Database Management System for Exotic Car Rental Platform

MSIS2603

Guided by:
Professor David Zimbra

Developed by:
Oleksandr Vozniuk

1. Business Overview

1.1 Proposed Organization

Proposed organization is a car rental business that targets specific categories - exotic and premium cars. Unlike a typical rental car company, this organization does not own or lease cars for further rent out to customers, it only provides a platform that connects independent rental companies with people that are interested in renting those cars. The platform provides real-time availability and prices for all available cars based on customer's search parameters, manages bookings, takes care of all payments, insurance, accident reports and customer claims. Additionally, extra services are offered: concierge delivery service and chauffeur service.

1.2 Database Contents

There are four major parties in a database:

Customers - make reservations for automobiles, drive their reserved automobiles, have automobiles delivered by concierges, use chauffeur services. If something goes wrong, customers can send accident reports, claims.

Rental Companies - independent rental companies that are already in the car rental business. They partner with the proposed company and provide information about their cars and prices. Using the proposed platform and the proposed database, their cars are listed on the platform's website and can be booked by customers based on their search parameters. The proposed platform takes care of everything related to reservations, customers, extra services, while rental companies only charge for the provided cars.

Contractor Companies - independent contractor companies that are doing business providing chauffeur and concierge services. The contractor companies have employees that are chauffeurs and concierges. The proposed company contracts with those businesses and they cooperate together, so every time a concierge or a chauffeur is needed for a reservation, a contractor company provides its employee for a reservation and charges for the work. All the

important information about the contractor companies and contractor employees is stored in the proposed database, including information about each delivery and chauffeur trip they make.

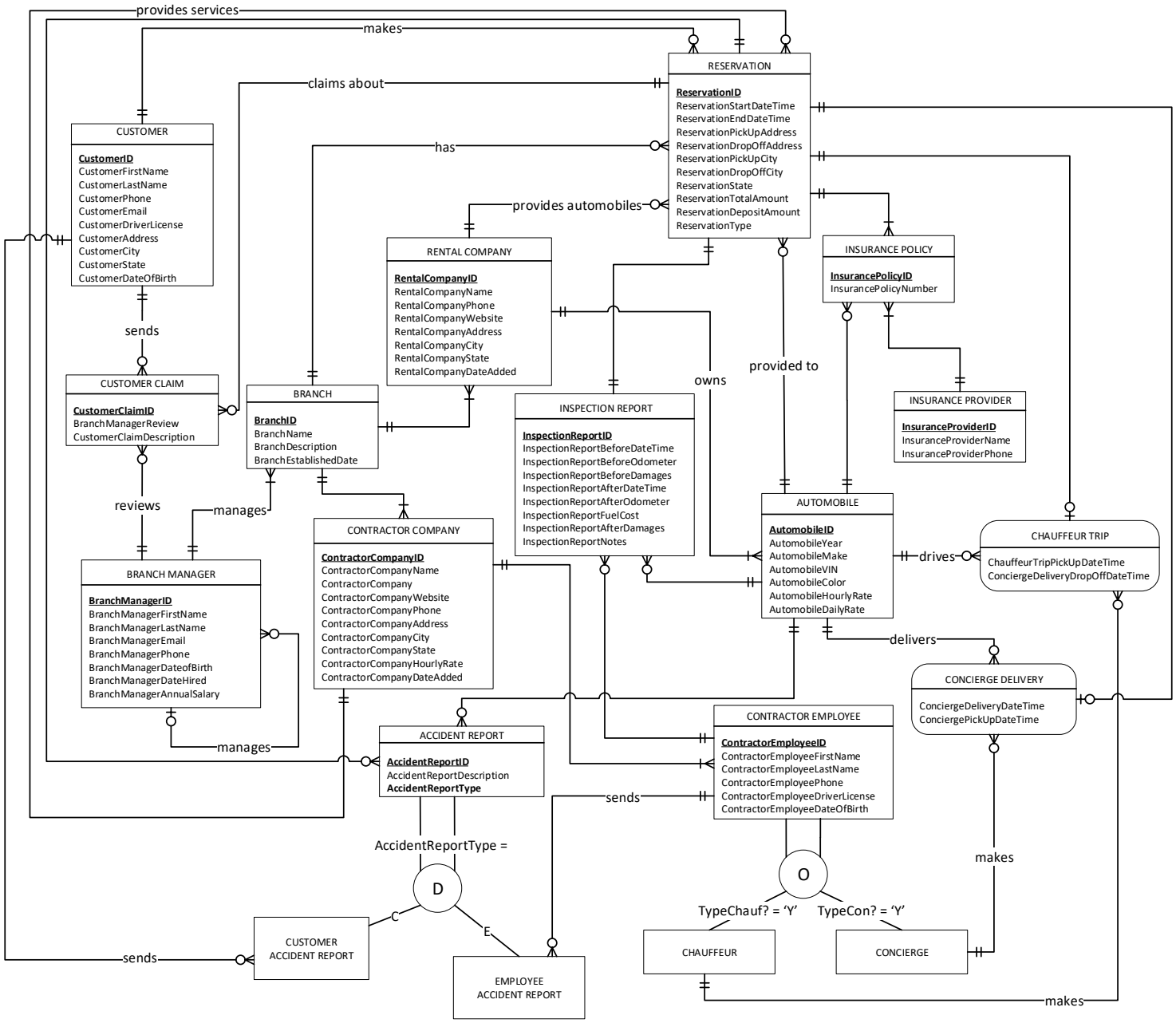
Managers - managers are the core employees in the proposed business. As it is a platform, it should be well managed in order to make all the parties satisfied and let the business grow. The company structure consists of a number of branches, which consist of geographic regions: states, cities. Each branch has a branch manager, who is responsible for his branch and rental companies along with contractor companies that are related to that branch. Also, branch managers take care of customer claims and any accidents that took place during reservations that are under their branches. One branch manager has an additional responsibility to managing his branch by being a general manager and controlling other managers.

1.3 Proposed value

For the proposed business organization the database is needed, as it will be the core part of the platform, containing all the important data about customers, rental companies, bookings, contractor companies, insurance, automobiles and other important parts of the rental business. The platform has value for customers by saving their time and providing fast/reliable service. Saving their time because exotic car rental market is segmented, and there is no just one company renting out cars in the whole United States. Each state and city has many different independent car rental companies. So, people that travel somewhere and do not know about any good company in that city will spend time looking and comparing different ones. Having a platform that people can use in every occasion (and which has all needed extra services) will provide a great value for customers and retain them for further bookings. As for the rental companies, they will get a new source of clients.

