



SYMBIOSIS INSTITUTE OF TECHNOLOGY (SIT)

Constituent of Symbiosis International (Deemed University), Pune

(Established under Section 3 of the UGC Act of 1956 vide notification number F-9-12/2001-U-3 of the Government of India)
Re-Accredited by NAAC with 'A' Grade

PROJECT PHASE 2
A DBMS Project Report on
OLA CABS DATABASE



Submitted by
Sakshi Joshi(20070122199)
Pratham Suryawanshi(20070122201)
Akshansh Sourabh(20070122202)
Aditya Khaladkar(21070122509)

UNDER THE GUIDANCE OF
Dr. Pooja Bagane

**Department of Computer Science
SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE**

Table of Contents



- Introduction
- Functional Requirements
- Anomalies of Relation Schema
- Functional Dependency of Schema
- Normalization of Anomalies
- Table Implementation
- Execution of queries, functions procedures and triggers



Introduction

Ola is India's largest mobility platform and one of the world's largest ride-hailing companies, serving more than 250 cities in India, Australia, New Zealand, and the United Kingdom. The Ola app connects customers to drivers and a wide range of vehicles, including bikes, auto-rickshaws, metered taxis, and cabs, allowing hundreds of millions of consumers and over 1.5 million driver-partners convenience and transparency.

This service necessitates a massive amount of data to run smoothly, necessitating the use of a Database Management System. The Ola Cab database organises all the necessary information about the customer who books the ride, the driver partners, employees, vehicles and booking. The correct source and destination information, driver and customer names and contact information, fare price, OTP, booking id (CRN number), and vehicle details are essential for the effective operation of the Ola Cab application and constitute the foundation of the database.

Problem Statement

Through our database, where we will keep the information of the passenger who books the journey, the driver partners, staff, vehicles, and booking, we want to comprehend Ola Cabs' data organization issues and present a clear image of their data and services. Earlier, it was fairly challenging to hunt out for appropriate transportation when we were in a rush and at a reasonable rate. Ola eliminates this issue and links you to what is best for you in no time. Additionally, it helped fleet operators do business and improved drivers' earnings by making it simple for them to reach customers. The goal of our project is to build a database of application that address this issue and comprehend it at a fundamental level.

Anomalies of Relation Schema

An anomaly occurs when we have redundant data in our database. There are 3 types of anomalies that can occur in a database – Insertion Anomalies, Deletion Anomalies and Update Anomalies. Our database features all 3 anomalies.

We have a table called Customer. It contains the basic information about a “Customer” who is availing Ola services. A customer books rides with a cab or an autorickshaw. Also, all users have a key called UserID. Thus, all Customers are users but not all users are customers. To connect a UserID to a Customer, we have created a table called customer_user which contains 2 columns – CustomerID and UserID. Both columns are foreign keys taken from tables Customer and User respectively.

	CustomerID	UserID
▶	C0000000001	1
	C0000000002	2
	C0000000003	3
	C0000000004	4
	C0000000005	5

Customer_User Table

	customerID	ProfileType	EmailId	OlaWallet_Amount
▶	C0000000001	P	akulz123@gmail.com	0
	C0000000002	N	yajwanbabu@gmail.com	822
	C0000000003	P	ccentina455@gmail.com	390
	C0000000004	N	devi.vishare134@gmail.com	1500
	C0000000005	N	ethanhuntMI@gmail.com	300

Customer Table (Without UserID)

When Ola inserts values in the Customer table, an anomaly occurs. The customer table will have a record of the new customer with his/her own CustomerID, but it won't reflect in customer_user table. Thus, the customer will not be assigned a UserID. The opposite can also be true. This is an example of an Insertion Anomaly in our database.

Now, if for some reason a customer decides to delete his or her account, or Ola decides to terminate accounts of users who haven't been active for some time, an anomaly occurs. Depending on which table these delete operations are performed, the other table will still have the customer records (which are supposed to be deleted!). This shows Deletion Anomaly in our database.

Anomalies of Relation Schema

These 2 anomalies can be removed by merging the UserID column into the Customer table and make assigning a UserID value to a new Customer compulsory (It cannot be NULL and should be UNIQUE). Also this way, any deletion will delete both the CustomerID and UserID from the database. However, we still have another table called Users which has the UserID. A trigger can be setup for deletion in the Customer table or the User table! This trigger should trigger deletion of the record with the same UserID in the other table.

Now coming to the Update Anomaly.

We have these 4 tables (as shown below) - Vehicle, Driver, Books and Trip

	VehicleNo	OperatorID	Type	Company	Model	Color	ChassiNumber	RCNumber	InsuranceNumber
▶	DL2C 0070	F000000002	PV	Chevrolet	Enjoy	Silver	CN202205	RC202205	ISN202205
	KA01 1975	F000000004	PP	Toyota	Etios	Silver	CN202206	RC202206	ISN202206
	MH12 5815	F000000001	MI	Nissan	Micra	Silver	CN202202	RC202202	ISN202202
	MH13 1602	F000000001	PS	Maruti	Swift-Dzire	Silver	CN202203	RC202203	ISN202203
	MH13 5005	F000000001	AT	Bajaj	Compact RE	Yellow-Green	CN202201	RC202201	ISN202201
	MH47 8820	F000000003	AT	Mahindra	Alpha	Black-Yellow	CN202204	RC202204	ISN202204
●	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Vehicle Table

	DriverID	PartnerID	VehicleNo	OperatorID	LicenseNo	Rating	Status
▶	DR00000001	P000000001	MH13 5005	F000000001	LIN34625	4.6	ON
	DR00000002	P000000002	MH12 5815	F000000001	LIN95672	3.7	ON
	DR00000003	P000000004	MH13 1602	NULL	LIN56724	4.8	OFF
	DR00000004	P000000005	MH47 8820	F000000003	LIN59157	4.4	OFF
	DR00000005	P000000006	DL2C 0070	F000000002	LIN56242	4.6	ON
	DR00000006	P000000009	KA01 1975	F000000004	LIN87641	4.8	ON
●	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Driver Table

