IT-214 Database Management System

Car_Bazaar

Group 14: (Lab group: 06)

- Atik Vohra (202301447)
- Vivek Parmar (202301475)
- Neelabh Rana (202301476)
- Jenish Bhagiya (202301480)

Report For: Relational Schema, Minimal FD set, BCNF and DDL script.

Github: Link

Project Description:

Car Bazaar is a comprehensive online platform that streamlines the process of buying, selling, and servicing cars and their accessories. It offers a robust database management system where users can easily search, filter, and compare vehicles based on specific features — even down to details like a number plate for second-hand cars. In addition to car listings, the platform includes a dedicated section for accessories, complete with user reviews.

For sellers, whether individuals, private stores, or official company outlets, Car Bazaar provides a dynamic marketplace and can also provide sales analytics to help optimize performance. Buyers can also add items to their wishlist and rate both products and sellers, ensuring a trustworthy environment and empowering users to make informed decisions.

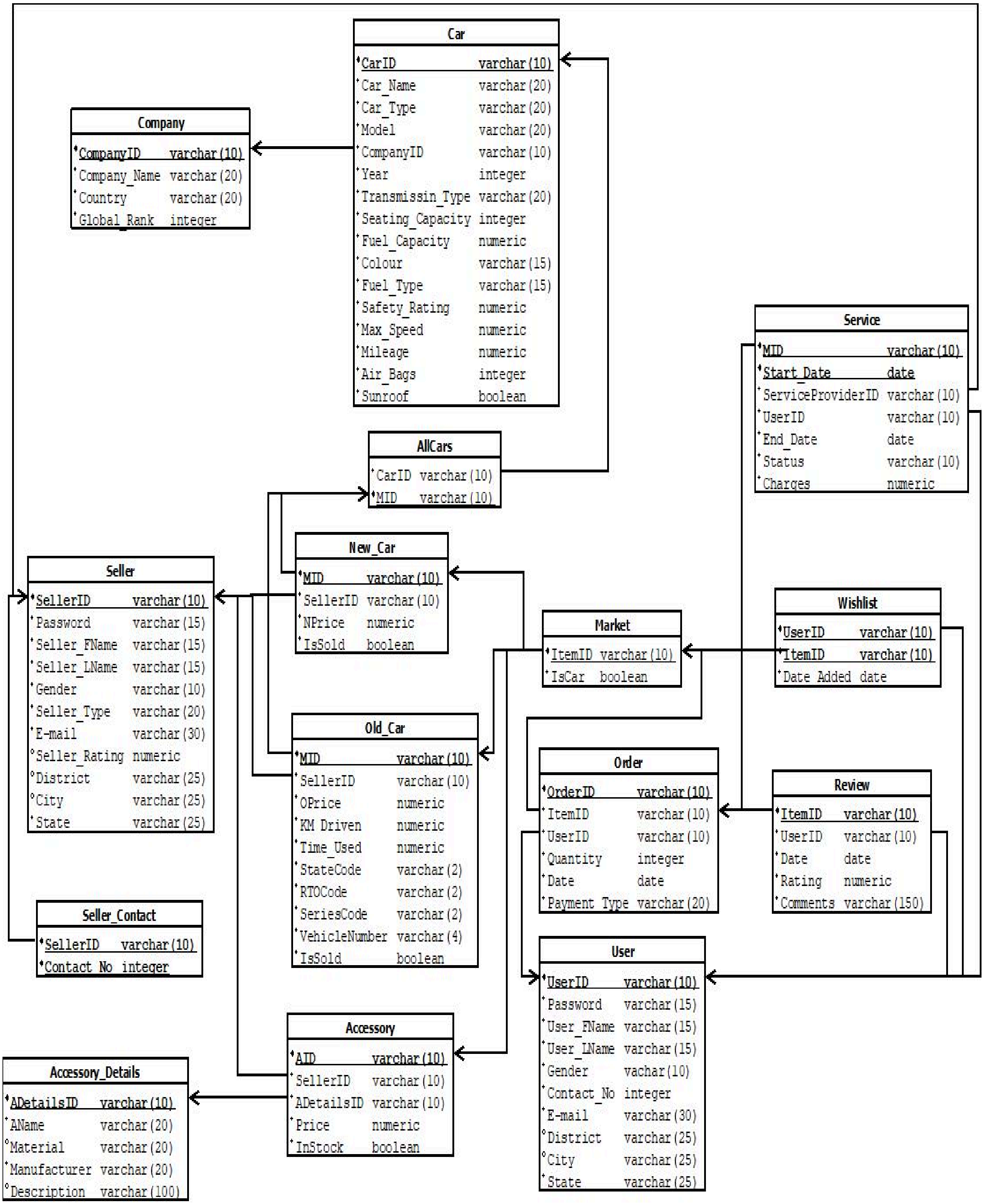
Overall, Car Bazaar delivers a realistic, end-to-end car commerce experience—from selection and purchase to after-sales service and maintenance.

