PROBLEM DEFINITION:

Databases are very important for storing, controlling and processing data. Travel agencies need to hide customers,agencies,agents,tours and all other information have been systematized.In this project, I decided to make a database system of a company that has more than one travel agency. Like ETS Tour.

USER REQUIREMENTS:

Store information about customers, including their ID, name, surname, contact details, gender, registration date, and address.

Store information about cities, including their ID and name.

Store information about hotels, including their ID, room ID, name, city ID, price, and telephone number.

Store information about drivers, including their ID, name, contact information, address, salary, and license details.

Store information about tours, including their ID, name, location, route, time, duration, and capacity.

Store information about buses, including their ID, driver ID, tour ID, name, seating capacity, availability, and destination.

Store booking information, including the booking ID, customer ID, bus ID, hotel ID, booking date, total day, room ID, price, start date, end date, and city name.

Store information about travel agencies, including their ID, name, city ID, phone, address, city, and email.

Store information about staff members working in the agency, including their ID, agency ID, agency telephone number, address, job code, title, and salary.

Store information about agents, including their ID, tour ID, agency ID, name, contact details, and salary.

Store information about user logins, including the user number, username, and user password.It is necessary to provide tour-customer relation. Each customer can attend many tours.

If tour requires to customer stay somewhere, the information of hotel must be stored

There is a many-to-many relationship between customers and agents. Agent id and customer id should be stored in order to keep this relationship.

Allow administrators to add, modify, and delete tour packages.

Allow customers to make reservations for specific dates and manage their bookings.

A new travel agency can be added, deleted or updated.

A new tour can be added, deleted or updated.

A new customer can be added, deleted or updated.

A new staff can be added, deleted or updated.

A new agent can be added, deleted or update.

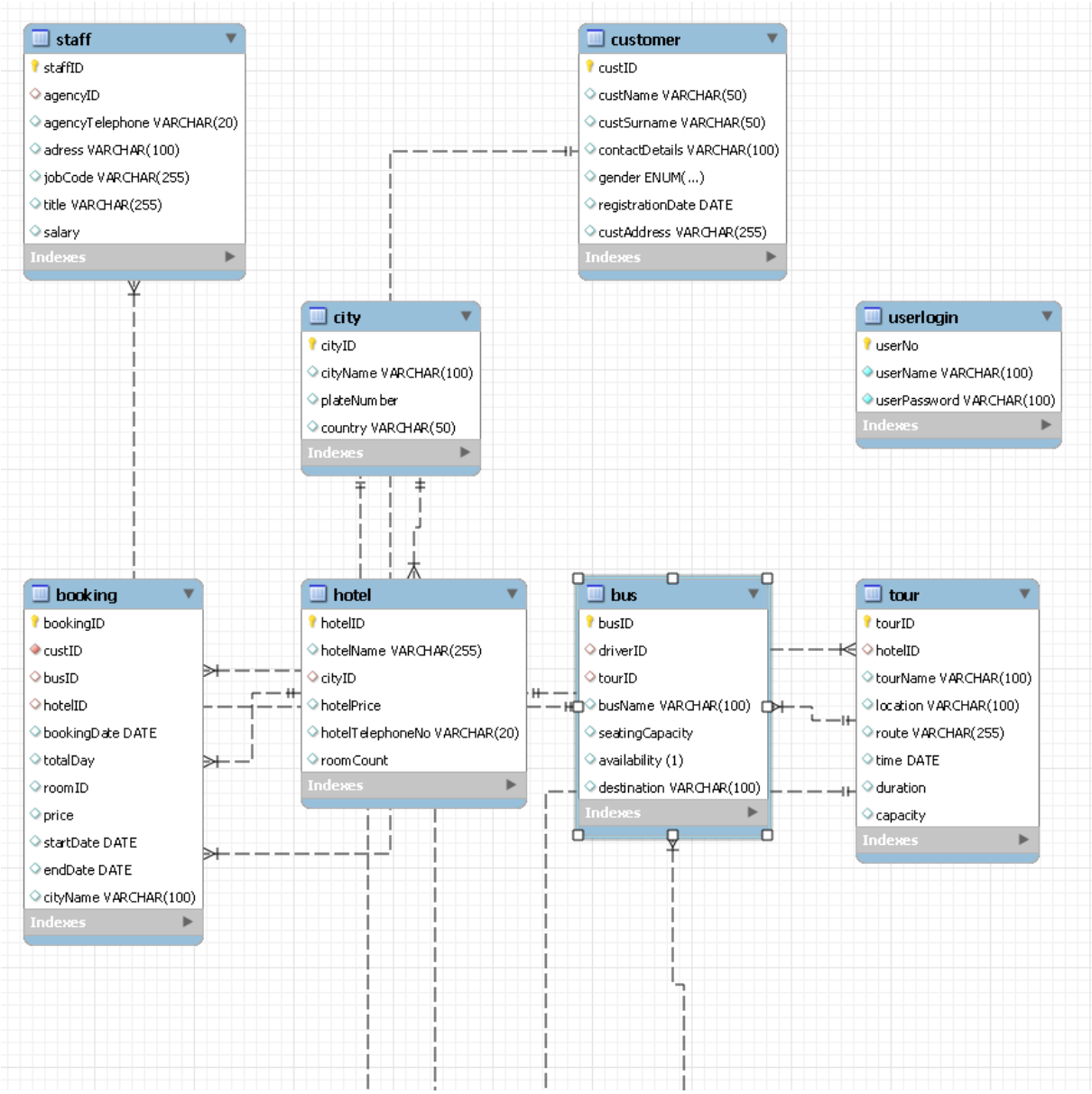
A new hotel can be added.

