

Database System Project

Part3: Relational Design

Group3 : Uber

This document is a complete relational design for our Uber database system project.

Individual contributions to this assignment:

Changze Han:

- Cover sheet
- Updated description and ER model
- A complete relational schema that derives from ER model and also normalized to the 3rd normal form

Cory Skeers:

- All non-trivial functional dependencies for each relation
- Relational integrity constraints

Josh Kamp:

- Data dictionary (include domain definitions)
- Authorization decisions
- Additional Normalization Problem

Cameron Chen:

- Justification for normalized relational schema

Update ER Model:

Updated relationship names based on feedback. Also made some minor changes to the design. (e.g. change payments from weak entity to not weak entity)

Updated ER model is submitted as a separate file under “Relational Model” dropbox.

Complete Relational Schema:

Drivers (Driver ID, Driver License, Name, Address, Phone, Photo)

Cars (Plate number, Driver ID, Make, Year)

Payment account (Bank Account number, routing number, SSN, Driver ID)

Payments (Payment ID, Date, Time, Amount, Tip, Tax, Payment status)

Driver-Customer-Payments (Payment ID, Driver ID, Customer ID)

Driver Rating (Driver ID, Date, Time, Rate, Comments)

Regions (Zipcode, State, City)

Trips (Trip ID, Date, Time, Distance, Pickup location, Dropoff location, cost)

Driver-Customer-Trip (Trip ID, Customer ID, Driver ID)

Customers (Customer ID, Name, Address, Phone, Photo)

Credit cards (Customer ID, Card Number, Expiration, Code)

Customer Rating (Customer ID, Date, Time, Rate, Comments)

Accidents (Accident ID, Date, Time, Description)

Reports (Report ID, Trip ID, Date, Time, Descriptions)

Normalized to 3rd normal form:

Drivers :

Driver_R1(Driver ID, Driver License)

Driver_R2(Driver ID, Name)

Driver_R3(Driver ID, Address)

Driver_R4(Driver ID, Phone)

Driver_R5(Driver ID, Photo)

Cars:

Cars_R1(Plate number, Make)

Cars_R2(Plate number, Year)

Cars_R3(Driver ID, Plate number)

Payment account:

Payment account_R1(Driver ID, Bank account number)

Payment account_R2(Driver ID, routing number)

Payment account_R3(Driver ID, SSN)

Payments:

Payments_R1(Payment ID, Date)

Payments_R2(Payment ID, Time)

Payments_R3(Payment ID, Amount)

Payments_R4(Payment ID, Tip)

Payments_R5(Payment ID, Tax)

Payments_R6(Payment ID, Payment Status)

Payments_R7(Payment ID, Driver ID)

Driver-Customer-Payments:

Payment_R1(Trip ID, Customer ID)

Payment_R2(Trip ID, Driver ID)

Driver Rating:

Driver Rating_R1(Date, Time, Driver ID, Rate)

Driver Rating_R2(Date, Time, Driver ID, Comments)

Regions:

Regions_R1(zipcode, state)

Regions_R2(zipcode, city)

Trips:

Trips_R1(Trip ID, Date)

Trips_R2(Trip ID, Time)

Trips_R3(Trip ID, Distance)

Trips_R4(Trip ID, Pickup location)

Trips_R5(Trip ID, Drop off location)

Trips_R6(Trip ID, Cost)

Driver-Customer-Trip:

Trip_R1(Trip ID, Customer ID)

Trip_R2(Trip ID, Driver ID)

Customers:

Customers_R1(Customer ID, Name)

Customers_R2(Customer ID, Address)

Customers_R3(Customer ID, Phone)

Customers_R4(Customer ID, Photo)

Credit cards:

Credit cards_R1(Customer ID, Card number)

Credit cards_R2(Customer ID, Expiration)

Credit cards_R3(Customer ID, Code)

Customer Rating:

Customer Rating_R1(Date, Time, Customer ID, Rate)

Customer Rating_R2(Date, Time, Customer ID, Comments)

Accidents:

Accidents_R1(Accident ID, Date)
Accidents_R2(Accident ID, Time)
Accidents_R3(Accident ID, Trip ID)
Accidents_R4(Accident ID, Description)

Reports:

Reports_R1(Report ID, Date)
Reports_R1(Report ID, Time)
Reports_R1(Report ID, Trip ID)
Reports_R1(Report ID, Description)

Non-trivial functional dependencies:

Drivers

Driver ID -> Name, Address, Phone, Photo
Driver ID -> Driver License

Cars

Plate number -> Make, Year
Driver ID -> Plate number

Payment Account

Driver ID -> Bank Account Number, Routing Number, SSN

Payments

Payment ID -> Date, Time, Amount, Tip, Tax, Payment Status

Driver-Customer-Payment

Payment ID -> Driver ID, Customer ID

Regions

Zipcode -> State, City

Reports

Report ID -> Trip ID, Date, Time, Descriptions

Customers

Customer ID -> Address, Name, Phone, Photo
Customer ID -> Card Number, Expiration, Code

Accidents

Accident ID -> Date, Time, Description, Trip ID

Customer Rating

Date, Time, Customer ID -> Rate, Comments, Driver ID

Driver Rating

Date, Time, Driver ID -> Rate, Comments

Trips

Trip ID -> Date, Time, Distance, Pickup Location, Drop-off Location, Cost

Driver-Customer-Trip

Trip ID -> Customer ID, Driver ID

Relational integrity constraints:

Customer ID: Primary key/Foreign Key

Region ID: Primary key

Driver ID: Primary key/Foreign Key

Booking ID: Primary key

Report ID: Primary key

Plate Number: Primary key

Bank Account Number: Primary key

SSN: Primary key

Payment ID: Primary key/Foreign Key

Trip ID: Primary key

Date: Primary key

Time: Primary key

Tip: Default = 0 // Assume that if no tip is entered, then \$0

Pickup Location: Can't be null // Trip must have a start location

Drop-off Location: Can't be null // Trip must have a destination

Payment Status: Can't be null // Must have a status for the payment. Even an error in payment processing must be represented

Amount: Default = 0 // Can't be negative (not paying people to take the service), but potential for a discounted rate or coupon to create a 0 payment amount. Similarly, a payment error should resolve to 0, if necessary.

Justification for Relational Schema to 3NF

Above we have translated our relational schema into 3NF, by this we mean we have divided our data so that each nonprime element in the table relied on the key, the whole key, and nothing but the key. We will now give some justification for each table. We have Drivers, Cars, Payment account, Payments, Driver-Customer-Payments, Driver Rating, Regions, Trips, Driver-Customer-Trips, Customers, Credit-Cards, Customer-Rating, Accidents, and Reports. Drivers is broken down into Driver_R1 to Driver_R5. In D_R1 we relate the Driver ID and Driver License because there should only be one Driver License to a deciding Driver ID. In D_R2 we relate Driver ID and Name for a similar reason. The subcategory of Driver includes each attribute that we considered Driver in a relation with Driver ID; this works because a unique Driver ID should be able to pick out all of the other information and the other information is not dependent on each other. In the Cars subsection we have written two relations that relate the Plate Number for the car to the Make and Year of the Car. In doing this we have assumed that a particular car is defined by its Plate Number and if a car gets a new Plate Number then it is considered a new car. In Cars_R3 we also relate Driver ID to Plate Number so that we can keep track of which drivers use which cars. Payment Account is a group of relations that relate Driver ID and information vital to complete the payments like SSN and Bank Routing information. In Payments we keep track of the data related to individual payments to drivers. We used a Payment ID to identify each payment and have created relations to each individual data attribute that relates to the payment. In Driver-Customer-Payments we have a key, the Trip ID, and the Customer ID or Driver ID. This allows us to identify who are included in a particular trip. In Driver Rating we have a key made of several elements, Date, Time, and Driver ID, this lets us keep track of not just a driver's most recent rating but past ones. The nonprime elements are the Rate and the Comments. We have used the key of ZipCode to determine the elements in the Region section such as State and City, in doing this we are working with the assumption that no Zipcodes are ever repeated throughout differing States or Cities. In the Trips section we have created a Trip ID that identifies each piece of information about a particular trip in our model. We also use the Trip ID to pick out both the Driver and Customer IDs for our recorded trips. Similar to the prior Payment Account section we have devised a Credit Card section that uses Customer ID to pick out information about the customer's card. In our model we only allow a customer to track a single credit card this way. Finally we have sections for Accidents and Reports. The Accidents use an Accident ID which is the unique identifier that is assigned to an accident to pick out information on that accident. The Report section works similarly with an assigned Report ID. In making our relational schema follow the rules of 3NF we were careful to separate and identify the nonprime elements and ensure that they did not decide other nonprime elements.