2. Database development

2.1 Conceptual schema - ER model



2.2 Logical Schema

CUSTOMER	**CustomerID**	CustomerFirstName	CustomerLastName	CustomerPhone	CustomerEmail	CustomerDriverLicense
	CustomerAddress	CustomerCity	CustomerState	CustomerDateOfBirth		
RESERVATION	**ReservationID**	***CustomerID***	***BranchID***	***RentalCompanyID***	***AutomobileID***	***ContractorCompanyID***
	ReservationStartDateTime	ReservationEndDateTime	ReservationPickUpAddress	ReservationDropOffAddress	ReservationPickUpCity	ReservationDropOffCity
	ReservationState	ReservationTotalAmount	ReservationDepositAmount	ReservationType		
RENTAL_COMPANY	**RentalCompanyID**	***BranchID***	RentalCompanyName	RentalCompanyPhone	RentalCompanyWebsite	RentalCompanyAddress
	RentalCompanyCity	RentalCompanyState	RentalCompanyDateAdded			
AUTOMOBILE	**AutomobileID**	***RentalCompanyID***	AutomobileYear	AutomobileMake	AutomobileModel	AutomobileVIN
	AutomobileColor	AutomobileHourlyRate	AutomobileDailyRate			
CONTRACTOR_COMPANY	**ContractorCompanyID**	***BranchID***	ContractorCompanyName	ContractorCompanyWebsite	ContractorCompanyPhone	ContractorCompanyAddress
	ContractorCompanyCity	ContractorCompanyState	ContractorCompanyHourlyRate	ContractorCompanyDateAdded		
CONTRACTOR_EMPLOYEE	**ContractorEmployeeID**	***ContractorCompanyID***	**ContractorEmployeeTypeChauf**	**ContractorEmployeeTypeCon**	ContractorEmployeeFirstName	ContractorEmployeeLastName
	ContractorEmployeePhone	ContractorEmployeeDriverLicense	ContractorEmployeeDateOfBirth			
CONTRACTOR_EMPLOYEE_ CHAUFFEUR	***CHContractorEmployeeID***					
CONTRACTOR_EMPLOYEE_ CONCIERGE	***COContractorEmployeeID***					
CONCIERGE_DELIVERY	***COContractorEmployeeID***	***AutomobileID***	***ReservationID***	ConciergeDeliveryDateTime	ConciergeDeliveryPickUpDateTime	
CHAUFFEUR_TRIP	***CHContractorEmployeeID***	***AutomobileID***	***ReservationID***	ChauffeurTripPickUpDateTime	ChauffeurTripDropOffDateTime	
BRANCH_MANAGER	**BranchManagerID**	*BMGeneralManagerID*	BranchManagerFirstName	BranchManagerLastName	BranchManagerEmail	BranchManagerPhone
	BranchManagerDateofBirth	BranchManagerDateHired	BranchManagerAnnualSalary			
BRANCH	**BranchID**	BranchName	BranchDescription	BranchEstablishedDate	***BranchManagerID***	
INSPECTION_REPORT	**InspectionReportID**	***AutomobileID***	***ReservationID***	***ContractorEmployeeID***	InspectionReportBeforeDateTime	InspectionReportBeforeOdometer
	InspectionReportBeforeDamages	InspectionReportAfterDateTime	InspectionReportAfterOdometer	InspectionReportFuelCost	InspectionReportAfterDamages	InspectionReportNotes
ACCIDENT_REPORT	**AccidentReportID**	***AutomobileID***	***ReservationID***	**AccidentReportType**	AccidentReportDescription	
ACCIDENT_REPORT_ CUSTOMER	***CAccidentReport***	***CustomerID***				
ACCIDENT_REPORT_ EMPLOYEE	***EAccidentReport***	***ContractorEmployeeID***				
CUSTOMER_CLAIM	**CustomerClaimID**	***CustomerID***	***ReservationID***	*BranchManagerReview*	CustomerClaimDescription	
INSURANCE_POLICY	**InsurancePolicyID**	***ReservationID***	***AutomobileID***	***InsuranceProviderID***	InsurancePolicyNumber	
INSURANCE_PROVIDER	**InsuranceProviderID**	InsuranceProviderName	InsuranceProviderPhone			

2.3 Data dictionary

CUSTOMER_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
CustomerID	INT	Not Null	PK	Unique identifier for a customer	12
CustomerFirstName	NVARCHAR(35)	Not Null		First name of a customer	John
CustomerLastName	NVARCHAR(35)	Not Null		Last name of a customer	Brown
CustomerPhone	CHAR(10)	Not Null		Phone number of a customer	7027775678
CustomerEmail	NVARCHAR(35)	Not Null		Email address of a customer	johnb89@gmail.com
CustomerDriverLicense	NVARCHAR(20)	Not Null		Customer's driver license number	A1234567
CustomerAddress	NVARCHAR(35)	Not Null		Address where a customer resides	40 Carson Dr.
CustomerCity	NVARCHAR(35)	Not Null		City where a customer resides	Cupertino
CustomerState	CHAR(2)	('AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC', 'FL', 'GA', 'HI', 'ID', 'IL', IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'), Not Null		State where a customer resides	CA
CustomerDateOfBirth	DATE	Not Null, Age>=18		Customer Date of Birth	09/12/1989

RENTALCOMPANY_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
RentalCompanyID	INT	Not Null	PK	Unique identifier for a rental company	16
BranchID	INT	Not Null	FK	Unique identifier for a branch	11
RentalCompanyName	NVARCHAR(35)	Not Null		Name of a rental company	Royal Exotic Car Rental

RENTALCOMPANY_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
RentalCompanyPhone	CHAR(10)	Not Null		Phone number of a rental company	4246539762
RentalCompanyWebsite	NVARCHAR(35)	Not Null		Website address of a rental company	https://royalexoticcarrental.com/
RentalCompanyAddress	NVARCHAR(35)	Not Null		Physical location street address of a rental company	1525 La Cienega Boulevard
RentalCompanyCity	NVARCHAR(35)	Not Null		City where a rental company is located	Los Angeles
RentalCompanyState	CHAR(2)	(‘AL’, ‘AK’, ‘AZ’, ‘AR’, ‘CA’, ‘CO’, ‘CT’, ‘DE’, ‘DC’, ‘FL’, ‘GA’, ‘HI’, ‘ID’, ‘IL’, IN’, ‘IA’, ‘KS’, ‘KY’, ‘LA’, ‘ME’, ‘MD’, ‘MA’, ‘MI’, ‘MN’, ‘MS’, ‘MO’, ‘MT’, ‘NE’, ‘NV’, ‘NH’, ‘NJ’, ‘NM’, ‘NY’, ‘NC’, ‘ND’, ‘OH', ‘OK’, ‘OR’, ‘PA’, ‘RI’, ‘SC’, ‘SD’, ‘TN’, ‘TX’, ‘UT’, ‘VT’, ‘VA’, ‘WA’, ‘WV’, ‘WI’, ‘WY’), Not Null		State where a rental company is located	CA
RentalCompanyDateAdded	DATE	Not Null		Date when a rental company was added to the database	02/01/2019