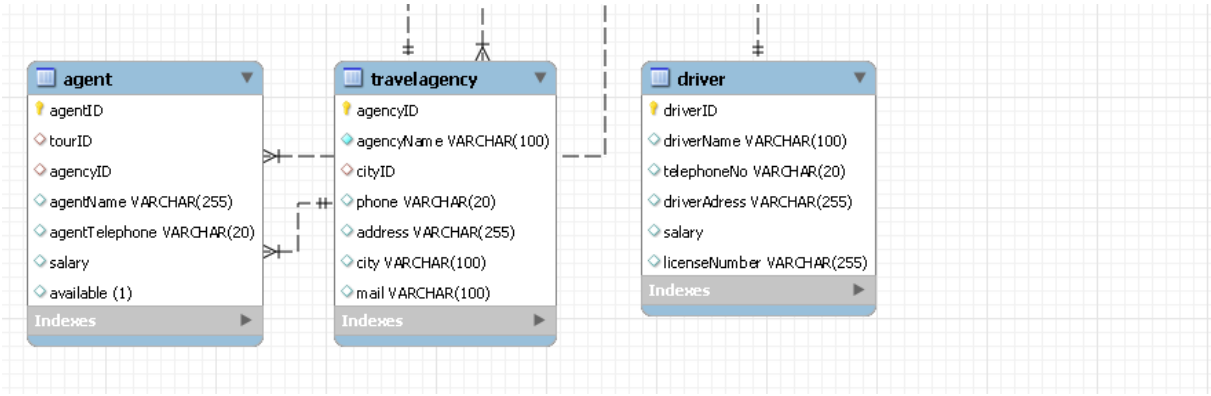
A new bus can be added.

A new driver can be added

A new hotel can be added.

ER Diagram





TABLES

Customer:

Information about customers.

```sql
CREATE TABLE customer (
    custID INT AUTO_INCREMENT PRIMARY KEY,
    custName VARCHAR(50),
    custSurname VARCHAR(50),
    contactDetails VARCHAR(100),
    gender ENUM('Male', 'Female'),
    registrationDate DATE,
    custAddress VARCHAR(255)
);
```

```sql
INSERT INTO customer (custName, custSurname, contactDetails, gender, registrationDate, custAddress)
VALUES
('Alperen', 'Karaçete', 'alperen@hotmail.com', 'Male', '2023-06-13', 'Tuzla/İstanbul'),
('Semih', 'Karaçete', 'semih@example.com', 'Female', '2023-06-26', 'Yalova/Merkez');

SELECT * FROM customer;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

custID	custName	custSurname	contactDetails	gender	registrationDate	custAddress
1	Alperen	Karaçete	alperen@hotmail.com	Male	2023-06-13	Tuzla/İstanbul
2	Semih	Karaçete	semih@example.com	Female	2023-06-26	Yalova/Merkez
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Booking:

Information about customer booking.