Data Dictionary

Data Item	Definition	Data Type	Notes
Customer ID	Unique ID assigned to customer when they make an account	Int	
Payment Method	Method customer uses to pay (credit/debit/cash)	Enum	
Region ID	Unique ID assigned to different locations, usually cities/towns	Int	
Driver ID	Unique ID number assigned to drivers when they become a driver	Int	
Paycheck account	Account routing number	Int	Needs to be encrypted
Regional Rate	Cost per minute for the given area. Will depend on factors like time of day, number of available drivers, number of customer requests, and customer battery percentage.	Float	
Pickup location	Address where customer wants to be picked up.	String	

Drop-off location	Address where customer wants to be dropped off.	String	
Booking ID	Unique number assigned to a booking when the customer submits their request	Int	
Transaction Type	Method of payment (pay with cash/credit/etc)	String	
Rating	Customer can rate their driver a number out of 5, with 5 being great and 0 being terrible, after they are dropped off.	Float	Will be used to determine if a driver needs to be disciplined or get a raise.
Accident Type	Category of accident, animal/car/Uber driver's fault/other driver's fault.	Enum	
Accident description	Brief paragraph describing what happened, whose fault it was, if anyone was injured, and any other relevant information.	String	
Report ID	Unique number assigned when Customer or driver submits a report	Int	

Report type	The type of report, harassment, rule breaking, etc.	Enum	
Report description	A brief paragraph that describes what happened and what rules were broken.	String	

Authorization Decisions

User role	Privilege (read, insert, update, delete)
Customer	Customers: Update(Address, payment method, phone number), Read Drivers: None Tax Info: None Regions: None Trips: Update(pickup, dropoff, date, time), Read Payments: Update(transaction type), Read Ratings: Insert, Update(Rating) Accidents: None Reports: Insert, Update(report type, report description) Credit Cards: Insert, Update
Driver	Customers: Read(name, address, phone number) Drivers: Read, Update(address, name, paycheck account) Tax Info: Read, Update(withholdings status) Regions: None Trips: None Payments: Read Ratings: Read Accidents: Insert, Update(accident description) Reports: Insert, Update(report description) Credit Cards: None
Manager	All for All

Additional normalization problem:

All attributes:

R(Driver ID, D_Name, D_Address, D_Phone, D_Photo, Driver License, Plate number, Make, Year, Bank Account Number, Routing Number, SSN, Payment ID, Payment_Date, Payment_Time, Payment_Amount, Tip, Tax, Payment Status, Zip-code, State, City, Report ID, Trip ID, Report_Date, Report_Time, Report_Descriptions, Customer ID, C_Address, C_Name, C_Phone, C_Photo, Card Number, Expiration, Code, Accident ID, Accident_Date, Accident_Time, Accident_Description, Trip_Date, Trip_Time, Rate, Comments, Distance, Pickup Location, Drop-off Location, Cost)

Derive :

R1(Driver ID, D_Name, D_Address, D_Phone, D_Photo)

R2(Driver ID, Driver License)

R3(Plate number, Make, Year)

R4(Driver ID, Plate number)

R5(Driver ID, Bank Account Number, Routing Number, SSN)

R6(Payment ID, Payment_Date, Payment_Time, Payment_Amount, Tip, Tax, Payment Status, Driver ID)

R7(Driver ID, Payment_Date, Payment_Time, Payment ID, Payment_Amount, Tip, Tax, Payment Status)

R8(Zip-code, State, City)

R9(Report ID, Trip ID, Report_Date, Report_Time, Report_Descriptions, Customer ID, Driver ID)

R10(Customer ID, C_Address, C_Name, C_Phone, C_Photo)

R11(Customer ID, Card Number, Expiration, Code)

R12(Accident ID, Accident_Date, Accident_Time, Accident_Description, Customer ID, Driver ID)

R13(Trip_Date, Trip_Time, Customer ID, Rate, Comments, Driver ID)

R14(Trip_Date, Trip_Time, Driver ID, Rate, Comments)

R15(Trip ID, Trip_Date, Trip_Time, Distance, Pickup Location, Drop-off Location, Cost, Driver ID, Customer ID)

R16(Customer ID, Trip_Date, Trip_Time, Trip ID, Distance, Pickup Location, Drop-off Location, Cost, Driver ID)

R17(Driver ID, Trip_Date, Trip_Time, Trip ID, Distance, Pickup Location, Drop-off Location, Cost, Customer ID)