AUTOMOBILE_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
AutomobileID	INT	Not Null	PK	Unique identifier for an automobile	35
RentalCompanyID	INT	Not Null	FK	Unique identifier for a rental company	24
AutomobileYear	SMALLINT	<= Current Year, Not Null		Production year of an automobile	2018
AutomobileMake	NVARCHAR(20)	Not Null		Make of an automobile	Ferrari
AutomobileModel	NVARCHAR(20)	Not Null		Model of an automobile	Portofino
AutomobileVIN	CHAR(17)	Not Null		Automobile VIN number	1672NG678NG156289
AutomobileColor	NVARCHAR(20)	Not Null		Automobile exterior color	Red
AutomobileHourlyRate	DECIMAL(6,2)	Not Null, > 0		Automobile hourly rate	200.00

AUTOMOBILE_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
AutomobileDailyRate	DECIMAL(6,2)	Not Null, > 0		Automobile daily rate	1200.00

RESERVATION_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
ReservationID	INT	Not Null	PK	Unique identifier for a reservation	345
CustomerID	INT	Not Null	FK	Unique identifier for a customer	56
BranchID	INT	Not Null	FK	Unique identifier for a branch	2
RentalCompanyID	INT	Not Null	FK	Unique identifier for a rental company	12
AutomobileID	INT	Not Null	FK	Unique identifier for an automobile	57
ContractorCompanyID	INT	Not Null	FK	Unique identifier for a contractor company	25
ReservationStartDateTime	DATETIME	Not Null		Date and time when a reservation starts	01/10/2019 11:00:00
ReservationEndDateTime	DATETIME	Not Null		Date and time when a reservation ends	01/12/2019 11:00:00
ReservationPickUpAddress	NVARCHAR(35)	Not Null		Address where a customer will pick up an automobile or will be picked up by a chauffeur	1525 La Cienega Boulevard
ReservationDropOffAddress	NVARCHAR(35)			Address where a customer will drop off an automobile	1525 La Cienega Boulevard
ReservationPickUpCity	NVARCHAR(35)	Not Null		City where a customer will pick up an automobile or will be picked up by a chauffeur	Los Angeles
ReservationDropOffCity	NVARCHAR(35)			City where a customer will drop off an automobile(null for a chauffeur trip)	Los Angeles
ReservationState	CHAR(2)	(‘AL’, ‘AK’, ‘AZ’, ‘AR’, ‘CA’, ‘CO’, ‘CT’, ‘DE’, ‘DC’, ‘FL’, ‘GA’, ‘HI’, ‘ID’, ‘IL’, IN’, ‘IA’, ‘KS’, ‘KY’, ‘LA’, ‘ME’, ‘MD’, ‘MA’, ‘MI’, ‘MN’, ‘MS’, ‘MO’, ‘MT’, ‘NE’, ‘NV’, ‘NH’, ‘NJ’, ‘NM’, ‘NY’, ‘NC’, ‘ND’, ‘OH’, ‘OK’, ‘OR’, ‘PA’, ‘RI’, ‘SC’, ‘SD’, ‘TN’, ‘TX’, ‘UT’, ‘VT’, ‘VA’, ‘WA’, ‘WV’, ‘WI’, ‘WY’), Not Null		State where a customer reserves a car	CA
ReservationTotalAmount	DECIMAL(8,2)	>0, Not Null		Total amount of a reservation	2550.89
ReservationDepositAmount	DECIMAL(8,2)	>0, Not Null		Deposit amount of a reservation	3000.00

RESERVATION_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
ReservationType	CHAR(9)	(‘CHAUFFEUR’, ’CONCIERGE’), Not Null		Type of a reservation, with a chauffeur or with concierge delivery	CONCIERGE

CONTRACTORCOMPANY_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
ContractorCompanyID	INT	Not Null	PK	Unique identifier for a contractor company	28
BranchID	INT	Not Null	FK	Unique identifier for a branch	3
ContractorCompanyName	NVARCHAR(35)	Not Null		Name of a contractor company	Dryver
ContractorCompanyWebsite	NVARCHAR(35)			Website of a contractor company	dryver.com
ContractorCompanyPhone	CHAR(10)	Not Null		Phone number of a contractor company	8778236933
ContractorCompanyAddress	NVARCHAR(35)	Not Null		Physical address of a contractor company	4209 Vernon Street
ContractorCompanyCity	NVARCHAR(35)	Not Null		City where a contractor company is located	Los Angeles
ContractorCompanyState	CHAR(10)	(‘AL’, ‘AK’, ‘AZ’, ‘AR’, ‘CA’, ‘CO’, ‘CT’, ‘DE’, ‘DC’, ‘FL’, ‘GA’, ‘HI’, ‘ID’, ‘IL’, IN’, ‘IA’, ‘KS’, ‘KY’, ‘LA’, ‘ME’, ‘MD’, ‘MA’, ‘MI’, ‘MN’, ‘MS’, ‘MO’, ‘MT’, ‘NE’, ‘NV’, ‘NH’, ‘NJ’, ‘NM’, ‘NY’, ‘NC’, ‘ND’, ‘OH', ‘OK’, ‘OR’, ‘PA’, ‘RI’, ‘SC’, ‘SD’, ‘TN’, ‘TX’, ‘UT’, ‘VT’, ‘VA’, ‘WA’, ‘WV’, ‘WI’, ‘WY’), Not Null		State where a contractor company is located	CA
ContractorCompanyHourlyRate	DECIMAL(5,2)	Not Null		Hourly rate for an employee that a contractor company provides	45.00
ContractorCompanyDateAdded	DATE	Not Null		Date where a contractor company was added to the database	12/16/2018

CONTRACTOREMPLOYEE_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
ContractorEmployeeID	INT	Not Null	PK	Unique identifier for a contractor employee	15
ContractorCompanyID	INT	Not Null	FK	Unique identifier for a contractor company that provides an employee	46
ContractorEmployeeTypeChauf	BIT	Not Null		Boolean value, 1 if a contractor employee can be a chauffeur, 0 if not	1
ContractorEmployeeTypeCon	BIT	Not Null		Boolean value, 1 if a contractor employee can be a concierge, 0 if not	1
ContractorEmployeePhone	CHAR(10)	Not Null		Phone number of a contractor employee	7026896754
ContractorEmployeeFirstName	NVARCHAR(35)	Not Null		First name of a contractor employee	James
ContractorEmployeeLastName	NVARCHAR(35)	Not Null		Last name of a contractor employee	Wright
ContractorEmployeeDriverLicense	NVARCHAR(20)	Not Null		Driver License Number of a contractor employee	A1234567
ContractorEmployeeDateOfBirth	DATE	Not Null		Date of birth of a contractor employee	04/25/1985

CONTRACTOREMPLOYEECONCIERGE_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
COContractorEmployeeID	INT	Not Null	PK, FK	Unique identifier for a contractor concierge employee	56