	BookingID	CustomerID	StartAddress	DestinationAddress	RideDate	RideTime	Preferred_Vehicle_Type	RideType	Approximate_Cost	BookingStatus
▶	BK000000001	C000000001	https://goo.gl/maps/fWtIX3pmCDr94kw8	https://goo.gl/maps/73aMH3LAz1ZJArV7	2022-05-25	16:00:00	AT	TX	245	OF
	BK000000002	C000000001	https://goo.gl/maps/aiH4AbarFgMW214c3A	https://g.page/SymbiosisEngineering?share	2022-06-18	16:45:00	MI	TX	162	OF
	BK000000003	C000000001	https://goo.gl/maps/vueHt8bjArpezfj7	https://g.page/BrowniePointMahim?share	2022-06-19	16:50:00	AT	TX	189	OF
	BK000000004	C000000002	https://g.page/SymbiosisEngineering?share	https://goo.gl/maps/hYrGC8aSEy4fChdMA	2022-09-02	09:30:00	PS	TX	345	OF
	BK000000005	C000000002	https://g.page/DEHICLUBHOUSE?share	https://goo.gl/maps/WACPSF2C8dGALDA9	2022-10-13	06:00:00	PV	OS	412	OF
	BK000000006	C000000003	https://goo.gl/maps/Q2n1GNthyHNcA4hQ8	https://goo.gl/maps/SKY5N2WyWstU75vW76	2022-10-20	18:40:00	AT	TX	299	OF
	BK000000007	C000000003	https://goo.gl/maps/dhrnVYDp6P5VKA	https://goo.gl/maps/z2WVWNgSC5bKaUz9	2022-10-22	19:00:00	AT	TX	150	OF
	BK000000008	C000000004	https://goo.gl/maps/rVvtvDZclJwRhf7	https://goo.gl/maps/SKY5N2WyWstU75vW76	2022-11-02	05:30:00	PS	OS	99	OF
	BK000000009	C000000004	https://goo.gl/maps/RH16BcdEpvHjYi8	https://goo.gl/maps/jk54oQvY6kWhmyM8	2022-11-11	05:30:00	PP	OS	190	OF
	BK000000010	C000000005	https://g.page/apricot-service-apartments?share	https://g.page/SymbiosisEngineering?share	2022-10-30	12:35:00	AT	TX	211	OF
	BK000000011	C000000005	https://goo.gl/maps/eUjM83ELxejck8	https://goo.gl/maps/AqPvvhNdtUxcsGsva9	2022-11-04	14:00:00	MI	TX	120	OF
	BK000000012	C000000005	https://goo.gl/maps/hYrGC8aSEy4fChdMA	https://goo.gl/maps/73aMH3LAz1ZJArV7	2022-11-08	13:30:00	AT	TX	79	CP
●	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Books Table

Anomalies of Relation Schema

	CRN	BookingID	DriverID	VehicleType	PaymentID	FinalCost	Status
▶	6789343901	BK00000001	DR00000001	AT	PAY0000001	250	RD
	6789343902	BK00000002	DR00000002	MI	PAY0000002	160	RD
	6789343903	BK00000003	DR00000004	AT	NULL	0	CC
	6789343904	BK00000004	DR00000003	PS	PAY0000003	345	RD
	6789343905	BK00000005	DR00000005	PV	PAY0000004	410	RD
	6789343906	BK00000006	DR00000001	AT	PAY0000005	300	RD
	6789343907	BK00000007	DR00000004	AT	PAY0000006	150	RD
	6789343909	BK00000009	DR00000006	PP	NULL	190	PP
	6789343910	BK00000010	DR00000001	AT	PAY0000008	210	RD
	6789343911	BK00000011	DR00000002	MI	PAY0000009	120	RD
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Books Table

A Trip record is only formed when a driver confirms a booking made by a customer. In other words, when the BookingStatus goes from CP (Confirmation Pending) to CF (Confirmed). Lets say that a customer books a trip on a date 2 weeks from today. A driver 'A' confirms the trip.

A trip record is created and the vehicle type is set. Sometime in between these 2 weeks, the driver gets a new car.

The new cars type is different from the type of the drivers previously driven vehicle. Thus, the driver table is updated and the vehicle number is changed. But this change is not reflected in the Trip table! The VehicleType in the Trip table is still the same type as that of the vehicle that was previously owned by the driver.

This causes an Update Anomaly in our database!

To resolve this anomaly, we can create a trigger on the Driver table. When there is an update to the vehicle number, we will update VehicleType for all PENDING trips of the driver. This way, an update of the vehicle will allow updation of VehicleType for the pending trips too, thus, resolving the anomalies!

Functional Dependency

	VehicleNo	OperatorID	Type	Company	Model	Color	ChassiNumber	RCNumber	InsuranceNumber
▶	DL2C 0070	F000000002	PV	Chevrolet	Enjoy	Silver	CN202205	RC202205	ISN202205
	KA01 1975	F000000004	PP	Toyota	Etios	Silver	CN202206	RC202206	ISN202206
	MH12 5815	F000000001	MI	Nissan	Micra	Silver	CN202202	RC202202	ISN202202
	MH13 1602	F000000001	PS	Maruti	Swift-Dzire	Silver	CN202203	RC202203	ISN202203
	MH13 5005	F000000001	AT	Bajaj	Compact RE	Yellow-Green	CN202201	RC202201	ISN202201
	MH47 8820	F000000003	AT	Mahindra	Alpha	Black-Yellow	CN202204	RC202204	ISN202204
●	NULL	NULL	HULL	HULL	NULL	HULL	HULL	HULL	HULL

