



SELECT * FROM CAR

Developed by

6833032521	Jesadakorn	Netthong
6833042821	Chonchanok	Srisongkram
6833071021	Napat	Putsorn
6833123021	Thiraphat	Pholboon
6833219621	Poopet	Suppatsat
6833246521	Wattana	Jumpathong

2110322 Semester 2/2025

Department of Computer Engineering,

Faculty of Engineering, Chulalongkorn University

Rental Car Booking

Abstract

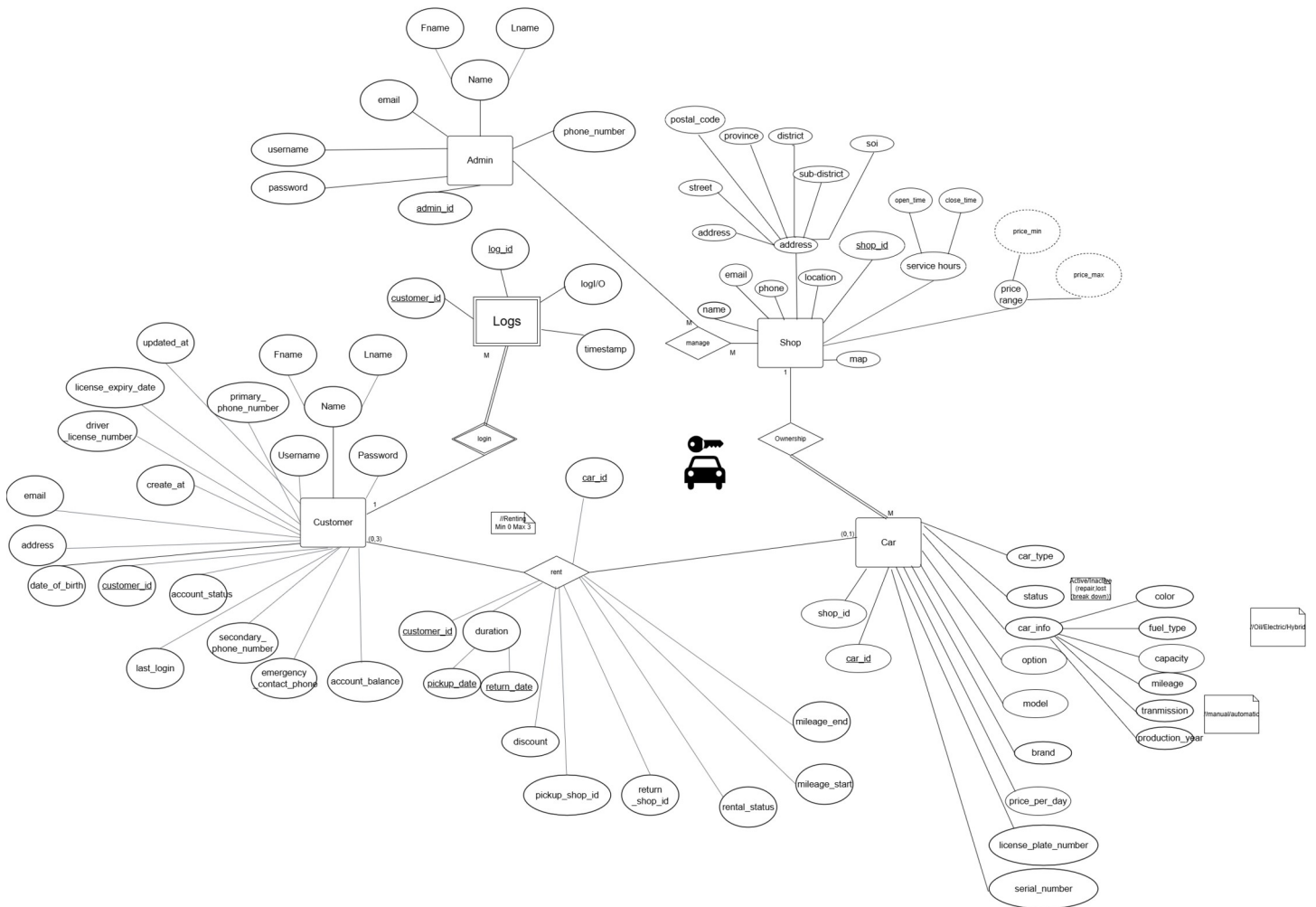
This web-based application used for car booking. When customer want to book a car, customer need to register their username, password, and customer's info (name, driver license number, email, and phone number). Then, system will generate customer_id for this user.