CONTRACTOREMPLOYEECHAUFFEUR_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
CHContractorEmployeeID	INT	Not Null	PK,FK	Unique identifier for a contractor chauffeur employee	35

CONCIERGEDELIVERY_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
COContractorEmployeeID	INT	Not Null	PK,FK	Unique identifier for a contractor concierge employee	46
AutomobileID	INT	Not Null	PK,FK	Unique identifier for an automobile	36
ReservationID	INT	Not Null	PK,FK	Unique identifier for a reservation	456
ConciergeDeliveryDateTime	DATETIME	Not Null		Time and date when a concierge delivers an automobile to a customer	01/10/2019 11:00:00
ConciergeDeliveryPickUpDateTime	DATETIME			Time and date when a concierge picks up an automobile from a customer	01/12/2019 11:00:00

CHAUFFEURTRIP_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
CHContractorEmployeeID	INT	Not Null	PK,FK	Unique identifier for a contractor chauffeur employee	25
AutomobileID	INT	Not Null	PK,FK	Unique identifier for an automobile	79
ReservationID	INT	Not Null	PK,FK	Unique identifier for a reservation	659
ChauffeurTripPickUpDateTime	DATETIME	Not Null		Time and date when a chauffeur picks up a customer	01/10/2019 11:05:56
ChauffeurTripDropOffDateTime	DATETIME			Time and date when a chauffeur drops off a customer	01/10/2019 18:59:30

BRANCHMANAGER_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
BranchManagerID	INT	Not Null	PK	Unique identifier for a branch manager	1
BMGeneralManagerID	INT		FK	Unique identifier for a branch manager who is a general manager for other branch managers	NULL
BranchManagerFirstName	NVARCHAR(35)	Not Null		First name of a branch manager	Kyle
BranchManagerLastName	NVARCHAR(35)	Not Null		Last name of a branch manager	Anderson
BranchManagerEmail	NVARCHAR(35)	Not Null		Email address of a branch manager	kyle@gmail.com
BranchManagerPhone	CHAR(10)	Not Null		Phone number of a branch manager	7029997733
BranchManagerDateofBirth	DATE	Not Null		Date of birth of a branch manager	06/10/1983
BranchManagerDateHired	DATE	Not Null		Date when a branch manager was hired	04/12/2018
BranchManagerAnnualSalary	DECIMAL(9,2)	Not Null		Annual Salary of a branch manager	180000.00

BRANCH_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
BranchID	INT	Not Null	PK	Unique identifier for a branch	3
BranchManagerID	INT	Not Null	FK	Unique identifier for a branch manager	2
BranchName	NVARCHAR(20)	Not Null		Branch name	West
BranchDescription	NVARCHAR(100)			Description of a branch	West branch consists of California and Nevada states.
BranchEstablishedDate	DATE	Not Null		Date when a branch was established	12/01/2018

INSPECTIONREPORT_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
InspectionReportID	INT	Not Null	PK	Unique identifier for an inspection report	780
AutomobileID	INT	Not Null	FK	Unique identifier for an automobile	26
ReservationID	INT	Not Null	FK	Unique identifier for a reservation	678
ContractorEmployeeID	INT	Not Null	FK	Unique identifier for a contractor employee	35
InspectionReportBeforeDateTime	DATETIME	Not Null		Date and time when an automobile was inspected before a reservation	01/10/2019 11:05:56
InspectionReportBeforeOdometer	DECIMAL(6,0)	Not Null		Odometer reading before a customer drives a car or before a chauffeur picks up a customer	2567
InspectionReportBeforeDamages	NVARCHAR(300)			Damages that an automobile has before a reservation starts	NULL
InspectionReportAfterDateTime	DATETIME			Date and time of the inspection after a reservation	01/12/2019 21:45:56
InspectionReportAfterOdometer	DECIMAL(6,0)			Odometer reading after a customer returns a car or after a chauffeur drops off a customer	2856
InspectionReportFuelCost	DECIMAL(6,2)			Cost of refueling an automobile after a customer dropped off a car or during/after trip with a chauffeur, NULL only if a customer returned an automobile with a full tank	60.00
InspectionReportAfterDamages	NVARCHAR(300)			Damages that an employee found after a customer finished a reservation	Scratch on the front bumper, left side
InspectionReportNotes	NVARCHAR(300)			Additional notes that an employee can add	Car was returned very dirty

ACCIDENTREPORT_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
AccidentReportID	INT	Not Null	PK	Unique identifier for an accident report	20
AutomobileID	INT	Not Null	FK	Unique identifier for an automobile	67
ReservationID	INT	Not Null	FK	Unique identifier for a reservation	450
AccidentReportType	CHAR(1)	Not Null, ('E','C')		Discriminator, E - if AccidentReportEmployee, C - if AccidentReportCustomer	E
AccidentReport Description	NVARCHAR (500)	Not Null		Description of an accident	Another car scratched rear bumper on a parking

ACCIDENTREPORTEMPLOYEE_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
EAccidentReport	INT	Not Null	PK, FK	Unique identifier for an accident report made by an employee	25
ContractorEmployeeID	INT	Not Null	FK	Unique identifier for a contractor employee	56

ACCIDENTREPORTCUSTOMER_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
CAccidentReport	INT	Not Null	PK, FK	Unique identifier for an accident report made by a customer	68
CustomerID	INT	Not Null	FK	Unique identifier for a customer	156

CUSTOMERCLAIM_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
CustomerClaimID	INT	Not Null	PK	Unique identifier for a customer claim	23
CustomerID	INT	Not Null	FK	Unique identifier for a customer	178
ReservationID	INT	Not Null	FK	Unique identifier for a reservation	786
BranchManagerReview	INT		FK	Shows if a claim was reviewed by a manager, NULL if not reviewed, BranchManageID - when reviewed	4
CustomerClaimDescription	NVARCHAR(500)	Not Null		Claim that a customer writes	This is a claim!