In this table, Company, Model, ChassiNumber and Color are independent of each other but dependent on VehicleNo. In this table, these four columns are said to be multivalue dependent on VehicleNo. This dependence can be represented like this:

VehicleNo → Model

VehicleNo → Color

VehicleNo → Company

VehicleNo → ChassiNumber

	CRN	BookingID	DriverID	VehicleType	PaymentID	FinalCost	Status
▶	6789343901	BK00000001	DR00000001	AT	PAY0000001	250	RD
	6789343902	BK00000002	DR00000002	MI	PAY0000002	160	RD
	6789343903	BK00000003	DR00000004	AT	NULL	0	CC
	6789343904	BK00000004	DR00000003	PS	PAY0000003	345	RD
	6789343905	BK00000005	DR00000005	PV	PAY0000004	410	RD
	6789343906	BK00000006	DR00000001	AT	PAY0000005	300	RD
	6789343907	BK00000007	DR00000004	AT	PAY0000006	150	RD
	6789343909	BK00000009	DR00000006	PP	NULL	190	PP
	6789343910	BK00000010	DR00000001	AT	PAY0000008	210	RD
	6789343911	BK00000011	DR00000002	MI	PAY0000009	120	RD
●	NULL	NULL	NULL	NULL	NULL	NULL	NULL

In this table, FinalCost, VehicleType, and PaymentID are independent of each other but dependent on CRN. In this table, these three columns are said to be multivalue dependent on CRN. This dependence can be represented like this:

CRN → VehicleType

CRN → FinalCost

CRN → PaymentID

	PaymentID	PaymentType
▶	PAY0000001	CH
	PAY0000002	CH
	PAY0000003	UP
	PAY0000004	UP
	PAY0000005	OW
	PAY0000006	OW
	PAY0000007	CD
	PAY0000008	CH
●	PAY0000009	CH
	NULL	NULL

Consider this table with two columns PaymentID and PaymentType.

{PaymentID, PaymentType} \rightarrow PaymentID is a trivial functional dependency as PaymentID is a subset of {PaymentID, PaymentType}.

	VehicleNo	OperatorID	Type	Company	Model	Color	ChassiNumber	RCNumber	InsuranceNumber
▶	DL2C 0070	F000000002	PV	Chevrolet	Enjoy	Silver	CN202205	RC202205	ISN202205
	KA01 1975	F000000004	PP	Toyota	Etios	Silver	CN202206	RC202206	ISN202206
	MH12 5815	F000000001	MI	Nissan	Micra	Silver	CN202202	RC202202	ISN202202
	MH13 1602	F000000001	PS	Maruti	Swift-Dzire	Silver	CN202203	RC202203	ISN202203
	MH13 5005	F000000001	AT	Bajaj	Compact RE	Yellow-Green	CN202201	RC202201	ISN202201
	MH47 8820	F000000003	AT	Mahindra	Alpha	Black-Yellow	CN202204	RC202204	ISN202204
●	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

{VehicleNo} \rightarrow {RCNumber} (if we know the vehicleNo, we know the RCNumber)

But RCNumber is not a subset of VehicleNo, and hence it's non-trivial functional dependency.

	VehicleNo	OperatorID	Type	Company	Model	Color	ChassiNumber	RCNumber	InsuranceNumber
▶	DL2C 0070	F000000002	PV	Chevrolet	Enjoy	Silver	CN202205	RC202205	ISN202205
	KA01 1975	F000000004	PP	Toyota	Etios	Silver	CN202206	RC202206	ISN202206
	MH12 5815	F000000001	MI	Nissan	Micra	Silver	CN202202	RC202202	ISN202202
	MH13 1602	F000000001	PS	Maruti	Swift-Dzire	Silver	CN202203	RC202203	ISN202203
	MH13 5005	F000000001	AT	Bajaj	Compact RE	Yellow-Green	CN202201	RC202201	ISN202201
	MH47 8820	F000000003	AT	Mahindra	Alpha	Black-Yellow	CN202204	RC202204	ISN202204
●	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

{VehicleNo} \rightarrow {RCNumber} (if we know the vehicleNo, we know the RCNumber)

{RCNumber} \rightarrow {InsuranceNumber} If we know the RCNumber, we know the InsuranceNumber Therefore according to the rule of transitive dependency:

{VehicleNo} \rightarrow {InsuranceNumber} should hold, that makes sense because if we know the VehicleNo, we can know his InsuranceNumber.

Normalization of Anomalies

Normalization: 1NF

CustomerID	ContactNumber
C0000000001	9112306148, 9446205053
C0000000002	6448029688, 8660480269
C0000000003	8660795058, 9556128020
C0000000004	6065556880
C0000000005	8866889804

In the above table, there are ContactNumber as multivalued attributes, So we rearranged it using 1NF.

	CustomerID	ContactNumber
▶	C0000000001	9112306148
	C0000000001	9446205053
	C0000000002	6448029668
	C0000000002	8660480269
	C0000000003	8660795058
	C0000000003	9556128020
	C0000000004	6065556880
	C0000000005	8866889804
*	NULl	NULl