Relational Schema Diagram:

We have made the Relational Schema for the ER diagram that we have submitted before, Also we have made some changes with one major change being inclusion of selling of Car Related Accessories.

If the Diagram in this Image is not completely readable, then please see this Dia file of the diagram:

Link: Link



Car

*CarID	varchar(10)
*Car_Name	varchar(20)
*Car_Type	varchar(20)
*Model	varchar(20)
*CompanyID	varchar(10)
*Year	integer
*Transmissin_Type	varchar(20)
*Seating_Capacity	integer
*Fuel_Capacity	numeric
*Colour	varchar(15)
*Fuel_Type	varchar(15)
*Safety_Rating	numeric
*Max_Speed	numeric
*Mileage	numeric
*Air_Bags	integer
*Sunroof	boolean

Company

*CompanyID	varchar(10)
*Company_Name	varchar(20)
*Country	varchar(20)
*Global_Rank	integer

Service

*MID	varchar(10)
*Start_Date	date
*ServiceProviderID	varchar(10)
*UserID	varchar(10)
*End_Date	date
*Status	varchar(10)
*Charges	numeric

AllCars

*CarID	varchar(10)
*MID	varchar(10)

New_Car

*MID	varchar(10)
*SellerID	varchar(10)
*NPrice	numeric
*IsSold	boolean

Seller

*SellerID	varchar(10)
*Password	varchar(15)
*Seller_FName	varchar(15)
*Seller_LName	varchar(15)
*Gender	varchar(10)
*Seller_Type	varchar(20)
*E-mail	varchar(30)
°Seller_Rating	numeric
°District	varchar(25)
°City	varchar(25)
*State	varchar(25)

Market

*ItemID	varchar(10)
*IsCar	boolean

Wishlist

*UserID	varchar(10)
*ItemID	varchar(10)
*Date_Added	date

Old_Car

*MID	varchar(10)
*SellerID	varchar(10)
*OPrice	numeric
*KM_Driven	numeric
*Time_Used	numeric
*StateCode	varchar(2)
*RTOCode	varchar(2)
*SeriesCode	varchar(2)
*VehicleNumber	varchar(4)
*IsSold	boolean

Order

*OrderID	varchar(10)
*ItemID	varchar(10)
*UserID	varchar(10)
*Quantity	integer
*Date	date
*Payment_Type	varchar(20)

Review

*ItemID	varchar(10)
*UserID	varchar(10)
*Date	date
*Rating	numeric
*Comments	varchar(150)

Seller_Contact

*SellerID	varchar(10)
*Contact_No	integer

User

*UserID	varchar(10)
*Password	varchar(15)
*User_FName	varchar(15)
*User_LName	varchar(15)
*Gender	vachar(10)
*Contact_No	integer
*E-mail	varchar(30)
°District	varchar(25)
°City	varchar(25)
*State	varchar(25)

Accessory

*AID	varchar(10)
*SellerID	varchar(10)
*ADetailsID	varchar(10)
*Price	numeric
*InStock	boolean

Accessory_Details

*ADetailsID	varchar(10)
*AName	varchar(20)
°Material	varchar(20)
*Manufacturer	varchar(20)
°Description	varchar(100)

Some of the Short Forms that we have used in this Report:

1) MID: Manufacturing ID
   - Just as humans have FingerPrints such that each and every fingerprint is unique, We assign MID to the Car after it is manufactured.
2) ADetailsID: Accessory Detail ID
   - It is a key for Grouping the same kind of accessories under 1 ID, so we can store the details for Identical items in one table, without any redundancy.
3) CarID:
   - It is an ID given to the Model of the car, so multiple car of same model will have same CarID, it is very helpful for us in avoiding the redundancy of storing each attribute of each car along with its MID
4) SellerID and UserID:
   - When a User logins to our Site, he/she will be asked to login either as a Seller or as a User/Buyer. So while logging in he/she will be assigned a UniqueID.
5) AID: Accessory ID
   - It is the same as MID but it is for accessories.
6) OrderID:
   - If a user buys more than 1 item in one go, then also each item will be assigned a unique orderId.