INSURANCEPOLICY_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
InsurancePolicyID	INT	Not Null	PK	Unique identifier for an insurance policy	90
ReservationID	INT	Not Null	FK	Unique identifier for a reservation	650
AutomobileID	INT	Not Null	FK	Unique identifier for an automobile	45
InsuranceProviderID	INT	Not Null	FK	Unique identifier for an insurance provider	StateFarm
InsurancePolicyNumber	NVARCHAR(20)	Not Null		Insurance Policy Number	A123456789

INSURANCEPROVIDER_T					
Name	**Data Type**	**Constraints**	**Key**	**Description**	**Example Value**
InsuranceProviderID	INT	Not Null	PK	Unique identifier for an insurance provider	5
InsuranceProviderName	NVARCHAR(35)	Not Null		Name of an insurance provider	StateFarm
InsuranceProviderPhone	CHAR(10)	Not Null		Phone number of an insurance provider	8007828332

2.4 Creating Tables

Customer_T

```
CREATE TABLE Customer_T
(CustomerID int not null,
CustomerFirstName nvarchar(35) not null,
CustomerLastName nvarchar(35) not null,
CustomerPhone char(10) not null,
CustomerEmail nvarchar(35) not null,
CustomerDriverLicense nvarchar(20) not null,
CustomerAddress nvarchar(35) not null,
CustomerCity nvarchar(35) not null,
CustomerState char(2) not null CHECK(CustomerState IN('AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC',
'FL', 'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH',
'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI',
'WY')),
CustomerDateOfBirth DATE not null CHECK(DATEDIFF(YEAR, CustomerDateOfBirth,
GETDATE())>=18),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID))
```

Branch_T

```
CREATE TABLE Branch_T
(BranchID int not null,
BranchManagerID int not null
BranchName nvarchar(20) not null,
BranchDescription nvarchar(100),
BranchEstablishedDate DATE not null,
CONSTRAINT Branch_PK PRIMARY KEY (BranchID),
CONSTRAINT Branch_FK FOREIGN KEY (BranchManagerID) REFERENCES
BranchManager_T(BranchManagerID))
```

RentalCompany_T

```
CREATE TABLE RentalCompany_T
(RentalCompanyID int not null,
BranchID int not null,
RentalCompanyName nvarchar(35) not null,
RentalCompanyPhone char(10) not null,
RentalCompanyWebsite nvarchar(35) not null,
RentalCompanyAddress nvarchar(35) not null,
RentalCompanyCity nvarchar(35) not null,
RentalCompanyState char(2) not null CHECK(RentalCompanyState IN('AL', 'AK', 'AZ', 'AR', 'CA', 'CO',
'CT', 'DE', 'DC', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT',
'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA',
'WA', 'WV', 'WI', 'WY')),
RentalCompanyDateAdded DATE not null,
CONSTRAINT RentalCompany_PK PRIMARY KEY (RentalCompanyID),
CONSTRAINT RentalCompany_FK FOREIGN KEY (BranchID) REFERENCES Branch_T(BranchID))
```

Automobile_T

```
CREATE TABLE Automobile_T
(AutomobileID int not null,
RentalCompanyID int not null,
AutomobileYear smallint not null,
AutomobileMake nvarchar(20) not null,
AutomobileModel nvarchar(20) not null,
AutomobileVIN char(17) not null,
AutomobileColor nvarchar(20) not null,
AutomobileHourlyRate decimal(6,2) not null CHECK(AutomobileHourlyRate > 0),
AutomobileDailyRate decimal(6,2) not null CHECK(AutomobileDailyRate > 0),
CONSTRAINT Automobile_PK PRIMARY KEY (AutomobileID),
CONSTRAINT Automobile_FK FOREIGN KEY (RentalCompanyID) REFERENCES
RentalCompany_T(RentalCompanyID))
```

ContractorCompany_T

```
CREATE TABLE ContractorCompany_T
(ContractorCompanyID int not null,
BranchID int not null,
ContractorCompanyName nvarchar(35) not null,
ContractorCompanyWebsite nvarchar(35),
ContractorCompanyPhone char(10) not null,
ContractorCompanyAddress nvarchar(35) not null,
ContractorCompanyCity nvarchar(35) not null,
ContractorCompanyState char(2) not null CHECK(ContractorCompanyState IN('AL', 'AK', 'AZ', 'AR', 'CA',
'CO', 'CT', 'DE', 'DC', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO',
'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT',
'VA', 'WA', 'WV', 'WI', 'WY')),
ContractorCompanyHourlyRate decimal(5,2) not null,
ContractorCompanyDateAdded DATE not null,
CONSTRAINT ContractorCompany_PK PRIMARY KEY (ContractorCompanyID),
CONSTRAINT ContractorCompany_FK FOREIGN KEY (BranchID) REFERENCES Branch_T(BranchID))
```

Reservation_T

```
CREATE TABLE Reservation_T
(ReservationID int not null,
CustomerID int not null,
BranchID int not null,
RentalCompanyID int not null,
AutomobileID int not null,
ContractorCompanyID int not null,
ReservationStartDateTime datetime not null,
ReservationEndDateTime datetime not null,
ReservationPickUpAddress nvarchar(35) not null,
ReservationDropOffAddress nvarchar(35),
ReservationPickUpCity nvarchar(35) not null,
ReservationDropOffCity nvarchar(35),
ReservationState char(2) not null CHECK(ReservationState IN('AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE',
'DC', 'FL', 'GA', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV',
'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'WA', 'WV',
'WI', 'WY')),
```

```
ReservationTotalAmount decimal(8,2) not null CHECK(ReservationTotalAmount > 0),
ReservationDepositAmount decimal(8,2) not null CHECK(ReservationDepositAmount > 0),
ReservationType char(9) not null CHECK(ReservationType in ('Concierge', 'Chauffeur')),
CONSTRAINT Reservation_PK PRIMARY KEY (ReservationID),
CONSTRAINT Reservation_FK1 FOREIGN KEY (CustomerID) REFERENCES
Customer_T(CustomerID),
CONSTRAINT Reservation_FK2 FOREIGN KEY (BranchID) REFERENCES Branch_T(BranchID),
CONSTRAINT Reservation_FK3 FOREIGN KEY (RentalCompanyID) REFERENCES
RentalCompany_T(RentalCompanyID),
CONSTRAINT Reservation_FK4 FOREIGN KEY (AutomobileID) REFERENCES
Automobile_T(AutomobileID),
CONSTRAINT Reservation_FK5 FOREIGN KEY (ContractorCompanyID) REFERENCES
ContractorCompany_T(ContractorCompanyID))
```

BranchManager_T

```
CREATE TABLE BranchManager_T
(BranchManagerID int not null,
BMGeneralManagerID int,
BranchManagerFirstName nvarchar(35) not null,
BranchManagerLastName nvarchar(35) not null,
BranchManagerEmail nvarchar(35) not null,
BranchManagerPhone char(10) not null,
BranchManagerDateOfBirth DATE not null,
BranchManagerDateHired DATE not null,
BranchManagerAnnualSalary decimal(9,2) not null,
CONSTRAINT BranchManager_PK PRIMARY KEY (BranchManagerID),
CONSTRAINT BranchManager_FK FOREIGN KEY (BMGeneralManagerID) REFERENCES
BranchManager_T(BranchManagerID))
```