```sql
CREATE TABLE booking (
    bookingID INT AUTO_INCREMENT PRIMARY KEY,
    custID INT NOT NULL,
    busID INT,
    hotelID INT,
    bookingDate DATE,
    totalDay INT,
    roomID INT,
    price INT,
    startDate DATE,
    endDate DATE,
    cityName VARCHAR(100),
    CONSTRAINT fkBookingCustomer FOREIGN KEY (custID) REFERENCES customer (custID),
    CONSTRAINT fkBookingHotel FOREIGN KEY (hotelID) REFERENCES hotel (hotelID),
    CONSTRAINT fkBookingBus FOREIGN KEY (busID) REFERENCES bus (busID)
);
```

```sql
INSERT INTO booking (custID, busID, hotelID, bookingDate, totalDay, roomID, price, startDate, endDate, cityName) VALUES
(1, 1, 1, '2023-07-01', 7, 101, 4000, '2023-07-15', '2023-07-22', 'Antalya'),
(2, 2, 2, '2023-08-10', 2, 201, 3500, '2023-08-20', '2023-08-22', 'Istanbul');

select * from booking
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

bookingID	custID	busID	hotelID	bookingDate	totalDay	roomID	price	startDate	endDate	cityName
1	1	1	1	2023-07-01	7	101	4000	2023-07-15	2023-07-22	Antalya
2	2	2	2	2023-08-10	2	201	3500	2023-08-20	2023-08-22	Istanbul
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

City:

Information about cities.

```sql
CREATE TABLE city (
    cityID INT AUTO_INCREMENT PRIMARY KEY,
    cityName VARCHAR(100),
    plateNumber INT
);
```

```sql
ALTER TABLE city ADD country VARCHAR(50)
```

```sql
INSERT INTO city (cityName, plateNumber, country) VALUES
('Adana', 1,'Turkey'),
('Adıyaman', 2,'Turkey'),
('Afyonkarahisar', 3,'Turkey'),
('Ağrı', 4,'Turkey'),
('Amasya', 5,'Turkey'),
('Ankara', 6,'Turkey'),
('Antalya', 7,'Turkey'),
('Artvin', 8,'Turkey'),
('Aydın', 9,'Turkey'),
('Balıkesir', 10,'Turkey'),
('Bilecik', 11,'Turkey'),
('Bingöl', 12,'Turkey'),
('Bitlis', 13,'Turkey'),
('Bolu', 14,'Turkey'),
('Burdur', 15,'Turkey'),
('Bursa', 16,'Turkey'),
('Çanakkale', 17,'Turkey'),
('Çankırı', 18,'Turkey'),
('Çorum', 19,'Turkey'),
('Denizli', 20,'Turkey'),
```

Result Grid | Filter Rows: | Edit: | Export/Import:

cityID	cityName	plateNumber	country
1	Adana	1	Turkey
2	Adıyaman	2	Turkey
3	Afyonkarahisar	3	Turkey
4	Ağrı	4	Turkey
5	Amasya	5	Turkey
6	Ankara	6	Turkey
7	Antalya	7	Turkey
8	Artvin	8	Turkey
9	Aydın	9	Turkey

```
('Paris', 75000,'French'),
('New York', 10001,'USA'),
('Tokyo', 100,'Japan'),
('Sydney', 2000,'Australia'),
('Rome', '00100','Italy'),
('Moscow', 101000,'Russia'),
('Rio de Janeiro', 20000,'Brezilya'),
('Cape Town', 8000,'South Africa');

SELECT * FROM city;
```