Customer that log-in/log-out will keep in logs using customer_id and logI/O. Customer are allowed to look for car in shops. Shop will have their own shop_id which have control over their cars by car_id.

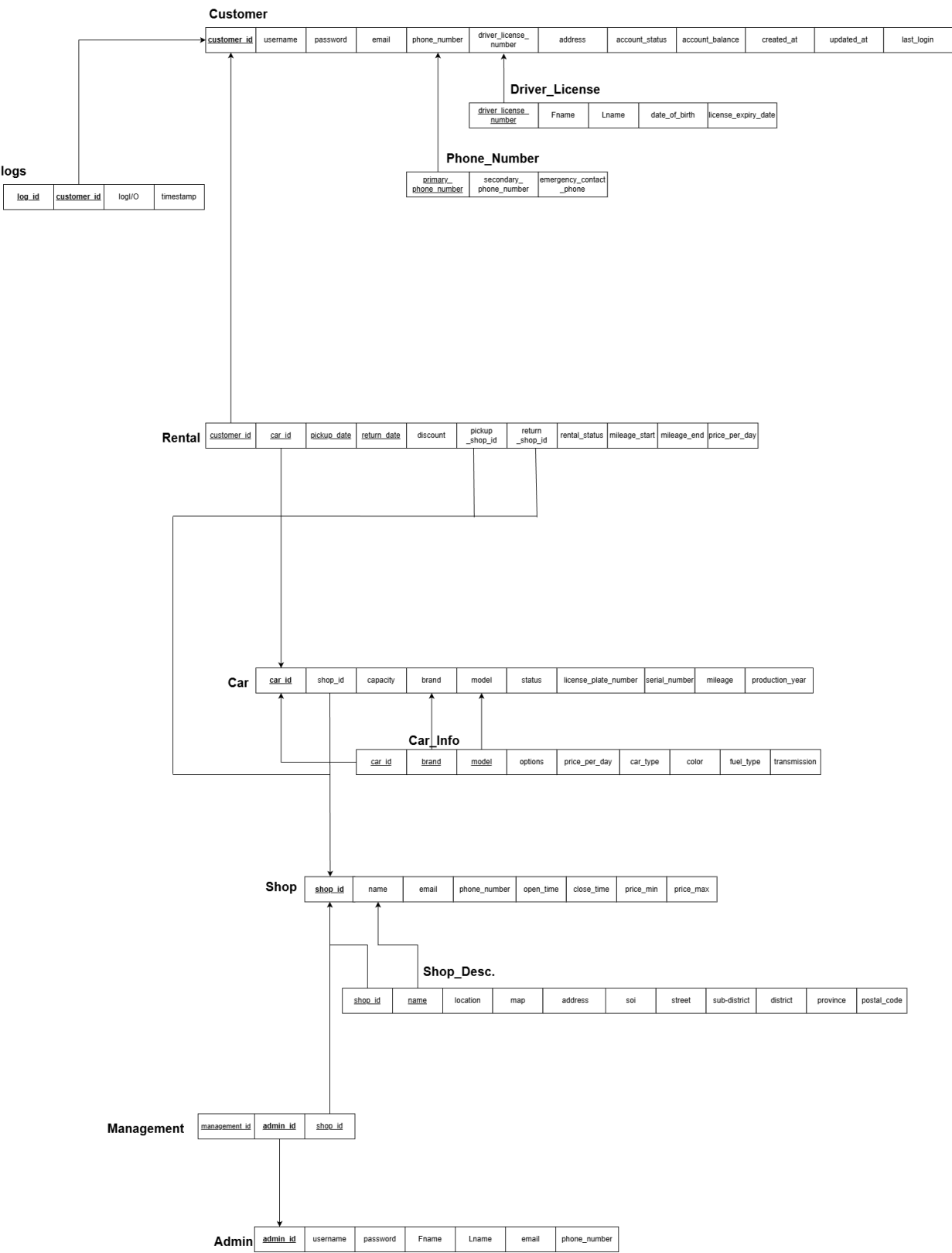
If customer wish to rent a car, customer can choose from brand, model, vehicle's capacity, fuel type, transmission, and extra options such as wheelchair accessibility etc. Customer chooses the car then add pickup date, return date, pickup shop, return shop. When the rental invoice is completed, will get rental_id which can be use to view, change or cancel the rental.