ContractorEmployee_T

```
CREATE TABLE ContractorEmployee_T
(ContractorEmployeeID int not null,
ContractorCompanyID int not null,
ContractorEmployeeTypeChauf BIT not null,
ContractorEmployeeTypeCon BIT not null,
ContractorEmployeeFirstName nvarchar(35) not null,
ContractorEmployeeLastName nvarchar(35) not null,
ContractorEmployeeDriverLicense nvarchar(20) not null,
ContractorEmployeeDateOfBirth DATE not null,
CONSTRAINT ContractorEmployee_PK PRIMARY KEY (ContractorEmployeeID),
CONSTRAINT ContractorEmployee_FK FOREIGN KEY (ContractorCompanyID) REFERENCES
ContractorCompany_T(ContractorCompanyID))
```

ContractorEmployeeChauffeur_T

```
CREATE TABLE ContractorEmployeeChauffeur_T
(ContractorEmployeeChauffeurID int not null,
CONSTRAINT ContractorEmployeeChauffeur_PK PRIMARY KEY (ContractorEmployeeChauffeurID),
CONSTRAINT ContractorEmployeeChauffeur_FK FOREIGN KEY (ContractorEmployeeChauffeurID)
REFERENCES ContractorEmployee_T(ContractorEmployeeID))
```

ContractorEmployeeConcierge_T

```
CREATE TABLE ContractorEmployeeConcierge_T
(ContractorEmployeeConciergeID int not null,
CONSTRAINT ContractorEmployeeConcierge_PK PRIMARY KEY (ContractorEmployeeConciergeID),
CONSTRAINT ContractorEmployeeConcierge_FK FOREIGN KEY (ContractorEmployeeConciergeID)
REFERENCES ContractorEmployee_T(ContractorEmployeeID))
```

ConciergeDelivery_T

```
CREATE TABLE ConciergeDelivery_T
(COContractorEmployeeID int not null,
AutomobileID int not null,
ReservationID int not null,
ConciergeDeliveryDateTime DATETIME not null,
ConciergePickUpDateTime DATETIME,
CONSTRAINT ConciergeDelivery_FK1 FOREIGN KEY (COContractorEmployeeID) REFERENCES
ContractorEmployeeConcierge_T(ContractorEmployeeConciergeID),
CONSTRAINT ConciergeDelivery_FK2 FOREIGN KEY (AutomobileID)
REFERENCES Automobile_T(AutomobileID),
CONSTRAINT ConciergeDelivery_FK3 FOREIGN KEY (ReservationID)
REFERENCES Reservation_T(ReservationID))
```

ChauffeurTrip_T

```
CREATE TABLE ChauffeurTrip_T
(CHContractorEmployeeID int not null,
AutomobileID int not null,
ReservationID int not null,
ChauffeurPickUpDateTime DATETIME not null,
ChauffeurDropOffDateTime DATETIME,
CONSTRAINT ChauffeurDelivery_FK1 FOREIGN KEY (CHContractorEmployeeID) REFERENCES
ContractorEmployeeChauffeur_T(ContractorEmployeeChauffeurID),
CONSTRAINT ChauffeurDelivery_FK2 FOREIGN KEY (AutomobileID)
REFERENCES Automobile_T(AutomobileID),
CONSTRAINT ChauffeurDelivery_FK3 FOREIGN KEY (ReservationID)
REFERENCES Reservation_T(ReservationID))
```

InspectionReport_T

```
CREATE TABLE InspectionReport_T
(InspectionReportID int not null,
AutomobileID int not null,
ReservationID int not null,
ContractorEmployeeID int not null,
InspectionReportBeforeDateTime DATETIME not null,
InspectionReportBeforeOdometer decimal(6,0) not null,
InspectionReportBeforeDamages nvarchar(300),
InspectionReportAfterDateTime DATETIME,
InspectionReportAfterOdometer decimal(6,0),
InspectionReportAfterDamages nvarchar(300),
InspectionReportFuelCost decimal(6,2),
InspectionReportNotes nvarchar(300),
CONSTRAINT InspectionReport_PK PRIMARY KEY (InspectionReportID),
```

```
CONSTRAINT InspectionReport_FK1 FOREIGN KEY (AutomobileID)
REFERENCES Automobile_T(AutomobileID),
CONSTRAINT InspectionReport_FK2 FOREIGN KEY (ReservationID)
REFERENCES Reservation_T(ReservationID),
CONSTRAINT InspectionReport_FK3 FOREIGN KEY (ContractorEmployeeID)
REFERENCES ContractorEmployee_T(ContractorEmployeeID))
```

AccidentReport_T

```
CREATE TABLE AccidentReport_T
(AccidentReportID int not null,
AutomobileID int not null,
ReservationID int not null,
AccidentReportType char(1) not null CHECK(AccidentReportType in('E','C')),
AccidentReportDescription nvarchar not null,
CONSTRAINT AccidentReport_PK PRIMARY KEY (AccidentReportID),
CONSTRAINT AccidentReport_FK1 FOREIGN KEY (AutomobileID)
REFERENCES Automobile_T(AutomobileID),
CONSTRAINT AccidentReport_FK2 FOREIGN KEY (ReservationID)
REFERENCES Reservation_T(ReservationID))
```

AccidentReportEmployee_T

```
CREATE TABLE AccidentReportEmployee_T
(EAccidentReportID int not null,
ContractorEmployeeID int not null,
CONSTRAINT AccidentReportEmployee_PK PRIMARY KEY (EAccidentReportID),
CONSTRAINT AccidentReportEmployee_FK1 FOREIGN KEY (EAccidentReportID)
REFERENCES AccidentReport_T(AccidentReportID),
CONSTRAINT AccidentReportEmployee_FK2 FOREIGN KEY (ContractorEmployeeID)
REFERENCES ContractorEmployee_T(ContractorEmployeeID))
```

AccidentReportCustomer_T

```
CREATE TABLE AccidentReportCustomer_T
(CAccidentReportID int not null,
CustomerID int not null,
CONSTRAINT AccidentReportCustomer_PK PRIMARY KEY (CAccidentReportID),
CONSTRAINT AccidentReportCustomer_FK1 FOREIGN KEY (CAccidentReportID)
REFERENCES AccidentReport_T(AccidentReportID),
CONSTRAINT AccidentReportCustomer_FK2 FOREIGN KEY (CustomerID)
REFERENCES Customer_T(CustomerID))
```

CustomerClaim_T

```
CREATE TABLE CustomerClaim_T
(CustomerClaimID int not null,
CustomerID int not null,
ReservationID int not null,
BranchManagerReview int,
CustomerClaimDescription nvarchar,
CONSTRAINT CustomerClaim_PK PRIMARY KEY (CustomerClaimID),
CONSTRAINT CustomerClaim_FK1 FOREIGN KEY (CustomerID)
REFERENCES Customer_T(CustomerID),
```

```
CONSTRAINT CustomerClaim_FK2 FOREIGN KEY (ReservationID)
REFERENCES Reservation_T(ReservationID),
CONSTRAINT CustomerClaim_FK3 FOREIGN KEY (BranchManagerReview)
REFERENCES BranchManager_T(BranchManagerID))
```

