



## Σχεσιακό Μοντέλο

### Vehicles

<u>vehicle_id</u>	brand	model	kilometers	color	vehicle_type	daily_cost	isRented	status
-------------------	-------	-------	------------	-------	--------------	------------	----------	--------

### Car

<u>vehicle_id</u>	license_plate	car_type	passenger_num
-------------------	---------------	----------	---------------

## Motorcycle

<u>vehicle_id</u>	license_plate
-------------------	---------------

## Bikes

<u>vehicle_id</u>	unique_num
-------------------	------------

## Scooter

<u>vehicle_id</u>	unique_num
-------------------	------------

## Users

<u>customer_id</u>	username	password	email	firstname	lastname	addresses
phone_num	driver_license_num	credit_card	age	<u>rent_id</u>		

## Rent

<u>rent_id</u>	<u>vehicle_id</u>	<u>user_id</u>	date_of_rent	return_date	rent_status	insurance	total_cost
----------------	-------------------	----------------	--------------	-------------	-------------	-----------	------------

## Πρωτεύοντα Κλειδιά

Η οντότητα Οχήματα όπως και οι οντότητες Αυτοκίνητο, Μηχανή, Ποδήλατο, Πατίνια έχουν ως πρωτεύον κλειδί το **vehicle\_id** που κάνει το κάθε όχημα μοναδικό στην βάση.

Η οντότητα Χρήστης έχει ως πρωτεύον κλειδί το **customer\_id** το οποίο δημιουργείται κατά την εγγραφή του χρήστη προκειμένου το σύστημα να αναγνωρίζει ποιος χρήστης είναι.

Η οντότητα Ενοικίαση έχει ως πρωτεύον κλειδί το **rent\_id** που προσδιορίζει την μοναδικότητα της κάθε ενοικίασης του κάθε πελάτη-χρήστη.

## Περιορισμοί Πληθικότητας

EVOL ----(N,1)---<Έχει>-(1,1)--Χρήστες----- (1,N)----<Κάνει>--(1,1)---Ενοικίαση  
EVOL ----(N,N)--<Διαθέτει>-(N,N)--Οχήματα

## Οντότητες και Γνωρίσματα Οντοτήτων

### Vehicles

1. vehicle\_id INT SERIAL PRIMARY KEY
2. brand VARCHAR(255)
3. model VARCHAR(255)
4. kilometers INT
5. color VARCHAR(255)
6. type VARCHAR(20)
7. daily\_cost INT UNSIGNED
8. isRented BOOLEAN
9. status VARCHAR(20)

### Car

1. vehicle\_id INT PRIMARY KEY
2. license\_plate VARCHAR(50)
3. car\_type VARCHAR(50)
4. numberofseats INT

### Motorcycle

1. vehicle\_id INT PRIMARY KEY
2. license\_plate VARCHAR(50)

### Bike

1. vehicle\_id INT PRIMARY KEY
2. unique\_num VARCHAR(10)

### Scooter

1. vehicle\_id INT PRIMARY KEY
2. unique\_num VARCHAR(10)

## **Users**

1. **user\_id INT PRIMARY KEY**
2. **username VARCHAR(100)**
3. **password VARCHAR(100)**
4. **email VARCHAR (100)**
5. **firstname VARCHAR(100)**
6. **lastname VARCHAR(100)**
7. **address VARCHAR(100)**
8. **phonenummer VARCHAR(100)**
9. **driverlicense VARCHAR(100)**
10. **creditCardNumber VARCHAR(100)**
11. **age INT**
12. **rent\_ids INT[ ]**

## **Rent**

1. **rent\_id INT SERIAL PRIMARY KEY**
2. **vehicle\_id INT**
3. **user\_id INT**
4. **date\_of\_rent TIMESTAMP**
5. **date\_of\_return TIMESTAMP**
6. **rentStatus VARCHAR(50)**
7. **ensurance BOOLEAN**
8. **totalcost INT**

## **Συναρτησιακές Εξαρτήσεις**

- **vehicle\_id -> brand, model, color, kilometers, type, daily cost, isRented, status.**
- **vehicle\_id -> license plate, car\_type, passenger\_num**
- **vehicle\_id -> license plate**
- **vehicle\_id -> unique\_num**
- **vehicle\_id -> unique\_num**
- **user\_id,rent\_id->username,password,email,firstname,lastname,address,phone\_num,driver\_license\_num,credit\_card,age**
- **rent\_id ->user\_id, vehicle\_id, date\_of\_rent, return\_date, rent\_status,ensurance,total\_cost**

## Τρίτη Κανονική Μορφή - 3NF

Μια σχέση R είναι σε 3η κανονική μορφή αν:

1. Αν είναι σε 1η κανονική μορφή δηλαδή δεν έχει σύνθετα και πλειότιμα γνωρίσματα.
2. Αν είναι σε 2η κανονική μορφή δηλαδή αν κάθε συναρτησιακή εξάρτηση  $X \rightarrow Y$  που υπάρχει στην R είναι full functional dependency. Μια συναρτησιακή εξάρτηση  $X \rightarrow Y$  είναι full functional dependency αν η συναρτησιακή εξάρτηση παύει να ισχύει αν αφαιρέσουμε οποιοδήποτε πεδίο από το X.
3. Αν δεν υπάρχουν μεταβατικές εξαρτήσεις. Μια συναρτησιακή εξάρτηση  $X \rightarrow Y$  δεν έχει μεταβατικές εξαρτήσεις αν ισχύει  $X \rightarrow Y$  και  $Z \rightarrow Y$  για κάποιο σύνολο από πεδία που δεν ανήκουν σε κανένα κλειδί.

Σύμφωνα με τα παραπάνω η βάση είναι σε 3η κανονική μορφή.

## Εντολές για την δημιουργία των πινάκων σε SQL

**CREATE TABLE** Vehicle (

vehicle\_id INT SERIAL PRIMARY KEY,  
brand VARCHAR(255),  
model VARCHAR(255),  
kilometers INT,  
color VARCHAR(255),  
type VARCHAR(20),  
daily\_cost INT UNSIGNED,  
isRented BOOLEAN,  
status VARCHAR(20)

);

**CREATE TABLE** Car(

vehicle\_id INT PRIMARY KEY,  
license\_plate VARCHAR(50),  
car\_type VARCHAR (50),  
numberofseats INT NOT NULL,  
FOREIGN KEY (vehicle\_id) REFERENCES Vehicle(vehicle\_id)

);

```
CREATE TABLE Motorcycle (  
    vehicle_id INT PRIMARY KEY,  
    license_plate VARCHAR(50),  
    FOREIGN KEY (vehicle_id) REFERENCES Vehicle(vehicle_id)  
);
```

```
CREATE TABLE Bikes (  
    vehicle_id INT PRIMARY KEY,  
    unique_num VARCHAR(10),  
    FOREIGN KEY (vehicle_id) REFERENCES Vehicle(vehicle_id)  
);
```

```
CREATE TABLE Scooter (  
    vehicle_id INT PRIMARY KEY,  
    unique_num VARCHAR(10),  
    FOREIGN KEY (vehicle_id) REFERENCES Vehicle(vehicle_id)  
);
```

```
CREATE TABLE User (  
    user_id SERIAL PRIMARY KEY,  
    username VARCHAR(100),  
    password VARCHAR(100),  
    email VARCHAR(100),  
    firstname VARCHAR(100)  
    lastname VARCHAR(100)  
    address VARCHAR(100)  
    phonenumber VARCHAR(100)  
    driverlicense VARCHAR(100)  
    creditcarddetails VARCHAR(100)  
    age INT  
    rent_ids INT[ ]  
);
```

```
CREATE TABLE Rent (  
    rent_id INT SERIAL PRIMARY KEY,  
    vehicle_id INT,  
    user_id INT,  
    date_of_rent TIMESTAMP,  
    date_of_return TIMESTAMP,  
    rentStatus VARCHAR(50),  
    ensurance BOOLEAN,  
    totalcost INT,  
    FOREIGN KEY (vehicle_id) REFERENCES Vehicle(vehicle_id),  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

## **Ερωτήσεις Προς την Βάση Δεδομένων με SQL**

### **Εισαγωγή Χρήστη στη Βάση**

```
INSERT INTO  
users(username,password,email,firstname,lastname,address,  
phonenumber,driverlicense,creditcardnumber,age,rent_ids)
```

### **Επιλογή Χρήστη από την Βάση**

```
Select * FROM users WHERE user_id = ?
```

### **Προσθήκη Οχήματος στη Βάση**

```
INSERT INTO  
Vehicle(color,brand,model,type,kilometers,dailyCost,isRented,status)
```

### **Προσθήκη Μηχανής στη Βάση**

```
INSERT INTO motorcycle(vehicle_id,licensePlate)
```

### **Προσθήκη Αυτοκινήτου στη Βάση**

```
INSERT INTO car(vehicle_id,numberOfSeats,licensePlate,carType)
```

### **Προσθήκη Ποδηλάτου στη Βάση**

```
INSERT INTO bicycle(vehicle_id,uniqueNumber)
```

### **Προσθήκη Πατινιού στη Βάση**

```
INSERT INTO scooter(vehicle_id,uniqueNumber)
```

### **Επιλογή Λίστας Αυτοκινήτων**

