Database Normalization — Car Rentals Management System

UNF → 1NF → 2NF → 3NF Analysis

This document provides a detailed normalization process for the **Car Rentals Management System**.
Each table is shown in **Unnormalized Form (UNF)**, **1st Normal Form (1NF)**, **2nd Normal Form (2NF)**, and **3rd Normal Form (3NF)**.

--

1. CUSTOMERS TABLE

--

UNNORMALIZED FORM (UNF)

```
customer_id | full_name      | email             | phone_numbers
1           | John Doe       | john@example.com  | 555-1111, 555-2222
```

Problems in UNF

* Multi-valued attribute: `phone_numbers`
* Non-atomic attribute: `full_name`

FIRST NORMAL FORM (1NF)

```
customer_id | first_name | last_name | email             | phone
1           | John       | Doe       | john@example.com  | 555-1111
1           | John       | Doe       | john@example.com  | 555-2222   (if multiple numbers exist)
```

1NF Requirements Met

* All values are atomic
* No repeating groups

SECOND NORMAL FORM (2NF)

Primary Key: **customer_id**

* Since PK is a single attribute → no partial dependencies

2NF Table:

```
customer_id | first_name | last_name | email | phone
```

THIRD NORMAL FORM (3NF)

Check for transitive dependencies:

* None (email, phone do not determine anything else)

FINAL 3NF TABLE:

```
Customers(customer_id PK, first_name, last_name, email UNIQUE, phone)
```

--

2. CARS TABLE

--

UNNORMALIZED FORM (UNF)

```
car_id | car_info       | colors     | price
1      | Toyota Camry   | Red, Blue  | 35/day
```

Problems

* car_info is not atomic
* color is multi-valued

FIRST NORMAL FORM (1NF)

```
car_id | model | type  | color | daily_rate
1      | Camry | Sedan | Red   | 35
```

SECOND NORMAL FORM (2NF)

Primary Key: **car_id**

* No partial dependencies

THIRD NORMAL FORM (3NF)

Check non-key dependencies:

* None

FINAL 3NF TABLE:

```
Cars(car_id PK, model, type, color, daily_rate)
```

--

3. RENTALS TABLE

--

UNNORMALIZED FORM (UNF)

```
rental_id | customer_details       | car_details        | rental_dates
R01       | (1, John Doe...)       | (101, Camry...)    | 2024-01-01 to 2024-01-03
```

Problems

* Embedded fields
* Non-atomic groupings

FIRST NORMAL FORM (1NF)

```
rental_id | customer_id | car_id | rental_start_date | rental_end_date
R01       | 1           | 101    | 2024-01-01        | 2024-01-03
```

SECOND NORMAL FORM (2NF)

Primary Key: **rental_id**

* All attributes depend on rental_id
* No partial dependency

THIRD NORMAL FORM (3NF)

No transitive dependencies.

FINAL 3NF TABLE:

```
Rentals(rental_id PK, customer_id FK, car_id FK, rental_start_date, rental_end_date)
```

----

4. INVOICES TABLE

UNNORMALIZED FORM (UNF)

```
invoice_id | rental_id | amount_details
I01        | R01       | subtotal: 100, tax: 10, total: 110
```

Problems

* Non-atomic amount_details

FIRST NORMAL FORM (1NF)

```
invoice_id | rental_id | subtotal | tax | total
```

SECOND NORMAL FORM (2NF)

Primary Key: **invoice_id**

* No partial dependencies

THIRD NORMAL FORM (3NF)

Check transitive dependencies:

* total = subtotal + tax, but this is a derived calculation (allowed)
* No attribute depends on another non-key attribute

FINAL 3NF TABLE:

```
Invoices(invoice_id PK, rental_id FK, subtotal, tax, total)
```

----

5. PAYMENTS TABLE

UNNORMALIZED FORM (UNF)

```
payment_id | invoice_id | payment_info
P01        | I01        | 50 via card on 2024-01-04
```

Problems

* payment_info not atomic

FIRST NORMAL FORM (1NF)

```
payment_id | invoice_id | amount | payment_date | method
```

SECOND NORMAL FORM (2NF)

Primary Key: **payment_id**

* No partial dependencies

THIRD NORMAL FORM (3NF)

Check non-key dependencies:

* None

FINAL 3NF TABLE:

```
Payments(payment_id PK, invoice_id FK, amount, payment_date, method)
```

--

FINAL 3NF TABLE LIST (FOR SUBMISSION)

--

Customers

```
(customer_id PK, first_name, last_name, email UNIQUE, phone)
```

Cars

```
(car_id PK, model, type, color, daily_rate)
```

Rentals

```
(rental_id PK, customer_id FK, car_id FK, rental_start_date, rental_end_date)
```

Invoices

```
(invoice_id PK, rental_id FK, subtotal, tax, total)
```

Payments

```
(payment_id PK, invoice_id FK, amount, payment_date, method)
```