Result Grid | Filter Rows: | Edit:

cityID	cityName	plateNumber	country
82	Paris	75000	French
83	New York	10001	USA
84	Tokyo	100	Japan
85	Sydney	2000	Australia
86	Rome	100	Italy
87	Moscow	101000	Russia
88	Rio de Janeiro	20000	Brezilya
89	Cape Town	8000	South ...
NULL	NULL	NULL	NULL

Hotel

Information about hotels.

```
CREATE TABLE hotel (
    hotelID INT AUTO_INCREMENT PRIMARY KEY,
    hotelName VARCHAR(255),
    cityID INT,
    hotelPrice INT,
    hotelTelephoneNo VARCHAR(20),
    roomCount INT,
    CONSTRAINT fkHotelCity FOREIGN KEY (cityID) REFERENCES city (cityID)
);
```

```sql
INSERT INTO hotel (hotelName, cityID, hotelPrice, hotelTelephoneNo,roomCount) VALUES
('Estera Silent Hotel', 7, 2100, '0552 699 76 70',300),
('Miss City Hotel', 34, 3200, '0536 615 61 75',150),
('Palan Hotel', 25, 1500, '0534 054 27 45',145);

SELECT * FROM hotel;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

hotelID	hotelName	cityID	hotelPrice	hotelTelephoneNo	roomCount
1	Estera Silent Hotel	7	2100	0552 699 76 70	300
2	Miss City Hotel	34	3200	0536 615 61 75	150
3	Palan Hotel	25	1500	0534 054 27 45	145
NULL	NULL	NULL	NULL	NULL	NULL

Driver

Information about drivers

```sql
CREATE TABLE driver (
    driverID INT AUTO_INCREMENT PRIMARY KEY,
    driverName VARCHAR(100),
    telephoneNo VARCHAR(20),
    driverAdress VARCHAR(255),
    salary INT,
    licenseNumber VARCHAR(255)
);
```

```sql
INSERT INTO driver (driverName, telephoneNo, driverAdress, salary, licenseNumber) VALUES
('Burak Bulut', '1234567890', 'Aydıntepe Mahallesi/İstanbul', 15000, '123456'),
('Ayşe Kiler', '9876543210', 'Palandöken Mahallesi/Erzurum', 15000, '845632'),
('Kevser Söz', '7890123456', 'Big Canyon/Oklahama', 15000, '485321');

SELECT * FROM driver;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

driverID	driverName	telephoneNo	driverAdress	salary	licenseNumber
1	Burak Bulut	1234567890	Aydıntepe Mahallesi/İstanbul	15000	123456
2	Ayşe Kiler	9876543210	Palandöken Mahallesi/Erzurum	15000	845632
3	Kevser Söz	7890123456	Big Canyon/Oklahama	15000	485321
NULL	NULL	NULL	NULL	NULL	NULL

Tour

Information about tours.

```sql
CREATE TABLE tour (
    tourID INT AUTO_INCREMENT PRIMARY KEY,
    hotelID INT,
    tourName VARCHAR(100),
    location VARCHAR(100),
    route VARCHAR(255),
    time DATE,
    duration INT,
    capacity INT,
    CONSTRAINT fktourhotel FOREIGN KEY (hotelID) REFERENCES hotel (hotelID)
);
```

```sql
INSERT INTO tour (hotelID, tourName, location, route, time, duration, capacity) VALUES
(1, 'Antalya Beach Tour', 'Antalya', 'Beach, Old Town, Waterfalls', '2023-07-15', 7, 50),
(2, 'Istanbul Historical Tour', 'Istanbul', 'Hagia Sophia, Blue Mosque, Topkapi Palace', '2023-08-20', 2, 40),
(3, 'Erzurum Ski Trip', 'Erzurum', 'Palandoken Ski Resort, Cifte Minareli Medrese', '2023-09-10', 5, 30),
(NULL, 'Trabzon City Tour', 'Trabzon', 'Ataturk\'s Villa, Sumela Monastery, Uzungol', '2023-10-05', 1, 25);

SELECT * FROM tour;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

tourID	hotelID	tourName	location	route	time	duration	capacity
1	1	Antalya Beach Tour	Antalya	Beach, Old Town, Waterfalls	2023-07-15	7	50
2	2	Istanbul Historical Tour	Istanbul	Hagia Sophia, Blue Mosque, Topkapi Palace	2023-08-20	2	40
3	3	Erzurum Ski Trip	Erzurum	Palandoken Ski Resort, Cifte Minareli Medrese	2023-09-10	5	30
4	NULL	Trabzon City Tour	Trabzon	Ataturk's Villa, Sumela Monastery, Uzungol	2023-10-05	1	25
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

BUS

Information about tour busses.