ER Diagram

ER diagram GROUP
37



Referential Integrity



SQL Query

Q.1

Suppose user have register with username, password, name, telephone number, email, driver_license (dl_number, date_of_brith, expiry_date), and address with this information (supposing user is the 67th members of the site)

Username : John123
Password : John88
First name : John
Last name : Doe
Primary phone number : 085-5555555
Secondary phone number : 087-7455654
Emergency phone number : 092-6346561
Email : johndoe123@example.com
Driver License Number : 00007543
Date of birth: 2000-01-01
Expiry date : 2030-12-31
Address : 123,Some Road,Some District, Some Country

[account_status is used for user flagging. If account_status is false ,user is getting banned or on investigation which will caused account to unusable/freeze]

Query:

```
INSERT INTO Customer (customer_id, username, pwd, email, phone_number,
dl_number, address, account_status)
VALUES
(67, 'John123', 'John88', 'johndoe123@example.com', '0855555555', '00007543',
'123,Some Road,Some District, Some Country ', true);
```

```
INSERT INTO driver_license (customer_id, dl_number, fname, lname,
date_of_birth, expire_date)
VALUES (67, '00007543', 'John', 'Doe', '2000-01-01', '2030-12-31');
```

```
INSERT INTO phone_number
(customer_id, primary_number, secondary_number, emergency_number)
VALUES (67, '085-5555555', '087-7455654', '087-7455654');
```

Q.2.1

Suppose same user ,which have user_id '67', is login (update logs)
[logI/O is used for online status. If logI/O is false, user is inactive (logged-out) but if logI/O is true then user is active (logged-in)]

Query:

```
INSERT INTO logs (log_id, customer_id, "logI/O", timestamp)
VALUES (108, 67, true, now());
```

Q.2.2

Suppose same user ,which have user_id '67', is logout (update logs) [logI/O is used for online status. If logI/O is false, user is inactive (logged-out) but if logI/O is true then user is active (logged-in)]

Query:

```
INSERT INTO logs (log_id, customer_id, "logI/O", timestamp)
VALUES (109,67,false,now());
```

Q.3

Suppose user ,which have user_id '67',is booking a car ,car_id '1234', from shop ,shop_id '30', from 2026-04-15 to 2026-04-18 and will return at the same shop with no discount

Query:

```
INSERT INTO "Rental"
(customer_id,car_id,pickup_date,return_date,discount,"pickupShop_id","return
Shop_id",rental_status,mileage_start,mileage_end,price_per_day)
VALUES (67,1234,'2026-04-15','2026-04-18',0,30,30,true,0,null,250);
```

Q.4

Suppose same user ,which have user_id '67', want to view all of his/her booking which are still active

Query:

```
SELECT * FROM "Rental"
WHERE customer_id = 67
AND return_date >= CURRENT_DATE;
```

Q.5

Suppose same user, which have user_id '67', want to edit return_date from 2026-04-18 to 2026-05-01 of the rental of car_id '1234'

Query:

```
UPDATE "Rental"
SET return_date = '2026-05-01'
WHERE customer_id = 67 AND pickup_date = '2026-04-15' AND car_id = 1234;
```

Q.6

Suppose same user, which have user_id '67', want to cancel the rental of car_id '1234'

Query:

```
DELETE FROM "Rental"  
WHERE customer_id = 67 AND pickup_date = '2026-04-15' AND return_date =  
'2026-05-01' AND car_id = 1234;
```

Q.7

Suppose the admin want to view all the rental car bookings between March 15th 2026 and March 18th 2026.

Query:

```
SELECT * FROM "Rental"  
WHERE return_date >= '2026-04-15' AND return_date < '2026-04-19';
```

Q.8

Suppose the admin want to edit price_per_day of all rental car bookings by increasing the price by \$1000.

Query:

```
UPDATE "Rental"  
SET price_per_day = price_per_day + 1000;
```

Q.9

Suppose the admin want to delete all rental car bookings that its rental_status is false.

