SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİTABANI YÖNETİM SİSTEMLERİ

Proje

Öğrenci adı: Bedru Umer

Öğrenci soyadı: Mohammed

Öğrenci No: B191210557

Şübe: 1B

Eposta: bedru.mohammed@ogr.sakarya.edu.tr

Uygulamanın tanıtımı

Bu Uygulama bir seyahat şirketine ait biletleri yolculara sunmaktadır. Yoluclar kalkış ve varış şehirleri seçip kolaylıkla bilet alabilmelerini sağılamayan uygulama geliştirdim. Ayrıca şirketin otobüsleri, şoförleri, ve seferlerin yönetmeliği kapsamaktadır. Yani hem yolcu hem de yönetici için hizmet veren uygulamadır.

İş Kuralları

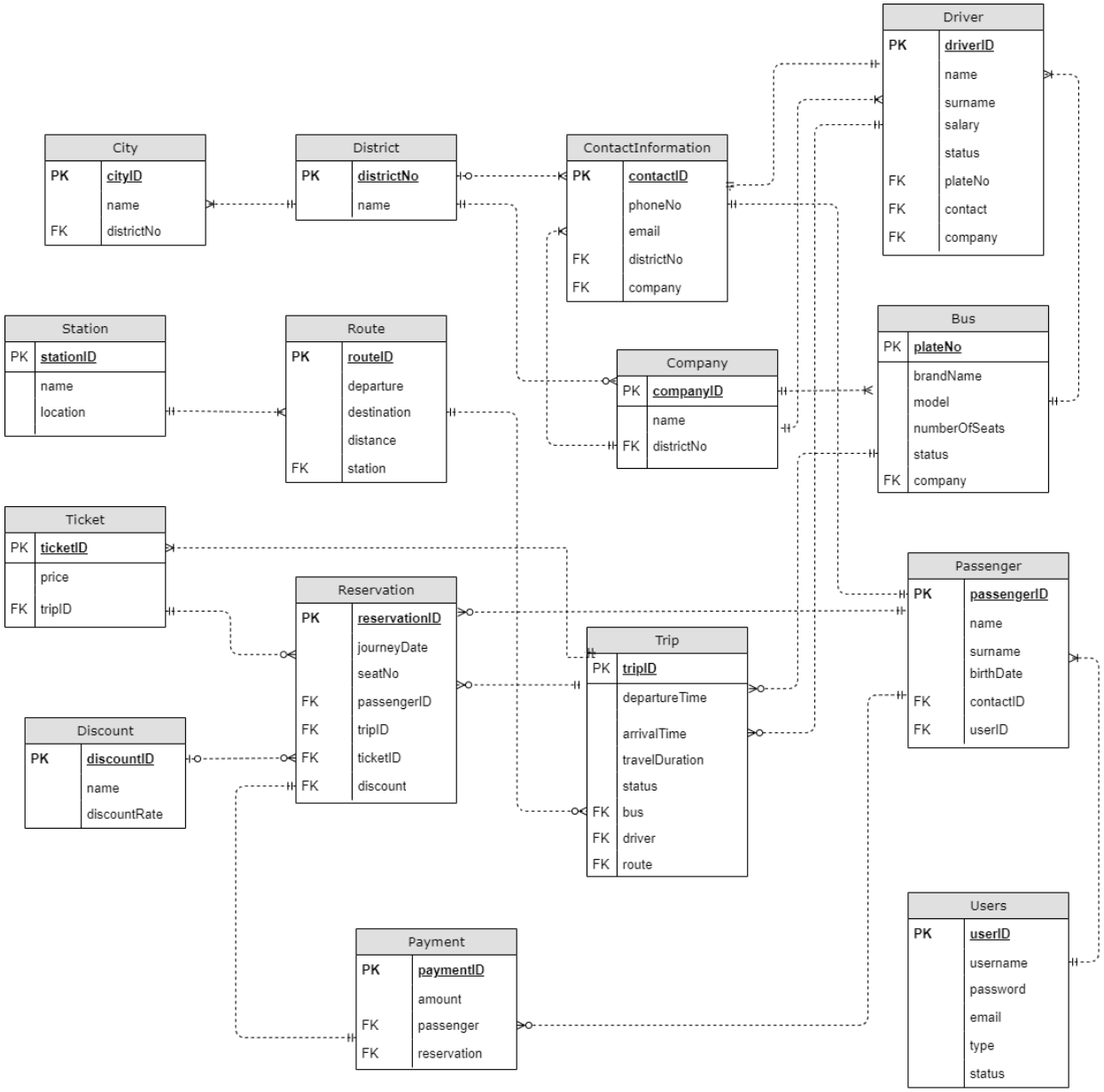
- Otobüs şirketinin adı, adresi ve iletişim bilgileri vardır.
- Otobüsün markası, modeli, koltuk sayısı ve durum bilgileri vardır.
- Bir otobüs yalnızca bir şirkete aittir. Otobüs şirketin çok sayıda otobüsleri vardır.
- Bir şoförün adı, soyadı, geliri ve durumu saklanır.
- Bir şoför bir şirkette çalışır. Otobüs şirketin çok sayıda şoförleri vardır.
- Bir şoför yalnızca bir otobüs sürerken bir otobüsün birden fazla şoförleri olabilir.
- Bir ilde çok sayıda ilçeler vardır.
- İletişim bilgileri içerisinde Telefon numarası, eposta ve adres bilgileri mevcuttur.
- Güzergah bilgileri içerisinde kalkış ve varış noktaları, uzaklık mevcuttur.
- Stasyon bilgileri içerisinde stasyonun adı ve konumu vardır.
- Bir güzergahın varış noktasında bir stasyonu varken bir stasyonun çok sayıda güzergahı olabilir.
- Seferler bilgileri içerisinde kalkış ve varış saati, yolculuğun süresi saklanmalı.
- Bir güzergahın çok sayıda seferleri vardır. Bir seferin sadece bir güzergahı vardır.
- Bir şöförün çok sayıda seferleri bulunabilir ama bir sefer sadece bir şoföre aittir.
- Bir otobüsün çok sayıda seferleri bulunabilir ama bir sefer sadece bir otobüse aittir.
- Rezervasyon bilgileri içerisinde yolculuğun tarihi ve saati, koltuk numarası bilgileri saklanır.
- Bilet bilgileri içerisinde baz fiyatı saklanır.
- Bir biletin bir seferi varken bir seferin fazla bileti bulunabilir.
- Bir biletin indirimi olabilir. Bir indirim tipi çok sayıda bilet için hesaplanabilir.
- Bir rezervasyon için yalnızca bir bilet hazırlanır ama bir bilet çok sayıda rezerve edilebilir.
- Bir sefer çok sayıda rezerve edilebilir ama bir rezervasyon sadece bir sefer için yapılabilir.
- Yolcu bilgileri içerisinde adı, soyadı, kimlik numarası, cinsiyeti, döğüm tarihi, iletişim bilgileri ve kullanıcı bilgileri saklanır.
- Bir yolcu çok sayıda rezervasyon yapabilir ama bir rezervasyon sadece bir yolcu tarafından yapılır.