```sql
CREATE TABLE bus (
    busID INT AUTO_INCREMENT PRIMARY KEY,
    driverID INT,
    tourID INT,
    busName VARCHAR(100),
    seatingCapacity INT,
    availability BOOLEAN,
    destination VARCHAR(100),
    CONSTRAINT fkBusDriver FOREIGN KEY (driverID) REFERENCES driver (driverID),
    CONSTRAINT fkBusTour FOREIGN KEY (tourID) REFERENCES tour (tourID)
);
```

```sql
INSERT INTO bus (driverID, tourID, busName, seatingCapacity, availability, destination) VALUES
(1, 1, 'Bus 1', 50, true, 'Antalya'),
(2, 2, 'Bus 2', 50, true, 'Istanbul'),
(3, 3, 'Bus 3', 50, true, 'Erzurum'),
(1, 4, 'Bus 4', 50, true, 'Trabzon');

SELECT * FROM bus;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

busID	driverID	tourID	busName	seatingCapacity	availability	destination
1	1	1	Bus 1	50	1	Antalya
2	2	2	Bus 2	50	1	Istanbul
3	3	3	Bus 3	50	1	Erzurum
4	1	4	Bus 4	50	1	Trabzon
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Booking

Information about customer bookings.

```sql
CREATE TABLE booking (
    bookingID INT AUTO_INCREMENT PRIMARY KEY,
    custID INT NOT NULL,
    busID INT,
    hotelID INT,
    bookingDate DATE,
    totalDay INT,
    roomID INT,
    price INT,
    startDate DATE,
    endDate DATE,
    cityName VARCHAR(100),
    CONSTRAINT fkBookingCustomer FOREIGN KEY (custID) REFERENCES customer (custID),
    CONSTRAINT fkBookingHotel FOREIGN KEY (hotelID) REFERENCES hotel (hotelID),
    CONSTRAINT fkBookingBus FOREIGN KEY (busID) REFERENCES bus (busID)
);
```

```sql
INSERT INTO booking (custID, busID, hotelID, bookingDate, totalDay, roomID, price, startDate, endDate, cityName) VALUES
(1, 1, 1, '2023-07-01', 7, 101, 4000, '2023-07-15', '2023-07-22', 'Antalya'),
(2, 2, 2, '2023-08-10', 2, 201, 3500, '2023-08-20', '2023-08-22', 'Istanbul');

select * from booking
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

bookingID	custID	busID	hotelID	bookingDate	totalDay	roomID	price	startDate	endDate	cityName
1	1	1	1	2023-07-01	7	101	4000	2023-07-15	2023-07-22	Antalya
2	2	2	2	2023-08-10	2	201	3500	2023-08-20	2023-08-22	Istanbul
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Travel Agency

Information about travel agencies.