InsuranceProvider_T

```
CREATE TABLE InsuranceProvider_T
(InsuranceProviderID int not null,
InsuranceProviderName nvarchar(35) not null,
InsuranceProviderPhone char(10) not null,
CONSTRAINT InsuranceProvider_PK PRIMARY KEY (InsuranceProviderID))
```

InsurancePolicy_T

```
CREATE TABLE InsurancePolicy_T
(InsurancePolicyID int not null,
ReservationID int not null,
AutomobileID int not null,
InsuranceProviderID int not null,
InsurancePolicyNumber nvarchar(20),
CONSTRAINT InsurancePolicy_PK PRIMARY KEY (InsurancePolicyID),
CONSTRAINT InsurancePolicy_FK1 FOREIGN KEY (ReservationID)
REFERENCES Reservation_T(ReservationID),
CONSTRAINT InsurancePolicy_FK2 FOREIGN KEY (AutomobileID)
REFERENCES Automobile_T(AutomobileID),
CONSTRAINT InsurancePolicy_FK3 FOREIGN KEY (InsuranceProviderID)
REFERENCES InsuranceProvider_T(InsuranceProviderID))
```

3. Views and Procedures

3.1 General Manager Views

3.1.1 View 1. Overall Branch Statistics

The view is created for the general manager who needs to see the overall statistics for all branches. For each branch number of reservations is calculated, as well as total rent earnings and the earned profit, which is 15% from all the reservations earnings. Manager names are also shown to demonstrate who is doing the best. This view is very important to analyze the company's current state of financials.

Creating table:

```
CREATE TABLE BranchStatistics_View
(BranchID int not null,
BranchName nvarchar(max) not null,
BranchDescription nvarchar(max) not null,
NumberOfReservations int not null,
TotalRent decimal(9,2) not null,
TotalProfit decimal(9,2) not null,
```

BranchMangerID int not null
ManagerName nvarchar(max) not null
ManagerLastName nvarchar(max) not null)

Creating procedure:

```
create procedure RefreshBranchStatistics_View as
delete from BranchStatistics_View
insert into BranchStatistics_View
select Reservation_T.BranchID, Branch_T.BranchName, Branch_T.BranchDescription,
count(Reservation_T.ReservationID) as NumberOfReservations,
sum(Reservation_T.ReservationTotalAmount) as TotalRent,
sum(Reservation_T.ReservationTotalAmount)*0.15 as TotalProfit, Branch_T.BranchManagerID,
BranchManager_T.BranchManagerFirstName as ManagerName,
BranchManager_T.BranchManagerLastName as ManagerLastName
from Reservation_T, Branch_T, BranchManager_T
where Reservation_T.BranchID = Branch_T.BranchID and Branch_T.BranchManagerID =
BranchManager_T.BranchManagerID
group by Reservation_T.BranchID, Branch_T.BranchDescription, Branch_T.BranchManagerID,
Branch_T.BranchName,BranchManager_T.BranchManagerFirstName,
BranchManager_T.BranchManagerLastName

execute RefreshBranchStatistics_View
select* from BranchStatistics_View
```

Result:

Results | Messages

	BranchID	BranchName	BranchDescription	NumberOfReservations	TotalRent	TotalProfit	BranchMangerID	ManagerName	ManagerLastName
1	1	West Branch	West US Branch: California and Nevada	13	15402.00	2310.30	1	James	Wright
2	2	East Branch	East US Branch: New York, Washington	8	9729.00	1459.35	3	Brandon	Cooper
3	3	Southeast Branch	Southeast US Branch: Florida	4	3249.00	487.35	2	Michael	Amato

3.1.2 View 2. Overall Automobile Statistic

This view is created to show the overall statistic of the fleet of automobiles. It helps the general manager to analyze which cars are more popular, so in the future the platform can partner with more rental companies that have such cars. It shows how many times a particular automobile was reserved and for how much.

Creating table:

```
CREATE TABLE CarStatistics_View
```

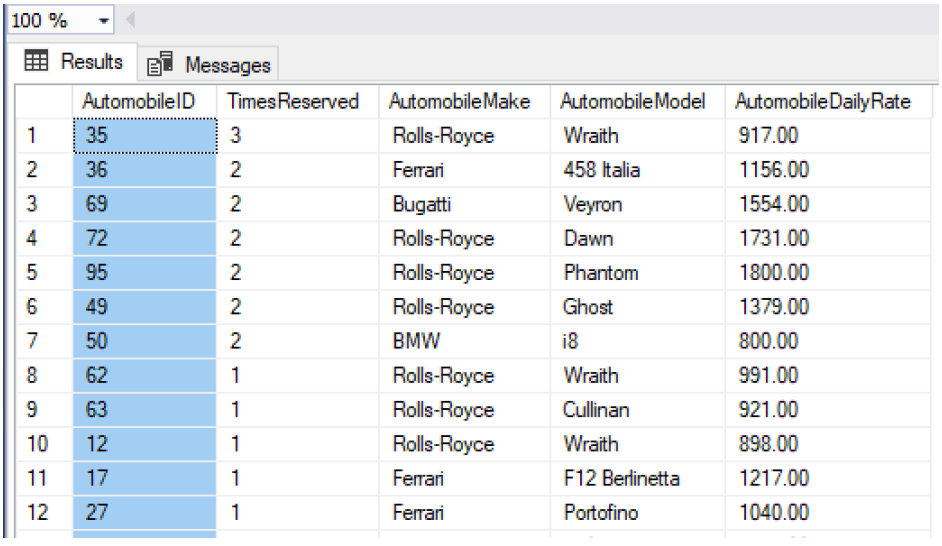
```
(AutomobileID int not null,
TimesReserved int not null,
AutomobileMake nvarchar(max) not null,
AutomobileModel nvarchar(max) not null,
AutomobileDailyRate decimal(6,2) not null)
```

Creating procedure:

```
create procedure RefreshCarStatistics_View as
delete from CarStatistics_View
insert into CarStatistics_View
select  Reservation_T.AutomobileID, count(Reservation_T.AutomobileID) as TimesReserved,
Automobile_T.AutomobileMake, Automobile_T.AutomobileModel,
Automobile_T.AutomobileDailyRate
from Reservation_T, Automobile_T
where Reservation_T.AutomobileID = Automobile_T.AutomobileID
group by Reservation_T.AutomobileID, Automobile_T.AutomobileMake,
Automobile_T.AutomobileModel,
Automobile_T.AutomobileDailyRate

execute RefreshCarStatistics_View
select* from CarStatistics_View order by TimesReserved desc
```

Result:



100 %

Results | Messages

	AutomobileID	TimesReserved	AutomobileMake	AutomobileModel	AutomobileDailyRate
1	35	3	Rolls-Royce	Wraith	917.00
2	36	2	Ferrari	458 Italia	1156.00
3	69	2	Bugatti	Veyron	1554.00
4	72	2	Rolls-Royce	Dawn	1731.00
5	95	2	Rolls-Royce	Phantom	1800.00
6	49	2	Rolls-Royce	Ghost	1379.00
7	50	2	BMW	i8	800.00
8	62	1	Rolls-Royce	Wraith	991.00
9	63	1	Rolls-Royce	Cullinan	921.00
10	12	1	Rolls-Royce	Wraith	898.00
11	17	1	Ferrari	F12 Berlinetta	1217.00
12	27	1	Ferrari	Portofino	1040.00

3.2 Branch Manager Views

3.2.1 View 1. Accidents statistic for the Southeast Branch

This view is created for the branch manager(Southeast Branch). While managing the branch it is important to see the statistic of accidents that happened to the cars. Also, this view shows whether it was an employee or customer who faced an accident.