Normalization of Anomalies

2. Normalization: Boyce Codd Normal Form

	CRN	BookingID	DriverID	VehicleType	PaymentID	FinalCost	Status
▶	6789343901	BK00000001	DR00000001	AT	PAY0000001	250	RD
	6789343902	BK00000002	DR00000002	MI	PAY0000002	160	RD
	6789343903	BK00000003	DR00000004	AT	NULL	0	CC
	6789343904	BK00000004	DR00000003	PS	PAY0000003	345	RD
	6789343905	BK00000005	DR00000005	PV	PAY0000004	410	RD
	6789343906	BK00000006	DR00000001	AT	PAY0000005	300	RD
	6789343907	BK00000007	DR00000004	AT	PAY0000006	150	RD
	6789343909	BK00000009	DR00000006	PP	NULL	190	PP
	6789343910	BK00000010	DR00000001	AT	PAY0000008	210	RD
	6789343911	BK00000011	DR00000002	MI	PAY0000009	120	RD
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	CRN	BookingID	DriverID	VehicleType	PaymentID
▶	6789343901	BK00000001	DR00000001	AT	PAY0000001
	6789343902	BK00000002	DR00000002	MI	PAY0000002
	6789343903	BK00000003	DR00000004	AT	NULL
	6789343904	BK00000004	DR00000003	PS	PAY0000003
	6789343905	BK00000005	DR00000005	PV	PAY0000004
	6789343906	BK00000006	DR00000001	AT	PAY0000005
	6789343907	BK00000007	DR00000004	AT	PAY0000006
	6789343909	BK00000009	DR00000006	PP	NULL
	6789343910	BK00000010	DR00000001	AT	PAY0000008
	6789343911	BK00000011	DR00000002	MI	PAY0000009
*	NULL	NULL	NULL	NULL	NULL

PaymentID	FinalCost	Status
PAY0000001	250	RD
PAY0000002	160	RD
NULL	0	CC
PAY0000003	345	RD
PAY0000004	410	RD
PAY0000005	300	RD
PAY0000006	150	RD
NULL	190	PP
PAY0000008	210	RD
PAY0000009	120	RD
NULL	NULL	NULL

Normalization of Anomalies

Normalization: 2NF

	PartnerID	UserID	DateOfJoining	PLang	AccHolderName
▶	P000000001	6	2022-11-23	EN	Sarfaraz Kapoor
	P000000002	P000000002	2022-02-11	HN	Gabriel McKenzee
	P000000003	8	2022-01-10	MT	Harish Chatarjee
	P000000004	9	2022-03-22	MT	Ishaan V Kulkarni
	P000000005	10	2021-12-23	EN	Jyn Russo
	P000000006	11	2020-09-19	HN	Kareen Abdul
	P000000007	12	2022-01-05	EN	Luke Skywalker
	P000000008	13	2022-04-18	MT	Manish Paranzpe
	P000000009	14	2021-08-24	HN	Nikita Kumar
	P000000010	15	2021-02-25	HN	Osama Raj
*	HULL	HULL	HULL	HULL	HULL

PartnerID	VehicleNo	OperatorID	LicenseNo	Rating	Status
P000000001	MH13 5005	F000000001	LIN34625	4.6	ON
P000000002	MH12 5815	F000000001	LIN95672	3.7	ON
P000000004	MH13 1602	HULL	LIN56724	4.8	OFF
P000000005	MH47 8820	F000000003	LIN59157	4.4	OFF
P000000006	DL2C 0070	F000000002	LIN56242	4.6	ON
P000000009	KA01 1975	F000000004	LIN87641	4.8	ON
HULL	HULL	HULL	HULL	HULL	HULL

	VehicleNo	OperatorID	Type	Company	Model	Color
▶	DL2C 0070	F000000002	PV	Chevrolet	Enjoy	Silver
	KA01 1975	F000000004	PP	Toyota	Etios	Silver
	MH12 5815	F000000001	MI	Nissan	Micra	Silver
	MH13 1602	F000000001	PS	Maruti	Swift-Dzire	Silver
	MH13 5005	F000000001	AT	Bajaj	Compact RE	Yellow-Green
	MH47 8820	F000000003	AT	Mahindra	Alpha	Black-Yellow
-	HULL	HULL	HULL	HULL	HULL	HULL

Normalization of Anomalies

Normalization: 2NF

	BookingID	CustomerID	StartAddress	DestinationAddress	RideDate	RideTime
▶	BK000000001	C0000000001	https://goo.gl/maps/fWt1X3pmCDr9i4Kw8	https://goo.gl/maps/73aMH3LAz1ZXJArV7	2022-05-25	16:00:00
	BK000000002	C0000000001	https://goo.gl/maps/aH4AbanFgMW214c3A	https://g.page/SymbiosisEngineering?share	2022-06-18	16:45:00
	BK000000003	C0000000001	https://goo.gl/maps/wueHr8bjkArpezk7	https://g.page/BrowniePointMahim?share	2022-06-19	16:50:00
	BK000000004	C0000000002	https://g.page/SymbiosisEngineering?share	https://goo.gl/maps/hVrGC8aSEy4fChcMA	2022-09-02	08:30:00
	BK000000005	C0000000002	https://g.page/DELHICLUBHOUSE?share	https://goo.gl/maps/WACP5F2C8dGALDAd9	2022-10-13	06:00:00
	BK000000006	C0000000003	https://goo.gl/maps/QZn1GNthyHNcA4Hq8	https://goo.gl/maps/SKY5N2WyWsU75vW76	2022-10-20	18:40:00
	BK000000007	C0000000003	https://goo.gl/maps/dhpnvYDYaxp6P5VKA	https://goo.gl/maps/t2WrWNgSCSbKAuZ9	2022-10-22	19:00:00
	BK000000008	C0000000004	https://goo.gl/maps/rVvtvDVZcUarRhf7	https://goo.gl/maps/SKY5N2WyWsU75vW76	2022-11-02	05:30:00
	BK000000009	C0000000004	https://goo.gl/maps/RH16BcdkEpvrHjYi8	https://goo.gl/maps/jk54qQvy8kWhnyM8	2022-11-11	05:30:00
	BK000000010	C0000000005	https://g.page/apricot-service-apartments?share	https://g.page/SymbiosisEngineering?share	2022-10-30	12:35:00
	BK000000011	C0000000005	https://goo.gl/maps/eUjfM8XELXecJck8	https://goo.gl/maps/AqPvvNNzNJxcGSva9	2022-11-04	14:00:00
	BK000000012	C0000000005	https://goo.gl/maps/hVrGC8aSEy4fChcMA	https://goo.gl/maps/73aMH3LAz1ZXJArV7	2022-11-08	13:30:00
*	NULL	NULL	NULL	NULL	NULL	NULL