Detail View of all The Relations:

1) **Company :**

- CompanyId→companyName
- CompanyId→country
- CompanyId→Global_ rank

Key: CompanyId

About BCNF:

So from the Projected FD set it can be seen that for every non-Trivial FD, the LHS of the Functional Dependency is the SuperKey of the Relation i.e.CompanyID. So this relation is in BCNF.

2) **Car:**

- CarID → Car_Name
- CarID → Car_Type
- CarID → Model
- CarID → CompanyID
- CarID → Year
- CarID → Transmission_Type
- CarID → Seating_Capacity

- CarID → Fuel_Capacity
- CarID → Colour
- CarID → Fuel_Type
- CarID → Safety_Rating
- CarID → Max_Speed
- CarID → Mileage
- CarID → Air_Bags
- CarID → Sunroof

Key: CarID

About BCNF:

So from the Projected FD set it can be seen that for every non-Trivial FD, the LHS of the Functional Dependency is the SuperKey of the Relation i.e.CarID. So this relation is in BCNF.

3)**All Cars**:

- MID→carID

Key: MID

About BCNF:

So from the Projected FD set it can be seen that for every non-Trivial FD, the LHS of the Functional Dependency is the SuperKey of the Relation i.e.MID. So this relation is in BCNF.

4)**New_Car:**

- MID→Nprice
- MID→SellerID
- MID→IsSold

Key: MID

About BCNF:

Here also, MID is the Superkey and all other attributes can be by the Superkey i.e.MID, So this Relation also satisfies BCNF.

5) **OldCar:**

- MID → SellerID
- MID → OPrice
- MID → KM_Driven
- MID → Time_Used
- MID → StateCode
- MID → RTOCode
- MID → SeriesCode
- MID → VehicleNumber
- MID → IsSold

Key: MID

About BCNF:

Here also, MID is the Superkey and all other attributes can be by the Superkey i.e.MID, So this Relation also satisfies BCNF.

6) **Market**:

- ItemID → IsCar

Key: ItemID

About BCNF:

So ItemID beingSuperkey it determines all other attributes of the relation and also there is no other Attribute which alone can determine any other Attribute. So this relation also satisfies BCNF.

7) **Accessory**:

- AID → SellerID
- AID → ADetailsID
- AID → Price
- AID → InStock

Key: AID

About BCNF:

So using only AID we can determine other attributes and also AID is the Superkey, So this table Also satisfies BCNF.

8) **Seller**:

- SellerID → Password
- SellerID → seller_FName
- SellerID → seller_LName
- SellerID → Gender
- SellerID → seller_type
- SellerID →E-mail
- SellerID → seller_Rating
- SellerID → District
- SellerID → City
- SellerID → State

Key: SellerID

About BCNF:

SellerId is the SuperKey and Also SName is not unique, so we cannot determine any other attribute using it, Same goes with Phone and email and other Attributes.

9) **SellerContact**:

{SellerId,ContactNo} is the Composite Key. Requirement of this relation is because, Seller can have 2 or more Contact numbers.

Key: {SellerId,ContactNo}
There are only 2 Trivial FDs:

- {SellerId,ContactNo} → SellerID
- {SellerId,ContactNo} → ContactNo

There is no Non-Trivial FD, so this Relation is by default BCNF.

10) **User**:

- UserID → Password
- UserID → User_FName
- UserID → User_LName
- UserID → gender
- UserID → Contact_No
- UserID → E-mail
- UserID → District
- UserID → City
- UserID → State

Key: UserID
About BCNF:
Here also UserId is the Superkey and UName is not Unique as many users can have Same Name. So using UserID we can get all the NonTrivial FDs.So this relation is also in BCNF.