- Ödeme bilgileri içersinde toplam fiyatı saklanır.
- Bir yolcu çok sayıda ödemeler yapabilir ama bir ödeme sadece bir yolcu tarafından yapılır.
- Bir rezervasyon için ödeme yapılır.

İlişisel şeması

- Company (**companyID: Serial**, name: varchar, districtNo: int)
- Bus(**plateNo:int**, brandName: varchar, model: varchar, numberOfSeats: smallint, status: varchar, company: int)
- Driver(**driverID: serial,** name: varchar, surname: varchar, salary:int, status: varchar plateNo: int, company: int, company: int)
- Passenger(**passengerID: serial,** name: varchar, surname: varchar, birthdate: date, contactID: int, userID: int)
- Users(**userID: serial,** username: varchar, password: varchar, email: varchar, type: char(1), status: varchar)
- ContactInformation(**contactID: serial,** phone: varchar, email: varchar, district: int, company: int)
- District(**districtNo: serial,** name: varchar)
- City(**cityID: serial,** name: varchar, districtNo: int)
- Route(**routeID: serial,** departure: varchar, destination: varchar, distance: int, station: int)
- Station(**stationID: serial,** name: varchar, location: varchar)
- Trip(**tripID: serial,** departureTime: time, arrivalTime: time, travelDuration: varchar, status: varchar, bus: int, driver: int, route: int)
- Reservation(**reservationID: serial,** journeyDate: date, seatNo: smallint, passengerID: int, tripID: int, ticketID: int, discount: int)
- Ticket(**ticketID: serial,** price: money, trip: int)
- Payment(**paymentID: serial,** amount: int, passenger: int, reservation: int)
- Discount(**discountID: serial,** name: varchar, discountRate: real)