Creating table:

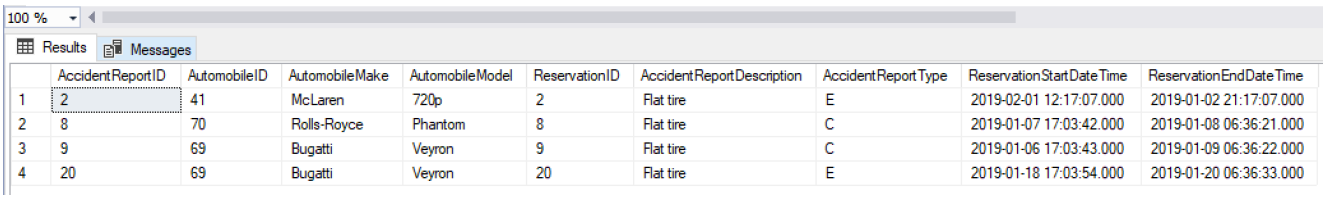
```
CREATE TABLE SoutheastBranchAccidents_View
(AccidentReportID int not null,
AutomobileID int not null,
AutomobileMake nvarchar(max) not null,
AutomobileModel nvarchar(max) not null,
ReservationID int not null,
AccidentReportDescription nvarchar(max) not null,
AccidentReportType char(1) not null,
ReservationStartDateTime datetime not null,
ReservationEndDateTime datetime not null)
```

Creating procedure:

```
create procedure RefreshSoutheastBranchAccidents_View as
delete from SoutheastBranchAccidents_View
insert into SoutheastBranchAccidents_View
select AccidentReport_T.AccidentReportID, AccidentReport_T.AutomobileID,
Automobile_T.AutomobileMake, Automobile_T.AutomobileModel,
AccidentReport_T.ReservationID,
AccidentReport_T.AccidentReportDescription, AccidentReport_T.AccidentReportType,
Reservation_T.ReservationStartDateTime,
Reservation_T.ReservationEndDateTime
from AccidentReport_T, Reservation_T, Automobile_T
where AccidentReport_T.ReservationID = Reservation_T.ReservationID and
Reservation_T.AutomobileID = Automobile_T.AutomobileID and BranchID = 3

execute RefreshSoutheastBranchAccidents_View
select* from SoutheastBranchAccidents_View
```

Result:



100 %

Results | Messages

	AccidentReportID	AutomobileID	AutomobileMake	AutomobileModel	ReservationID	AccidentReportDescription	AccidentReportType	ReservationStartDateTime	ReservationEndDateTime
1	2	41	McLaren	720p	2	Flat tire	E	2019-02-01 12:17:07.000	2019-01-02 21:17:07.000
2	8	70	Rolls-Royce	Phantom	8	Flat tire	C	2019-01-07 17:03:42.000	2019-01-08 06:36:21.000
3	9	69	Bugatti	Veyron	9	Flat tire	C	2019-01-06 17:03:43.000	2019-01-09 06:36:22.000
4	20	69	Bugatti	Veyron	20	Flat tire	E	2019-01-18 17:03:54.000	2019-01-20 06:36:33.000

3.2.2 View 2. Reservation Statistic for the West Branch

The view is created for the manager of the West Branch, and it helps to analyze the popularity of concierge and chauffeur services. Along with the total number of reservations for each reservation type, the manager can see the total earnings for those two types as well as their percentage of the total earnings in the branch that he manages.

Creating table:

```
CREATE TABLE WestBranchReservations_View
(ReservationType char(9) not null,
NumberOfReservations int not null,
Earnings decimal(9,2) not null,
PercentageOfTotalEarnings decimal(5,2) not null)
```

Creating procedure:

```
create procedure RefreshWestBranchReservations_View as
delete from WestBranchReservations_View
insert into WestBranchReservations_View
select ReservationType, count (ReservationType) as NumberOfReservations,
sum(Reservation_T.ReservationTotalAmount) as Earnings,
(sum(Reservation_T.ReservationTotalAmount)/Total)*100 as PercentageOfTotalEarnings
from Reservation_T, (select sum(ReservationTotalAmount) as Total from Reservation_T where
BranchID = 1) as derivedtable
where BranchID = 1
group by ReservationType, Total

execute RefreshWestBranchReservations_View
select* from WestBranchReservations_View
```

Result:

Results | Messages

	ReservationType	NumberOfReservations	Earnings	PercentageOfTotalEarnings
1	CHAUFFEUR	9	10897.00	70.75
2	CONCIERGE	4	4505.00	29.25

4. Database trigger

This trigger is very important, as it is used for logging any changes to the automobile daily rate from the table Automobile_T. Automobile daily rate is the main price maker that must not be altered freely. This information should stay protected from editing and that is why there is a need to log any activities regarding changing the daily rate.

Creating table where logs are located:

```
CREATE TABLE AutomobileRateUpdates_Log
(AutomobileID int,
AutomobileMake nvarchar(max),
AutomobileModel nvarchar(max),
OldDailyRate float,
NewDailyRate float,
UpdateDate datetime)
```

Creating trigger:

```
create trigger AutomobileRateUpdate on Automobile_T
for update
as
if update(AutomobileDailyRate)
begin
insert into AutomobileRateUpdates_Log (AutomobileID, AutomobileMake, AutomobileModel,
olddailyrate, newdailyrate, updatedate)
select inserted.AutomobileID, inserted.AutomobileMake, inserted.AutomobileModel,
deleted.AutomobileDailyRate, inserted.AutomobileDailyRate, GETDATE() from inserted, deleted
where inserted.AutomobileID = deleted.AutomobileID
end
```

Result:

Results Messages

	AutomobileID	AutomobileMake	AutomobileModel	OldDailyRate	NewDailyRate	UpdateDate
1	5	Porsche	GT3RS	1251	400	2019-03-20 16:15:03.743
2	20	Rolls-Royce	Phantom	1558	300	2019-03-20 16:15:23.300
3	22	Rolls-Royce	Dawn	1197	250	2019-03-20 16:16:10.317