11) **AccessoriesDetail:**

- ADetailsID→ AName
- ADetailsID→ Material
- ADetailsID→ Manufacturer
- ADetailsID→ Description

Key: ADetailsID
About BCNF:
Here ADetailsID is the SuperKey, so we can determine all other attributes using it, and for any other attribute, none of them can alone determine any other attribute, So this Relation satisfies BCNF.

12) **Order**:

- OrderId→itemId
- OrderId→UserId
- OrderId→Quantity
- OrderId→Date
- OrderId→Payment_type

Key: OrderID

About BCNF:

Here OrderId is the SuperKey, so we can determine all other attributes using it, and for any other attribute, none of them can alone determine any other attribute, So this relation satisfies BCNF.

13) **Service**:

- {MID,StartDate} → ServiceProviderID
- {MID,StartDate} →UserID
- {MID,StartDate} → End_Date
- {MID,StartDate} → Status
- {MID,StartDate} → Charges

Key:{MID,StartDate}

About BCNF:

All non-trivial FDs have a superkey{MID,StartDate} as their determinant, so this Relation also Satisfies all the BCNF Constraint, hence it is in BCNF.

14) **WishList**:

- {UserID, ItemID} → Date_Added

Key: {UserID,ItemID}

About BCNF:

There is only one non-trivial FD, and for that FD the determinant is the SuperKey itself, hence this relation is also in BCNF.

15) **Review**:

Here we have assumed that a user can give review to only the items he/he has bought and once he **gives a review and wants to again give review, then his/her old Review will be updated**.

- ItemID → UserID
- ItemID → Date
- ItemID → Rating
- ItemID → Comments

Key: ItemID

About BCNF:

Here ItemID is the Superkey, so all other attributes can be determined using ItemID, and no other attribute is a candidate Key.

Minimal FD Set:

CompanyID → companyName
CompanyID → country
CompanyID → Global_rank
CarID → Car_Name
CarID → Car_Type
CarID → Model
CarID → CompanyID
CarID → Year
CarID → Transmission_Type
CarID → Seating_Capacity
CarID → Fuel_Capacity
CarID → Colour
CarID → Fuel_Type
CarID → Safety_Rating
CarID → Max_Speed
CarID → Mileage
CarID → Air_Bags
CarID → Sunroof
MID → carID
MID → Nprice
MID → SellerID
MID → IsSold
MID → OPrice
MID → KM_Driven
MID → Time_Used
MID → StateCode
MID → RTOCode
MID → SeriesCode
MID → VehicleNumber
MID → Start_Date
{MID,StartDate}→ ServiceProviderID
{MID,StartDate} → UserID
{MID,StartDate}→ End_Date
{MID,StartDate} → Status
{MID,StartDate} → Charges
ItemID → IsCar
AID → SellerID

ItemID → UserID
ItemID → Date
ItemID → Rating
ItemID → Comments
AID → ADetailsID
AID → Price
AID → InStock
SellerID → Password
SellerID → seller_FName
SellerID → seller_LName
SellerID → Gender
SellerID → seller_type
SellerID → E-mail
SellerID → seller_Rating
SellerID → District
SellerID → City
SellerID → State
UserID → Password
UserID → User_FName
UserID → User_LName
UserID → gender
UserID → Contact_No
UserID → E-mail
UserID → District
UserID → City
UserID → State
ADetailsID → AName
ADetailsID → Material
ADetailsID → Manufacturer
ADetailsID → Description
OrderID → ItemID
OrderID → UserID
OrderID → Quantity
OrderID → Date
OrderID → Payment_type
{UserID, ItemID} → Date_Added

DDL Script:

