

Car Dealer management system - Normalization and Schema Refinement

Date : 20-10-2023

Course : MC212 DBMS

Name : Jagda Mohnish - 202203009

Puvar Kishansinh - 202203010

Deep Chhayani - 202203012

Dhaval Malsattar - 202203028

Sabalpara Jay - 202203039

Objective : To refine and normalize the initial database design to remove anomalies and redundancies

Context	page no.
1. List all the Relations & Schemas	2
2.Normalization	3
3.final relations	8

ENTITY SETS

- 1) car(car_id,price,colour,model,company_name,fuel_type,add_date,car_type)
- 2) Rental_car(rental_car_id,availability)
- 3) New_car(car_id, safety_ratings,transmission_type,warranty_year)
- 4) Old_car(car_id, year, km, seller_id, Description)
- 5) Customers(customer_id, first_name, middle_name, last_name, street_number, area_name, apt_number, city, state, pin_code, password)
- 6) phone_no(customer_id,mobile_no)
- 7) Transaction(Transaction_id, Amount,Buyer_ID, Seller_ID, Date,type,car_id)
- 8) Admin(admin_id)
- 9) Mechanic(mechanic_id, starting_date,rating)
- 10) Feedback(customer_id,time, description)
- 11) Service_Request(customer_id,request_time,description_about_service, Mechanic_id)
- 12) Bargain_Request(customer_id,car_id, price, request_time,accept_time)

RELATIONSHIP SETS

- 1) Rent: Relating Customer to Rental_car
Rent(car_id,customer_id,rented_date,due date)
- 2) Buy_newcar: relating customer to new_car
Buy_newcar(car_id,customer_id,date)
- 3) Service_accept: relating Mechanic to service_request
service_accept(customer_id,request_time,mechanic_id,accept_time, rating_per_service)

NORMALIZATION

1) car(car_id,price,colour,model,company_name,fuel_type,add_date,car_type)

Primary dependencies:

car_id → price,colour,model,company_name,fuel_type,add_date,car_type

Other dependencies:

Model → company_name

PK: car_id

Anomalies:

Update anomalies: if we change company name in 1 tuple then dependency not preserving.

insert anomalies : what if we insert model and don't know company name

Delete anomalies: if we delete all model of same company then we lose company name

So it is not in 3NF/BCNF.

Solution: we need to decompose relation

car(car_id,price,colour,model,fuel_type,add_date,car_type)

PK: car_id

FK: model references car_company

car_company(model,company_name)

PK: model

now, no non-trivial functional dependencies and therefore the both relation is in BCNF.

2) Rental_car(rental_car_id,availability)

Primary dependencies :

rental_car_id → availability

PK: rental_car_id

FK: rental_car_id references to car

no non-trivial functional dependencies and therefore the relation is in BCNF.

3) New_car(car_id, safety_ratings,transmission_type,warranty_year)

Primary dependencies :

car_id → safety_ratings,transmission_type,warranty_year

PK: car_id

FK: car_id

no non-trivial functional dependencies and therefore the relation is in BCNF.

4) Old_car(car_id, year, km, seller_id, Description)

Primary dependencies :

car_id → year, km, seller_id, Description

PK: car_id

FK: seller_id references from customers.customer_id
Car_id references to car

no non-trivial functional dependencies and therefore the relation is in BCNF

5) Customers(customer_id, first_name, middle_name, last_name, street_number, area_name, apt_number, city, state, pin_code, password)

Primary dependencies :

customer_id → first_name, middle_name, last_name, street_number, area_name
apt_number, city, state, pin_code, password

other dependencies:

pin_code →state,city

PK: customer_id

We have non-primary dependencies, So it is not in the third normal form.
So we have to decompose it.

Solution:

customer(customer_id, first_name, middle_name, last_name, street_number, area_name, apt_number,password,pin_code)

PK:customer_id
FK:pin_code references to address

address(pin_code,city,state)
PK:pin_code

Now, no non-trivial functional dependencies and therefore the relation is in BCNF.