Query:

```
DELETE FROM "Rental"  
WHERE rental_status = FALSE;
```

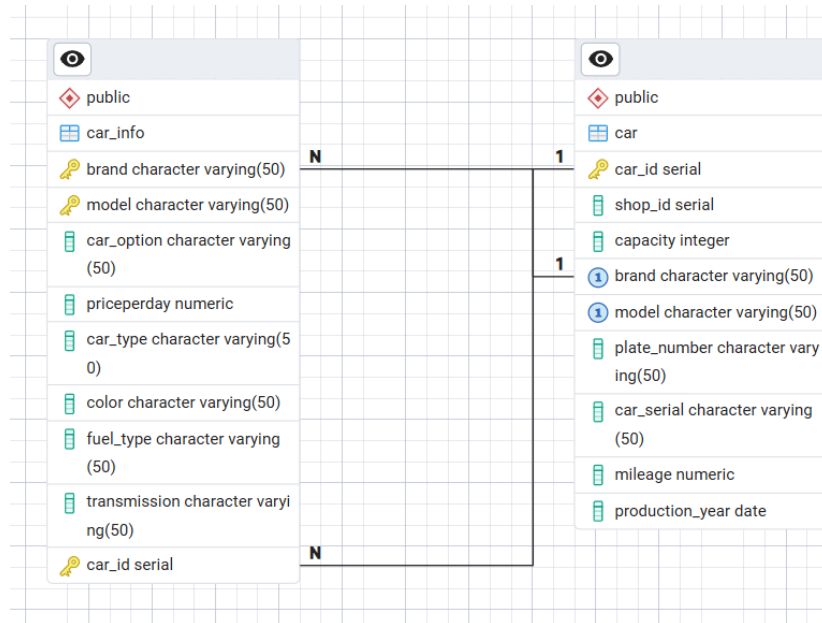
SQL Complex Query

Complex Query show the 5 busiest car brands from the last 90 days, but only the ones making more money than average.

```
SELECT
    booking_rank AS "Rank",
    brand AS "Brand",
    total_bookings AS "Total Bookings"
FROM (
    SELECT
        brand,
        total_bookings,
        brand_revenue,
        DENSE_RANK() OVER (ORDER BY total_bookings DESC) AS booking_rank
    FROM (
        SELECT
            brand,
            COUNT(car_id) AS total_bookings,
            SUM(daily_revenue) AS brand_revenue
        FROM (
            SELECT
                c.car_id,
                c.brand,
                r.customer_id,
                r.price_per_day * (r.return_date - r.pickup_date) AS
daily_revenue
            FROM "public"."car" c
            INNER JOIN "public"."Rental" r ON c.car_id = r.car_id
            WHERE r.rental_status = true
                AND r.pickup_date >= CURRENT_DATE - INTERVAL '90 days'
        ) rental_data
        GROUP BY brand
        HAVING SUM(daily_revenue) > (
            SELECT AVG(brand_revenue)
            FROM (
                SELECT SUM(price_per_day * (return_date - pickup_date)) AS
brand_revenue
                FROM "public"."car" c
                JOIN "public"."Rental" r ON c.car_id = r.car_id
                WHERE r.rental_status = true
                    AND r.pickup_date >= CURRENT_DATE - INTERVAL '90 days'
                GROUP BY c.brand
            ) avg_calc
        )
    ) brand_metrics
) ranked_brands
WHERE booking_rank <= 5
ORDER BY booking_rank ASC, brand_revenue DESC, brand ASC;
```


Document-based design schema 1 collection

ER Diagram



JSON

```
{
  title: "car",
  required: ["_id", "shop_id"],
  properties: {
    _id: { bsonType: "objectId" },
    car_id: { bsonType: "int" },
    shop_id: { bsonType: "int" },
    capacity: { bsonType: "int" },
    brand: {
      bsonType: "string",
      maxLength: 50
    },
    model: {
      bsonType: "string",
      maxLength: 50
    }
  }
}
```

```
    },
    plate_number: {
      bsonType: "string",
      maxLength: 50
    },
    car_serial: {
      bsonType: "string",
      maxLength: 50
    },
    mileage: { bsonType: ["double", "int", "decimal"] },
    production_year: { bsonType: "date" },
    car_option: {
      bsonType: "string",
      maxLength: 50
    },
    priceperday: { bsonType: ["double", "int", "decimal"] },
    car_type: {
      bsonType: "string",
      maxLength: 50
    },
    color: {
      bsonType: "string",
      maxLength: 50
    },
    fuel_type: {
      bsonType: "string",
      maxLength: 50
    },
    transmission: {
      bsonType: "string",
      maxLength: 50
    }
  }
}
}
```