BookingID	DriverID	VehidleType	PaymentID	FinalCost	Status
BK000000001	DR00000001	AT	PAY00000001	250	RD
BK000000002	DR00000002	MI	PAY00000002	160	RD
BK000000003	DR00000004	AT	NULL	0	CC
BK000000004	DR00000003	PS	PAY00000003	345	RD
BK000000005	DR00000005	PV	PAY00000004	410	RD
BK000000006	DR00000001	AT	PAY00000005	300	RD
BK000000007	DR00000004	AT	PAY00000006	150	RD
BK000000009	DR00000006	PP	NULL	190	PP
BK000000010	DR00000001	AT	PAY00000008	210	RD
BK000000011	DR00000002	MI	PAY00000009	120	RD
NULL	NULL	NULL	NULL	NULL	NULL

	PaymentID	PaymentType
▶	PAY0000001	CH
	PAY0000002	CH
	PAY0000003	UP
	PAY0000004	UP
	PAY0000005	OW
	PAY0000006	OW
	PAY0000008	CD
	PAY0000009	CH
*	NULL	NULL

Table Implementation

	UserID	FirstName	LastName	PhoneNumber	DOB	Password
▶	1	Akul	Zameendar	9112348090	2000-05-19	*****
	2	Babu	Yajwan	8998685910	1997-02-01	*****
	3	Clentina	Waghmare	9668610050	1990-10-04	*****
	4	Devi	Vishare	8660807050	2002-12-21	*****
	5	Ethan	Hunt	9668012345	1994-08-30	*****
	6	Fatima	Kapoor	9847832293	1990-04-04	*****
	7	Gabriel	McKenzee	8661552505	1995-03-17	*****
	8	Harish	Chatarjee	6880492580	1978-06-23	*****
	9	Ishaan	Kulkarni	8622928850	1990-08-19	*****
	10	Jyn	Russo	8098553109	1987-05-30	*****
	11	Kareen	Abdul	94480152205	1991-04-01	*****
	12	Luke	Skywalker	7559050179	1993-10-16	*****
	13	Manish	Paranzpe	9660608042	1986-08-21	*****
	14	Nikita	Kumar	7980808050	1995-06-13	*****
	15	Osama	Raj	9005065018	1989-11-29	*****
*	HULL	HULL	HULL	HULL	HULL	HULL

User

	CustomerID	UserID	ProfileType	EmailId	OlaWallet_Amount
▶	C0000000001	1	P	akulz123@gmail.com	0
	C0000000002	2	N	yajwanbabu@gmail.com	822
	C0000000003	3	P	cdentina455@gmail.com	390
	C0000000004	4	N	devi.vishare134@gmail.com	1500
*	C0000000005	5	N	ethanhuntMI@gmail.com	200
*	HULL	HULL	HULL	HULL	HULL

Customer

	CustomerID	CardHolderName	CardNumber	CVV	ExpiryDate
▶	C0000000001	Akul Z	1520 8105 9802 5550	629	05/30
	C0000000002	Babu Yajwan	8098 6504 52330 5086	88	06/26
	C0000000003	Clentina Waghmare	0658 9852 1561 2608	988	12/28
	C0000000004	Devi Vishare	6801 5050 4098 9871	123	03/29
	C0000000004	Devi V	6804 9845 0288 9556	465	01/31
	C0000000005	Ethan Hunt	6480 8122 2348 3580	8	06/26
	HULL	HULL	HULL	HULL	HULL

Customer Card Details

Table Implementation

	CustomerID	ContactNumber
▶	C0000000001	9112306148
	C0000000001	9446205053
	C0000000002	6448029668
	C0000000002	8660480269
	C0000000003	8660795058
	C0000000003	9556128020
	C0000000004	6065556880
	C0000000005	8866889804
●	NULL	NULL

Customer Emergency Contact Number

	CustomerID	Location
▶	C0000000001	https://goo.gl/maps/1zP1ANdtq1fRcGQw7
	C0000000001	https://goo.gl/maps/47EgFkTPzEo4ATMT9
	C0000000001	https://goo.gl/maps/XqTjpqeN9NyvoEHfA
	C0000000002	https://goo.gl/maps/2n1d7i6a2Avw5mm9A
	C0000000002	https://goo.gl/maps/3qag5CxKSJTzJ1Q38
	C0000000002	https://goo.gl/maps/sgSfxzLkNavUwpnK6
	C0000000003	https://goo.gl/maps/heyty5qVhuf3tHD56
	C0000000003	https://goo.gl/maps/uuBheMHhWHRFvCmy7
	C0000000004	https://goo.gl/maps/afrWYRMbHzEkMcw38
	C0000000004	https://goo.gl/maps/Uh8UJtxkeR58xyBU6
●	NULL	NULL

Customer Favorite Location

Table Implementation

	CustomerID	UPIID
▶	C0000000001	akul123@okhdfcbank
	C0000000002	babuy15@okhdfcbank
	C0000000003	denty44@okicicibank
	C0000000004	devdev909@okyesbank
	C0000000005	ethanmi6@okaxisbank
●	NULL	NULL