```
SELECT * FROM vehicle INNER JOIN car ON vehicle.vehicle_id =  
car.vehicle_id
```

### **Επιλογή Λίστας Μηχανών**

```
SELECT * FROM vehicle INNER JOIN motorcycle ON  
vehicle.vehicle_id = motorcycle.vehicle_id
```

### **Επιλογή Λίστας Μηχανών**

```
SELECT * FROM vehicle INNER JOIN bicycle ON vehicle.vehicle_id  
= bicycle.vehicle_id
```

### **Επιλογή Λίστας Πατινιών**

```
SELECT * FROM vehicle INNER JOIN scooter ON vehicle.vehicle_id  
= scooter.vehicle_id
```

### **Ενοικίαση Οχήματος**

```
INSERT INTO
```

```
rental(user_id,vehicle_id,rentDate,returnDate,rentStatus,ensurance,total  
Cost)
```

```
UPDATE vehicle SET isRented = true WHERE vehicle_id = ?
```

```
SELECT * FROM vehicle WHERE vehicle_id = ?
```

```
UPDATE users SET rent_ids = ? WHERE user_id = ?
```

### **Επιστροφή Οχήματος**

```
SELECT * FROM rental WHERE rent_id = ? AND rentStatus = 'rented'
```

```
UPDATE vehicle SET isRented = false WHERE vehicle_id = ?
```

```
UPDATE rental SET rentStatus = 'returned', totalCost = ? WHERE  
rent_id = ?
```

```
UPDATE users SET rent_ids = ? WHERE user_id = ?
```



### **Ζημιά Οχήματος**

```
SELECT * FROM rental WHERE rent_id = ? AND rentStatus = 'rented'  
UPDATE vehicle SET status = 'damaged',isRented = false WHERE  
vehicle_id = ?  
SELECT * FROM vehicle WHERE vehicle_id = ?  
SELECT * FROM vehicle WHERE type = ? AND isRented = false AND  
status = 'available' LIMIT 1  
UPDATE rental SET vehicle_id = ? WHERE rent_id = ?  
UPDATE users SET rent_ids = ? WHERE user_id = ?
```

### **Ατύχημα Οχήματος**

```
SELECT * FROM rental WHERE rent_id = ? AND rentStatus = 'rented'  
UPDATE rental SET totalCost = ? WHERE rent_id = ?
```

### **Λίστα Ενοικιάσεων**

```
SELECT * FROM rental WHERE rentDate >= ? AND returnDate <= ?
```

### **Ελάχιστος Χρόνος Ενοικίασης**

```
SELECT MIN(returnDate - rentDate) AS minRentTime FROM rental  
WHERE vehicle_id IN (SELECT vehicle_id FROM " + vehicleType + ")
```

### **Μέγιστος Χρόνος Ενοικίασης**

```
SELECT MAX(returnDate - rentDate) AS maxRentTime FROM rental  
WHERE vehicle_id IN (SELECT vehicle_id FROM " + vehicleType + ")
```

### **Μέσος Χρόνος Ενοικίασης**

```
SELECT AVG(returnDate - rentDate) AS averageRentTime FROM  
rental WHERE vehicle_id IN (SELECT vehicle_id FROM " +  
vehicleType + ")
```

### **Επιλογή Οχήματος**

```
SELECT * FROM vehicle WHERE vehicle_id = ?
```

### **Επιλογή Αυτοκινήτου**

```
SELECT * FROM vehicle INNER JOIN car ON vehicle.vehicle_id =  
car.vehicle_id WHERE vehicle.vehicle_id = ?
```

### **Επιλογή Μηχανής**

```
SELECT * FROM vehicle INNER JOIN motorcycle ON  
vehicle.vehicle_id = motorcycle.vehicle_id WHERE vehicle.vehicle_id =  
?
```

### **Επιλογή Ποδηλάτου**

```
SELECT * FROM vehicle INNER JOIN bicycle ON vehicle.vehicle_id =  
bicycle.vehicle_id WHERE vehicle.vehicle_id = ?
```

### **Επιλογή Πατινιού**

```
SELECT * FROM vehicle INNER JOIN scooter ON vehicle.vehicle_id =  
scooter.vehicle_id WHERE vehicle.vehicle_id = ?
```

### **Συνολικό Κόστος**

```
SELECT SUM(totalCost) AS income FROM rental WHERE rentDate >=  
? AND returnDate <= ? AND rentStatus = 'returned' AND vehicle_id IN  
(SELECT vehicle_id FROM "+type+")
```

### **Το πιο Νοικιασμένο Όχημα**

```
SELECT vehicle_id,COUNT(vehicle_id) AS count FROM rental WHERE  
vehicle_id IN (SELECT vehicle_id FROM "+type+") GROUP BY  
vehicle_id ORDER BY count DESC LIMIT 1
```