```sql
INSERT INTO travelagency (agencyName, cityID, phone, address, city, mail) VALUES
('KARA TOUR', 1, '0539 999 99 99', 'Lara 10', 'Antalya', 'karaagencyantalya@example.com'),
('KARA TOUR', 2, '0538 888 88 88', 'Tuzla İstasyon', 'Istanbul', 'karaagencyistanbul@example.com');

SELECT * FROM travelagency;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

agencyID	agencyName	cityID	phone	address	city	mail
1	KARA TOUR	1	0539 999 99 99	Lara 10	Antalya	karaagencyantalya@example.com
2	KARA TOUR	2	0538 888 88 88	Tuzla İstasyon	Istanbul	karaagencyistanbul@example.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
INSERT INTO agent (tourID, agencyID, agentName, agentTelephone, salary,available) VALUES
(1, 1, 'Burak Aksu', '0532 777 77 77', 20000,false),
(2, 1, 'Sevda Hacıoğlu', '0531 222 22 22', 20000,false),
(3, 2, 'Duran Yılmaz', '0541 111 11 11', 20000,false),
(4, 2, 'Gökay Türe', '0554 333 33 33', 20000,false),
(NULL, 2, 'Berke Geçmen', '0554 666 66 66', 20000,true);

SELECT * FROM agent;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

agentID	tourID	agencyID	agentName	agentTelephone	salary	available
1	1	1	Burak Aksu	0532 777 77 77	20000	0
2	2	1	Sevda Hacıoğlu	0531 222 22 22	20000	0
3	3	2	Duran Yılmaz	0541 111 11 11	20000	0
4	4	2	Gökay Türe	0554 333 33 33	20000	0
5	NULL	2	Berke Geçmen	0554 666 66 66	20000	1

Staff

Information about staff in tour agency.

```sql
CREATE TABLE staff (
    staffID INT AUTO_INCREMENT PRIMARY KEY,
    agencyID INT,
    agencyTelephone VARCHAR(20),
    adress VARCHAR(100),
    jobCode VARCHAR(255),
    title VARCHAR(255),
    salary INT,
    CONSTRAINT fkStaffAgency FOREIGN KEY (agencyID) REFERENCES travelagency (agencyID)
);
```

```sql
INSERT INTO staff (staffID, agencyID, agencyTelephone, adress, jobCode, title, salary) VALUES
(1, 1, '0539 999 99 99', 'Kağıthane/İstanbul', 'JC001', 'Cleaner', 8500);

SELECT * FROM staff;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

staffID	agencyID	agencyTelephone	adress	jobCode	title	salary
1	1	0539 999 99 99	Kağıthane/İstanbul	JC001	Cleaner	8500
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Agent:

Information about tour agents.

```sql
CREATE TABLE agent (
    agentID INT AUTO_INCREMENT PRIMARY KEY,
    tourID INT,
    agencyID INT,
    agentName VARCHAR(255),
    agentTelephone VARCHAR(20),
    salary INT,
    available BOOLEAN,
    CONSTRAINT fkAgentTour FOREIGN KEY (tourID) REFERENCES tour (tourID),
    CONSTRAINT fkAgentAgency FOREIGN KEY (agencyID) REFERENCES travelagency (agencyID)
);
```

```sql
INSERT INTO agent (tourID, agencyID, agentName, agentTelephone, salary,available) VALUES
(1, 1, 'Burak Aksu', '0532 777 77 77', 20000,false),
(2, 1, 'Sevda Hacıoğlu', '0531 222 22 22', 20000,false),
(3, 2, 'Duran Yılmaz', '0541 111 11 11', 20000,false),
(4, 2, 'Gökay Türe', '0554 333 33 33', 20000,false),
(NULL, 2, 'Berke Geçmen', '0554 666 66 66', 20000,true);

SELECT * FROM agent;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

agentID	tourID	agencyID	agentName	agentTelephone	salary	available
1	1	1	Burak Aksu	0532 777 77 77	20000	0
2	2	1	Sevda Hacıoğlu	0531 222 22 22	20000	0
3	3	2	Duran Yılmaz	0541 111 11 11	20000	0
4	4	2	Gökay Türe	0554 333 33 33	20000	0
5	NULL	2	Berke Geçmen	0554 666 66 66	20000	1

User

Adding admin user to database.

```sql
INSERT INTO userlogin (userName, userPassword) VALUES
('admin', 'password');

SELECT * FROM userlogin;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

userNo	userName	userPassword
1	admin	password
NULL	NULL	NULL