Customer UPI Details

	PartnerID	UserID	DateOfJoining	PLang	AccHolderName	BankAc	Street	City	State	PinCode	PanCardNo	AadharCardNo
▶	P000000001	6	2022-11-23	EN	Sarfraz Kapoor	SBI200708	Baner	Pune	Maharashtra	411021	N20156	5829 3294 4921
	P000000002	P000000002	2022-02-11	HN	Gabriel McKenzee	SBI211202	Aundh Baner	Pune	Maharashtra	411007	N63421	5724 2472 9516
	P000000003	8	2022-01-10	MT	Harish Chatarjee	SBI299102	Balewadi	Pune	Maharashtra	411045	N53487	5692 4867 9862
	P000000004	9	2022-03-22	MT	Ishaan V Kulkarni	SBI233109	Kalyani Nagar	Pune	Maharashtra	411014	N56581	7184 8767 9162
	P000000005	10	2021-12-23	EN	Jyn Russo	SBI201709	Goregaon	Mumbai	Maharashtra	210021	N73666	6692 9274 8966
	P000000006	11	2020-09-19	HN	Kareen Abdul	SBI287106	RK Puram	New Delhi	Delhi	110211	N58912	5491 4867 9424
	P000000007	12	2022-01-05	EN	Luke Skywalker	SBI293105	Azadpur	New Delhi	Delhi	110033	N65483	7292 6667 8765
	P000000008	13	2022-04-18	MT	Manish Paranzpe	SBI274309	Mahim	Mumbai	Maharashtra	400016	N95487	9876 5432 1234
	P000000009	14	2021-08-24	HN	Nikita Kumar	SBI288602	Brigade Road	Bangalore	Karnataka	560001	N42165	4321 9876 1198
	P000000010	15	2021-02-25	HN	Osama Raj	SBI205107	Adugodi	Bangalore	Karnataka	560030	N77248	5678 1234 1020
●	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Partner

	VehicleNo	OperatorID	Type	Company	Model	Color	ChassiNumber	RCNumber	InsuranceNumber
▶	DL2C 0070	F00000002	PV	Chevrolet	Enjoy	Silver	CN202205	RC202205	ISN202205
	KA01 1975	F00000004	PP	Toyota	Etios	Silver	CN202206	RC202206	ISN202206
	MH12 5815	F00000001	MI	Nissan	Micra	Silver	CN202202	RC202202	ISN202202
	MH13 1602	F00000001	PS	Maruti	Swift-Dzire	Silver	CN202203	RC202203	ISN202203
	MH13 5005	F00000001	AT	Bajaj	Compact RE	Yellow-Green	CN202201	RC202201	ISN202201
	MH47 8820	F00000003	AT	Mahindra	Alpha	Black-Yellow	CN202204	RC202204	ISN202204
●	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Vehicle

Table Implementation

	OperatorID	PartnerID
▶	F000000001	P000000003
	F000000002	P000000007
	F000000003	P000000008
	F000000004	P000000010
●	HULL	HULL

Fleet Operator

	DriverID	PartnerID	VehicleNo	OperatorID	LicenseNo	Rating	Status
▶	DR00000001	P000000001	MH13 5005	F000000001	LIN34625	4.6	ON
	DR00000002	P000000002	MH12 5815	F000000001	LIN95672	3.7	ON
	DR00000003	P000000004	MH13 1602	HULL	LIN56724	4.8	OFF
	DR00000004	P000000005	MH47 8820	F000000003	LIN59157	4.4	OFF
	DR00000005	P000000006	DL2C 0070	F000000002	LIN56242	4.6	ON
	DR00000006	P000000009	KA01 1975	F000000004	LIN87641	4.8	ON
●	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Driver

	PaymentID	PaymentType
▶	PAY0000001	CH
	PAY0000002	CH
	PAY0000003	UP
	PAY0000004	UP
	PAY0000005	OW
	PAY0000006	OW
	PAY0000008	CD
	PAY0000009	CH
●	HULL	HULL

Payment

Table Implementation

	CRN	BookingID	DriverID	VehideType	PaymentID	FinalCost	Status	TripRating
▶	6789343901	BK00000001	DR00000001	AT	PAY0000001	250	RD	5
	6789343902	BK00000002	DR00000002	MI	PAY0000002	160	RD	5
	6789343903	BK00000003	DR00000004	AT	NULL	0	CC	NULL
	6789343904	BK00000004	DR00000003	PS	PAY0000003	345	RD	3
	6789343905	BK00000005	DR00000005	PV	PAY0000004	410	RD	5
	6789343906	BK00000006	DR00000001	AT	PAY0000005	300	RD	5
	6789343907	BK00000007	DR00000004	AT	PAY0000006	150	RD	4
	6789343909	BK00000009	DR00000006	PP	NULL	190	CD	NULL
	6789343910	BK00000010	DR00000001	AT	PAY0000008	210	RD	NULL
●	6789343911	BK00000011	DR00000002	MI	PAY0000009	120	RD	5
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trip

