLD price: 0
Updated Booking Price: 1600

Book ID: 2
OLD price: 0
Updated Booking Price: 8000

Book ID: 3
OLD price: 0
Updated Booking Price: 5400

Data Updated Successfully


PL/SQL procedure successfully completed.
```

Report 13 : Procedure To Calculate and Display No of Days of Car Taken on Rent

```
Declare

        rent_id rentals.rental_id%type:=1;
        brw_dt rentals.borrowing_date%type;
        rtrn_dt rentals.return_date%type;

Begin

        Loop

        Select rental_id,borrowing_date,return_date into rent_id, brw_dt, rtrn_dt from
rentals where rental_id = rent_id;

        IF rent_id > 0 THEN

        Update rentals set no_of_days = rtrn_dt - brw_dt where rental_id = rent_id;

        END IF;

        rent_id := rent_id + 1;
        EXIT WHEN rent_id > 5;

        END LOOP;

        Exception
        WHEN no_data_found THEN
        dbms_output.put_line('Data Updated Successfully');
        WHEN others THEN
        dbms_output.put_line('Error!');
End;
/

--To Display No of Days of the Car taken for Rent
Declare

        rent_id rentals.rental_id%type:=1;
```

```
        brw_dt rentals.borrowing_date%type;
        rtrn_dt rentals.return_date%type;
        nod rentals.no_of_days%type;

Begin

        Loop

        Select rental_id,borrowing_date,return_date, no_of_days into rent_id, brw_dt,
rtrn_dt, nod from rentals where rental_id = rent_id;

        IF rent_id > 0 THEN

        dbms_output.put_line('Id: ' || rent_id || ' Borrow Date: ' || brw_dt || ' Return Date: ' ||
rtrn_dt || ' NO of Days: ' || nod);

        END IF;

        rent_id := rent_id + 1;
        EXIT WHEN rent_id > 5;

        END LOOP;

        Exception
        WHEN no_data_found THEN
        dbms_output.put_line('Data Updated Successfully');
        WHEN others THEN
        dbms_output.put_line('Error!');
End;
/
```

```
Data Updated Successfully


PL/SQL procedure successfully completed.

Id: 1 Borrow Date: 07-10-21 Return Date: 10-10-21 NO of Days: 3
Id: 2 Borrow Date: 07-10-21 Return Date: 12-10-21 NO of Days: 5
Id: 3 Borrow Date: 07-10-21 Return Date: 14-10-21 NO of Days: 7
Data Updated Successfully


PL/SQL procedure successfully completed.
```

Report 14 :Procedure To Calculate Car Rental Price

```
Declare

        pay_id rent_price.id%type:=1;
        price rent_price.price%type;

Begin

        Loop

        Select id,price into pay_id, price from rent_price where id = pay_id;

        IF pay_id > 0 THEN

        Update rent_price set price = (select r.no_of_days *(select price_per_day from
vehicle_details_rent where vehicle_id = r.vehicle_id )
        as Rental_Price from rentals r where rental_id = rent_price.rent_id) where id =
pay_id;

        END IF;

        pay_id := pay_id + 1;
        EXIT WHEN pay_id > 5;

        END LOOP;

        Exception
        WHEN no_data_found THEN
        dbms_output.put_line('Data Updated Successfully');
        WHEN others THEN
        dbms_output.put_line('Error!');
End;
/
```

```
Rental ID: 1
OLD price: 0
Updated Rental Price: 2400

Rental ID: 2
OLD price: 0
Updated Rental Price: 7000

Rental ID: 3
OLD price: 0
Updated Rental Price: 7000

Data Updated Successfully


PL/SQL procedure successfully completed.
```

Report 15 : Procedure To Implement a Fine if Car Not Returned On Time

```
DECLARE
        Price rent_price.price%type;
        Return_date rentals.return_date%type:='11-10-21';
BEGIN
        IF Return_date  = '10-10-21' THEN

        UPDATE rent_price SET price = price + 0;

        ELSIF  Return_date  > '10-10-21' THEN

        UPDATE rent_price SET price = price + 500 Where id IN
        (Select rp.id From rentals r,rent_price rp, rent_payment rpp WHERE
        (r.Rental_id = rp.rent_id) and (r.Rental_id = rpp.rent_id) AND (Return_date
        >'10-10-21'));

        dbms_output.put_line(  'Car has not Returned on Time, Fine By ₹500 Per Day');
        END IF;
END;
/
```

```
Rental ID: 2
OLD price: 7000
Updated Rental Price: 7500

Rental ID: 3
OLD price: 7000
Updated Rental Price: 7500

Car has not Returned on Time, Fine By RS500 Per Day


PL/SQL procedure successfully completed.
```

SQL FILE:
https://drive.google.com/file/d/1y0TDoPMxjuLVRLTCcCxeyuCCFM9P9LUX/view?usp=sharing