6) phone_no(customer_id,mobile_no)

Primary dependencies :

customer_id,mobile_no \rightarrow mobile_no
customer_id,mobile_no \rightarrow customer_id

PK: customer_id,mobile_no
FK: customer_id references customers.customer_id

no non-trivial functional dependencies and therefore the relation is in BCNF

7) Transaction(Transaction_id, Amount,Buyer_ID, Seller_ID, Date,type,car_id)

Primary dependencies :

Transaction_id \rightarrow Amount,Buyer_ID, Seller_ID, Date,type,car_id

PK:Transaction_id
FK: buyer_id references customers.customer_id
seller_id references customers.customer_id
car_id references to car or null

no non-trivial functional dependencies and therefore the relation is in BCNF.

8) Admin(admin_id)

PK: admin_id
FK: admin_id references customers

no non-trivial functional dependencies and therefore the relation is in BCNF.

9) Mechanic(mechanic_id, starting_date, rating)

Primary dependencies :

mechanic_id \rightarrow starting_date, rating

PK: mechanic_id

FK: mechanic_id references customers.customer_id

no non-trivial functional dependencies and therefore the relation is in BCNF.

10) Feedback(customer_id, time, description)

Primary dependencies :

customer_id, time \rightarrow description

PK: customer_id, time

FK: customer_id references customers

no non-trivial functional dependencies and therefore the relation is in BCNF

11) Service_Request(customer_id, request_time, description_about_service, Mechanic_id)

Primary dependencies :

customer_id, request_time \rightarrow description_about_service, mechanic_id

PK: customer_id, request_time

FK: customer_id references customers.customer_id

no non-trivial functional dependencies and therefore the relation is in BCNF

12) Bargain_Request(customer_id, car_id, price, request_time, accept_time)

Primary dependencies :

customer_id, car_id, price \rightarrow request_time, accept_time

PK: customer_id, car_id, price

FK: customer_id references customers.customer_id

car_id references old_car.car_id

no non-trivial functional dependencies and therefore the relation is in BCNF

13) Rent: Relating Customer to Rental_car
Rent(car_id,customer_id,rented_date,due date)

Primary dependencies :

car_id,customer_id,rented_date → due date

PK: car_id,customer_id,rented_date

FK: car_id references Rented_car.car_id

customer_id references customer.customer_id

no non-trivial functional dependencies and therefore the relation is in BCNF

14) Buy_newcar: relating customer to new_car
Buy_newcar(car_id,customer_id,date)

Primary dependencies :

car_id → customer_id,date

PK: car_id

FK: car_id references New_car.car_id

no non-trivial functional dependencies and therefore the relation is in BCNF

15) Service_accept: relating Mechanic to service_request
service_accept(customer_id,request_time,mechanic_id,accept_time,rating_per_service)

Primary dependencies :

customer_id,request_time → mechanic_id,accept_time,rating_per_service

PK: customer_id,request_time

FK: customer_id,request_time references to Service_Request

no non-trivial functional dependencies and therefore the relation is in BCNF

FINAL SCHEMA

Entity sets:

- 1) car(car_id,price,colour,model,fuel_type,add_date,car_type)
- 2) car_company(model,company_name)
- 3) Rental_car(rental_car_id,availability)
- 4) New_car(car_id, safety_ratings,transmission_type,warranty_year)
- 5) Old_car(car_id, year, km, seller_id, Description)
- 6) customer(customer_id, first_name, middle_name, last_name, street_number, area_name, apt_number,password,pin_code)
- 7) address(pin_code,city,state)
- 8) phone_no(customer_id,mobile_no)
- 9) Transaction(Transaction_id, Amount,Buyer_ID, Seller_ID, Date,type,car_id)
- 10) Admin(admin_id)
- 11) Mechanic(mechanic_id, starting_date,rating)
- 12) Feedback(customer_id,time, description)
- 13) Service_Request(customer_id, request_time, description_about_service,Mechanic_id)
- 14) Bargain_Request(customer_id,car_id, price, request_time,accept_time)

Relationship sets:

- 1) Rent: Relating Customer to Rental_car
Rent(car_id,customer_id,rented_date,due date)
- 2) Buy_newcar: relating customer to new_car
Buy_newcar(car_id,customer_id,date)
- 3) Service_accept: relating Mechanic to service_request
service_accept(customer_id,request_time,mechanic_id,accept_time,rating_per_service)