	BookingID	CustomerID	StartAddress	DestinationAddress	RideDate	RideTime	Preferred_Vehicle_Type	RideType	Approximate_Cost	BookingStatus
▶	BK00000001	C000000001	https://goo.gl/maps/fWt1X3pmCDr9t4Kw8	https://goo.gl/maps/73aMH3Laz1ZXJArV7	2022-05-25	16:00:00	AT	TX	245	CF
	BK00000002	C000000001	https://goo.gl/maps/aH4AbanFgMW214c3A	https://g.page/SymbiosisEngineering?share	2022-06-18	16:45:00	MI	TX	162	CF
	BK00000003	C000000001	https://goo.gl/maps/vueHr8bjkArpezkj7	https://g.page/BrowniePointMahim?share	2022-06-19	16:50:00	AT	TX	189	CF
	BK00000004	C000000002	https://g.page/SymbiosisEngineering?share	https://goo.gl/maps/hVrGC8aSEy4fChdMA	2022-09-02	08:30:00	PS	TX	345	CF
	BK00000005	C000000002	https://g.page/DELHICLUBHOUSE?share	https://goo.gl/maps/WACP5F2C8dGALdA9	2022-10-13	06:00:00	PV	OS	412	CF
	BK00000006	C000000003	https://goo.gl/maps/QZn1GNthyhNcA-#Iq8	https://goo.gl/maps/5XY5N2WryWsU75vW76	2022-10-20	18:40:00	AT	TX	299	CF
	BK00000007	C000000003	https://goo.gl/maps/dhpnyYDYaxp6PSvKA	https://goo.gl/maps/L2WVNNgSCsBkAuZ9	2022-10-22	19:00:00	AT	TX	150	CF
	BK00000008	C000000004	https://goo.gl/maps/rVvtvDVzclJenRHf7	https://goo.gl/maps/SKY5N2WryWsU75vW76	2022-11-02	05:30:00	PS	OS	99	CF
	BK00000009	C000000004	https://goo.gl/maps/RH16BckEpvrhJy8	https://goo.gl/maps/jk54cQvy8kWhnyM8	2022-11-11	05:30:00	PP	OS	190	CF
	BK00000010	C000000005	https://g.page/apricot-service-apartments?share	https://g.page/SymbiosisEngineering?share	2022-10-30	12:35:00	AT	TX	211	CF
	BK00000011	C000000005	https://goo.gl/maps/eUjFm8xELXecjXk8	https://goo.gl/maps/AqPvvNNtzUxcGSva9	2022-11-04	14:00:00	MI	TX	120	CF
●	BK00000012	C000000005	https://goo.gl/maps/hVrGC8aSEy4fChdMA	https://goo.gl/maps/73aMH3Laz1ZXJArV7	2022-11-08	13:30:00	AT	TX	79	CP
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Books

Execution of queries, functions procedures and triggers

Queries

1. How many users have Premium access?

```
select COUNT(CustomerID) from Customer where ProfileType='P';
```

The screenshot shows a query editor with the following details:

- Query number: 68
- Query text: `select COUNT(CustomerID) from Customer where ProfileType='P';`
- Result Grid:

COUNT(CustomerID)
2

2. What model car does Kareen drive?

```
select Company, Model from Vehicle JOIN Driver JOIN Partner JOIN User ON
Driver.VehicleNo= Vehicle.VehicleNo and
Driver.PartnerID = Partner.PartnerID and User.UserID= Partner.UserID and
User.FirstName='Kareen';
```

The screenshot shows a query editor with the following details:

- Query number: 70
- Query text: `select Company, Model from Vehicle JOIN Driver JOIN Partner JOIN User ON
Driver.VehicleNo= Vehicle.VehicleNo and
Driver.PartnerID = Partner.PartnerID and User.UserID= Partner.UserID and
User.FirstName='Kareen';`
- Result Grid:

Company	Model
Chevrolet	Enjoy

Execution of queries, functions procedures and triggers

Queries

3. How many times has Driver Nikita cancelled a trip?

```
select COUNT(CRN) from Trip JOIN Driver JOIN Partner JOIN User  
ON  
    Trip.DriverID=Driver.DriverID and  
    Driver.PartnerID=Partner.PartnerID and  
    Partner.userID=User.UserID  
    and  
    user.FirstName='Nikita' and  
    trip.Status='CD';
```

```
76      select COUNT(CRN) from Trip JOIN Driver JOIN Partner JOIN User ON  
77          Trip.DriverID=Driver.DriverID and  
78              Driver.PartnerID=Partner.PartnerID and Partner.userID=User.UserID  
79          and  
80              user.FirstName='Nikita' and  
81                  trip.Status='CD';  
82
```

Result Grid	
	Filter Rows:
COUNT(CRN)	1

4. How many trips have been cancelled by either a driver or a customer?

```
select COUNT(CRN) from trip where Status like 'C%';
```

```
83  
84      select COUNT(CRN) from trip where Status like 'C%';  
85
```

Result Grid	
	Filter Rows:
COUNT(CRN)	2

Execution of queries, functions procedures and triggers

Queries

5. How many Auto drivers does Ola have?

```
select COUNT(driverID) from Driver JOIN Vehicle ON  
Driver.VehicleNo = Vehicle.VehicleNo and  
Type = 'AT';
```

```
87      select COUNT(driverID) from Driver JOIN Vehicle ON  
88      Driver.VehicleNo = Vehicle.VehicleNo and  
89      Type = 'AT';
```

Result Grid	
	COUNT(driverID)
▶	2

6. How much money have Ola drivers received via cash?

```
select SUM(FinalCost) from Trip JOIN Payment ON  
Trip.PaymentID = Payment.PaymentID and  
PaymentType = 'CH';
```

```
91      select SUM(FinalCost) from Trip JOIN Payment ON  
92      Trip.PaymentID = Payment.PaymentID and  
93      PaymentType = 'CH';
```

Result Grid	
	SUM(FinalCost)
▶	530

Execution of queries, functions procedures and triggers

Queries

7. How many independent drivers does Ola have?

```
select COUNT(DriverID) from Driver where OperatorID IS NULL;
```

The screenshot shows a database query results grid. At the top, there is a SQL command: "95 select COUNT(DriverID) from Driver where OperatorID IS NULL;". Below the command is a horizontal toolbar with various icons: "Result Grid" (selected), "Filter Rows:", "Export:" (with a file icon), and "Wrap Cell Content:" (with a text icon). The main area displays a single row of data in a table format:

COUNT(DriverID)
1

Execution of queries, functions procedures and triggers

Function Procedure

Function 1

```
DELIMITER :  
create function updateOlaWalletAmount(custID varchar(15), val float)  
returns float  
deterministic  
begin  
    declare totalAmount float;  
  
    update Customer set OlaWallet_Amount = OlaWallet_Amount + val where CustomerID = custID;  
  
    select olaWallet_Amount into totalAmount from Customer where CustomerID = custID;  
  
    return totalAmount;  
end :  
delimiter ;  
  
select updateOlaWalletAmount('C0000000005', 100);
```

	updateOlaWalletAmount('C0000000005', 100)
▶	300

	CustomerID	UserID	ProfileType	EmailId	OlaWallet_Amount
▶	C0000000001	1	P	akulz123@gmail.com	0
	C0000000002	2	N	yajwanbabu@gmail.com	822
	C0000000003	3	P	cdentina455@gmail.com	390
	C0000000004	4	N	devi.vishare134@gmail.com	1500
*	C0000000005	5	N	ethanhuntMI@gmail.com	300
	NULL	NULL	NULL	NULL	NULL

Updated Customer Table

Execution of queries, functions procedures and triggers

Function Procedure

Function 2

```
DELIMITER :
create function recalculateRating(cr varchar(12), did varchar(15), rate int)
returns float
deterministic
BEGIN
    declare oldRating float;
    declare newRating float;
    declare totalRatedRides float;

    update Trip set TripRating = rate where CRN = cr;

    select COUNT(CRN) into totalRatedRides FROM Trip where DriverID=did and TripRating IS NOT NULL;

    select Rating into oldRating from Driver where DriverID = did;

    set newRating = (oldRating + rate) / totalRatedRides;

    update Driver set Rating = newRating where DriverID = did;

    return newRating;
end ;
delimiter ;

select recalculateRating('6789343910', 'DR00000001', 2);
```

```
recalculateRating('6789343910',  
'DR00000001', 2)
```

```
recalculateRating('6789343910',  
'DR00000001', 2)
```

2.2

Updated Driver Table

Updated Trip Table

Execution of queries, functions procedures and triggers

Function Procedure

Function 2

```
DELIMITER :  
create function recalculateRating(cr varchar(12), did varchar(15), rate int)  
returns float  
deterministic  
BEGIN  
    declare oldRating float;  
    declare newRating float;  
    declare totalRatedRides float;  
  
    update Trip set TripRating = rate where CRN = cr;  
  
    select COUNT(CRN) into totalRatedRides FROM Trip where DriverID=did and TripRating IS NOT NULL;  
  
    select Rating into oldRating from Driver where DriverID = did;  
  
    set newRating = (oldRating + rate) / totalRatedRides;  
  
    update Driver set Rating = newRating where DriverID = did;  
  
    return newRating;  
end :  
delimiter ;  
  
select recalculateRating('6789343910', 'DR00000001', 2);
```

Output

	recalculateRating('6789343910', 'DR00000001', 2)
▶	2.2

Execution of queries, functions procedures and triggers

Function Procedure

Procedure 1

```
delimiter :  
CREATE PROCEDURE AddFavLocation(  
    IN CID varchar(15),  
    IN FLOCATION varchar(50))  
BEGIN  
    insert into customer_favourite_location values(CID,Flocation) ;  
END :  
delimiter ;  
  
• Call AddFavLocation('C0000000005','https://goo.gl/maps/4JS43ULPbf77Ygtu8');  
• select * from customer_favourite_location
```

Output

	CustomerID	Location
	C0000000001	https://goo.gl/maps/47EgFkTPzEo4ATMT9
	C0000000001	https://goo.gl/maps/XqTjpqeN9NyvoEHfA
	C0000000002	https://goo.gl/maps/2n1d7i6a2Avw5mm9A
	C0000000002	https://goo.gl/maps/3qag5CxKSJTzJ1Q38
	C0000000002	https://goo.gl/maps/sgSfxzLkNavUwpnK6
	C0000000003	https://goo.gl/maps/heyty5qVhuf3tHD56
	C0000000003	https://goo.gl/maps/uuBheMHhWHRFvCmy7
	C0000000004	https://goo.gl/maps/afrWYRMbHzEkMcw38
	C0000000004	https://goo.gl/maps/Uh8UJtXkeR58xyBU6
*	C0000000005	https://goo.gl/maps/4JS43ULPbf77Ygtu8
	NULL	NULL

Updated Customer_Favourite_Location Table

Execution of queries, functions procedures and triggers

Function Procedure

Procedure 2

```
delimiter :
```

```
CREATE PROCEDURE MoneyCollected(
    OUT CASH float,OUT CARD float, OUT wallet float,OUT UPI float)
BEGIN
    SELECT SUM(finalcost) into cash from trip join payment on
        trip.paymentid=payment.paymentid and
        paymenttype='CH';
    SELECT SUM(finalcost) into card from trip join payment on
        trip.paymentid=payment.paymentid and
        paymenttype='CD';
    SELECT SUM(finalcost) into wallet from trip join payment on
        trip.paymentid=payment.paymentid and
        paymenttype='OW';
    SELECT SUM(finalcost) into UPI from trip join payment on
        trip.paymentid=payment.paymentid and
        paymenttype='UP';
END ;
delimiter ;
```



```
call MoneyCollected(@Cash,@Card,@wallet,@UPI);
select @Cash,@Card,@wallet,@UPI;
```

Output

	@Cash	@Card	@wallet	@UPI
▶	530	210	450	755

Execution of queries, functions procedures and triggers

Trigger

```
DELIMITER :  
CREATE TRIGGER AgeCheck  
AFTER INSERT  
ON user FOR EACH ROW  
BEGIN  
    IF YEAR(NEW.DOB) > '2004' THEN  
  
        SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Person must be older than 18.';  
  
    END IF;  
END :  
  
DELIMITER ;  
  
INSERT INTO USER Values (16,'Ganesh','Chand',9999999999, '2005-04-01','*****');
```

Output

```
132 22.25.38 INSERT INTO USER Values (16,'Ganesh','Chand',9999999999, '2005-04-01','*****') Error Code: 1644. Person must be older than 18. 0.000 sec
```

Thank You