```sql
INSERT INTO userlogin (userName, userPassword) VALUES
('admin', 'password');

SELECT * FROM userlogin;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

userNo	userName	userPassword
1	admin	password
NULL	NULL	NULL

Functional dependecies

customer:

custID → custName, custSurname, contactDetails, gender, registrationDate, custAddress

city:

cityID → cityName, plateNumber

hotel:

hotelID → roomID, hotelName, cityID, hotelPrice, hotelTelephoneNo

driver:

driverID → driverName, telephoneNo, driverAddress, salary, licenseNumber

tour:

tourID → hotelID, tourName, location, route, time, duration, capacity

bus:

busID → driverID, tourID, busName, seatingCapacity, availability, destination

booking:

bookingID → custID, busID, hotelID, bookingDate, totalDay, roomID, price, startDate, endDate, cityName

travelagency:

agencyID → agencyName, cityID, phone, address, city, mail

staff:

staffID → agencyID, agencyTelephone, address, jobCode, title, salary

agent:

agentID → tourID, agencyID, agentName, contact, salary

userlogin:

userNo → userName, userPassword

Table Attributes

customer table:

custID: int (auto-increment, primary key)

custName: varchar(50)

custSurname: varchar(50)

contactDetails: varchar(100)

gender: enum('Male', 'Female')

registrationDate: date

custAddress: varchar

city table:

cityID: int (primary key)

cityName: varchar(100)

plateNumber INT

hotel table:

hotelID: int (auto-increment, primary key)

roomID: int

hotelName: varchar(255)

cityID: int (foreign key referencing city table)

hotelPrice: int

hotelTelephoneNo: varchar(20)

driver table:

driverID: int (auto-increment, primary key)

driverName: varchar(100)

telephoneNo: varchar(20)

driverAddress: varchar(255)

salary: int

licenseNumber: varchar(255)

tour table:

tourID: int (auto-increment, primary key)

tourName: varchar(100)

location: varchar(100)

route: varchar(255)

time: date

duration: int

capacity: int

bus table:

busID: int (auto-increment, primary key)

driverID: int (foreign key referencing driver table)

tourID: int (foreign key referencing tour table)

busName: varchar(100)

seatingCapacity: int

availability: boolean

destination: varchar(100)

booking table:

bookingID: int (auto-increment, primary key)

custID: int (foreign key referencing customer table)

busID: int (foreign key referencing bus table)

hotelID: int (foreign key referencing hotel table)

bookingDate: date

totalDay: int

roomID: int

price: int

startDate: date

endDate: date

cityName: varchar(100)

travelagency table:

agencyID: int (primary key)

agencyName: varchar(100)

cityID: int (foreign key referencing city table)

phone: varchar(20)

address: varchar(255)

city: varchar(100)

mail: varchar(100)

staff table:

staffID: int (auto-increment, primary key)

agencyID: int (foreign key referencing travelagency table)

agencyTelephone: varchar(20)

address: varchar(100)

jobCode: varchar(255)

title: varchar(255)

salary: int

agent table:

agentID: int (auto-increment, primary key)

tourID: int (foreign key referencing tour table)

agencyID: int (foreign key referencing travelagency table)

agentName: varchar(255)

contact: varchar(255)

salary: int

userlogin table:

userNo: int (auto-increment, primary key)

userName: varchar(100) (not null)

userPassword: varchar(100) (not null)

TRIGGERS:

Increase roomCount when a booking is canceled.

```
DELIMITER //

CREATE TRIGGER deleteBookingUpdateRooms
AFTER DELETE ON booking
FOR EACH ROW
BEGIN
    IF OLD.hotelID IS NOT NULL THEN
        UPDATE hotel SET roomCount = roomCount + 1 WHERE hotelID = OLD.hotelID;
    END IF;
END//

DELIMITER ;
```