This link is for the DDL Script file: Link

Code:

```
create schema DDA;
set search_path to DDA;
create table company (
    CompanyID varchar(10) primary key,
    Company_Name varchar(20) not null unique,
    Country varchar(20)  not null,
    Global_Rank integer  not null unique
);


create table car (
    CarID varchar(10) primary key,
    Car_Name varchar(20)  not null ,
    Car_Type varchar(20)  not null ,
    Model varchar(20)  not null unique,

    CompanyID varchar(10)  not null
    references company(CompanyID)
    on update cascade on delete set null,


    Year integer  not null ,
    Transmission_Type varchar(20)  not null ,
    Seating_capacity integer  not null ,
    Fuel_Capacity numeric  not null,
    Colour varchar(15)  not null ,
    Fuel_Type varchar(15)  not null ,
    Safety_Rating numeric  not null ,
    Max_Speed numeric  not null ,
    Mileage numeric  not null ,
    Air_Bags numeric not null,
    Sunroof boolean  not null
);

create table seller (
    SellerID varchar(10) primary key,
    Password varchar(15)  not null,
    Seller_FName varchar(15)  not null ,
    Seller_LName varchar(15)  not null ,
    Gender varchar(10) not null,
    Seller_Type varchar(20) not null,
    E_mail varchar(30) not null,
    Seller_Rating numeric,
    District varchar(25),
    City varchar(25),
    State varchar(25) not null
);

create table seller_contact (
    SellerID varchar(10) references seller(SellerID) on update cascade on delete
cascade,
    Contact_No integer not null unique
);
create table availablecars (
    MID varchar(10) not null unique,
    CarID varchar(10) not null references car(CarID) on update cascade on delete set null
);
create table new_car (
    MID varchar(10) primary key references availablecars(MID) on update cascade on
delete cascade,
    SellerID varchar(10) not null references seller(SellerID) on update cascade on delete
cascade,
    NPrice numeric not null,
    IsSold boolean not null
);

create table old_car (
    MID varchar(10) primary key references availablecars(MID) on update cascade on
delete cascade,
    SellerID varchar(10) not null references seller(SellerID) on update cascade on delete
cascade,
    OPrice numeric not null,
    Kmdriven numeric not null,
    Time_Used numeric not null,
    StateCode varchar(2) not null,
    RTOCode varchar(2) not null,
    SeriesCode varchar(2) not null,
    VehicleNumber varchar(4) not null,
    IsSold boolean not null
);


create table accessory_details (
    ADetailID varchar(10) primary key,
    AName varchar(20) not null,
    Material varchar(20),
    Manufacturer varchar(20) not null,
    Description varchar(100)
);

create table accessories (
    AID varchar(10) primary key,
    ADetailID varchar(10)  not null references accessory_details(ADetailID) on update cascade on delete cascade,
    Seller_ID varchar(10) not null references seller(SellerID) on update cascade on delete set null,
    Price numeric not null,
    InStock boolean not null
    );



create table users(
    UserID varchar(10) primary key,
    Password varchar(15) not null,
    User_FName varchar(15) not null,
    User_LName varchar(15) not null,
    Gender varchar(10) not null,
    Contact_No integer not null unique,
    Age numeric not null,
    E_mail varchar(30) not null unique,
    District varchar(25) ,
    City varchar(25),
    State varchar(25) not null
);




create table market (
    ItemID varchar(10) primary key
    references new_car(MID) on update cascade on delete cascade
    references old_car(MID) on update cascade on delete cascade
    references accessories(AID) on update cascade on delete cascade,
    IsCar boolean not null
);


create table orders (
    OrderID varchar(10) primary key,
    UserID varchar(10) not null references users(UserID) on update cascade on delete cascade,
    ItemID varchar(10) not null unique references market(ItemID) on update cascade on delete cascade,
    Quantity integer not null,
    "Date" date not null,
    Payment_Type varchar(20) not null
);

create table wishlist (
    UserID varchar(10) not null unique references users(UserID) on update cascade on delete cascade,
    ItemID varchar(10) not null unique references market(ItemID) on update cascade on delete cascade,
    Date_Added date not null
);

create table review (
    ItemID varchar(10) not null unique references orders(ItemID) on update cascade on delete set null,
    UserID varchar(10) not null references users(UserID) on update cascade on delete set null,
    "Date" date not null ,
    Rating numeric not null ,
    Comments varchar(150) not null
);


create table service (
    MID varchar(10) primary key references orders(ItemID) on update cascade on delete cascade,
    UserID varchar(10) not null references users(UserID) on update cascade on delete cascade,
    ServiceProviderID varchar(10) not null,
    "Start_Date" date not null unique,
    End_Date date not null ,
    "Status" varchar(10) not null ,
    Charges numeric not null
);
```

Conclusion:

We outlined the complete relational schema and provided the DDL script for the CarBazaar DBMS. In our design, we detailed the projected functional dependencies for each relation and demonstrated that every relation complies with BCNF requirements. Finally, we derived the minimal (canonical) set of functional dependencies for the entire schema.