Decrease roomCount when a booking is completed.

```
DELIMITER //

CREATE TRIGGER insertBookingUpdateRooms
AFTER INSERT ON booking
FOR EACH ROW
BEGIN
    IF NEW.hotelID IS NOT NULL THEN
        UPDATE hotel SET roomCount = roomCount - 1 WHERE hotelID = NEW.hotelID;
    END IF;
END//

DELIMITER ;
```

Decrease tour capacity when someone register

```
DELIMITER //

CREATE TRIGGER updateCapacityOfTourWhenRegister
AFTER INSERT ON tour
FOR EACH ROW
BEGIN
    IF NEW.capacity IS NOT NULL THEN
        UPDATE tour SET capacity = capacity - 1 WHERE tourID = NEW.tourID;
    END IF;
END//

DELIMITER ;
```

Increase tour capacity when someone cancel registration.

```
DELIMITER //

CREATE TRIGGER updateCapacityOfTourWhenDelete
AFTER DELETE ON tour
FOR EACH ROW
BEGIN
    IF OLD.capacity IS NOT NULL THEN
        UPDATE tour SET capacity = capacity + 1 WHERE tourID = OLD.tourID;
    END IF;
END//

DELIMITER ;
```

When tour totalDay changes, with this trigger booking duration too changed.

```
DELIMITER //

CREATE TRIGGER updateBookingTotalDay
AFTER UPDATE ON tour
FOR EACH ROW
BEGIN

    IF OLD.duration <> NEW.duration THEN

        UPDATE booking
        SET totalDay = NEW.duration
        WHERE tourID = NEW.tourID;
    END IF;
END//

DELIMITER ;
```

When I user books a trip, seatingCapacity on related bus is decreases.

```
DELIMITER //

CREATE TRIGGER decreaseSeatingCapacity
AFTER INSERT ON booking
FOR EACH ROW
BEGIN
    IF (SELECT seatingCapacity FROM bus WHERE busID = NEW.busID) = 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot insert tour. Bus seating capacity is already 0.';
    ELSE
        UPDATE bus SET seatingCapacity = seatingCapacity - 1 WHERE busID = NEW.busID;
    END IF;
END//

DELIMITER ;
```

When I user cancels a trip, seatingCapacity on related bus is incrases.

```
DELIMITER //

CREATE TRIGGER IncreaseSeatingCapacity
AFTER DELETE ON booking
FOR EACH ROW
BEGIN
    UPDATE bus SET seatingCapacity = seatingCapacity + 1 WHERE busID = OLD.busID;
END//

DELIMITER ;
```

Transactions

This transaction controls if there is a tourID with given ID if there is it deletes. If not It prints screen error.

```
DELIMITER //

CREATE PROCEDURE deleteBooking(
    IN tempBookingID INT
)
BEGIN
    DECLARE bookingCount INT;

    START TRANSACTION;

    SELECT COUNT(*) INTO bookingCount FROM booking WHERE bookingID = tempBookingID;

    IF bookingCount > 0 THEN
        DELETE FROM booking WHERE bookingID = tempBookingID;
        COMMIT;
    ELSE
        ROLLBACK;
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error deleting booking!';
    END IF;

END//

DELIMITER ;
```

We can CALL deleteBooking(bookingID);

Procedures

Update all staff salaries for % rate.

```
DELIMITER //

CREATE PROCEDURE updateStaffSalaries(
    IN newSalaryRate INT
)
BEGIN
    UPDATE staff SET salary = salary + newSalaryRate*salary/100 WHERE staffID > 0;
END//

DELIMITER ;
```

Change route of tour.

```
DELIMITER //

CREATE PROCEDURE updateTourRoute(
    IN tempTourID INT,
        newRoute VARCHAR(255)
)
BEGIN
    UPDATE tour SET route = newRoute WHERE tourID = tempTourID;
END//

DELIMITER ;
```

Update customer name.

```
DELIMITER //

CREATE PROCEDURE updateCustomerName(
    IN tempCustomerID INT,
        newName VARCHAR(255)
)
BEGIN
    UPDATE customer SET custName = newName WHERE custID = tempCustomerID;
END//

DELIMITER ;
```