VARLIK BAĞINTI DİAGRAMI



SQL kodu

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 13.4

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: OnlineBusReservation; Type: DATABASE; Schema:
-; Owner: postgres
--

CREATE DATABASE "OnlineBusReservation" WITH TEMPLATE =
template0 ENCODING = 'UTF8' LOCALE =
'Turkish_Turkey.1254';


ALTER DATABASE "OnlineBusReservation" OWNER TO
postgres;

\connect "OnlineBusReservation"
```

```
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: public; Type: SCHEMA; Schema: -; Owner:
postgres
--

CREATE SCHEMA public;


ALTER SCHEMA public OWNER TO postgres;

--
-- Name: SCHEMA public; Type: COMMENT; Schema: -;
Owner: postgres
--

COMMENT ON SCHEMA public IS 'standard public schema';


--
-- Name: addContactInformation(); Type: FUNCTION;
Schema: public; Owner: postgres
```

```
--

CREATE FUNCTION public."addContactInformation"()
RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    IF NEW."email" NOT LIKE '%@%' THEN
            NEW."email"
            := CONCAT(NEW.email,
            '@epixtravels.com');
    END IF;
    RETURN NEW;
END;
$$;


ALTER FUNCTION public."addContactInformation"() OWNER
TO postgres;

--
-- Name: addDriverTR1(); Type: FUNCTION; Schema:
public; Owner: postgres
--

CREATE FUNCTION public."addDriverTR1"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    NEW."name" = LTRIM(NEW."name");
    NEW."surname" = LTRIM(NEW."surname");
    NEW."surname" = UPPER(NEW."surname");
    IF NEW."salary" < 2800 THEN
```

```
            RAISE EXCEPTION 'Salary should be greater
than Minimum wage - 2800৳';
    END IF;
    RETURN NEW;
END;
$$;


ALTER FUNCTION public."addDriverTR1"() OWNER TO
postgres;

--
-- Name: bringavailablebuses(); Type: FUNCTION; Schema:
public; Owner: postgres
--

CREATE FUNCTION public.bringavailablebuses() RETURNS
text
    LANGUAGE plpgsql
    AS $$
DECLARE
    buses "Bus"%ROWTYPE;
   busPlateNo Text;
BEGIN
    busPlateNo := '';
    FOR buses IN SELECT * FROM "Bus"
    LEFT JOIN "Driver" ON "Bus"."plateNo" =
"Driver"."plateNo"
    WHERE "driverID" IS NULL LOOP
        busPlateNo := busPlateNo || buses."plateNo" ||
E'\r\n';
    END LOOP;
    RETURN busPlateNo;
```

```
END;
$$;


ALTER FUNCTION public.bringavailablebuses() OWNER TO
postgres;

--
-- Name: calculateprice(integer, date); Type: FUNCTION;
Schema: public; Owner: postgres
--

CREATE FUNCTION public.calculateprice(baseprice
integer, birthdate date) RETURNS integer
    LANGUAGE plpgsql
    AS $$
DECLARE
    age INT;
    finalPrice INT;
BEGIN
    age := EXTRACT(year FROM
age(current_date,birthdate));
    finalPrice := baseprice;

    IF getDiscountID(birthdate) = 1 THEN
        finalPrice := baseprice * 0.7;
    END IF;
    IF getDiscountID(birthdate) = 2 THEN
        finalPrice := basePrice * 0.8;
    END IF;
    RETURN finalPrice;
END;
$$;
```

```
ALTER FUNCTION public.calculateprice(baseprice integer,
birthdate date) OWNER TO postgres;

--
-- Name: controlStatus(); Type: FUNCTION; Schema:
public; Owner: postgres
--

CREATE FUNCTION public."controlStatus"() RETURNS
trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    IF OLD."status" != 'Unbooked' THEN
            RAISE EXCEPTION 'Trip can not be deleted
since it is already booked';
    ELSE
        UPDATE "Driver"
        SET "status" = 'Available'
        WHERE "Driver"."driverID" = OLD."driver";

        UPDATE "Bus"
        SET "status" = 'Available'
        WHERE "Bus"."plateNo" = OLD."bus";
    END IF;
     RETURN OLD;
END;
$$;
```

```
ALTER FUNCTION public."controlStatus"() OWNER TO
postgres;

--
-- Name: getdiscountid(date); Type: FUNCTION; Schema:
public; Owner: postgres
--

CREATE FUNCTION public.getdiscountid(birthdate date)
RETURNS integer
    LANGUAGE plpgsql
    AS $$
DECLARE
    discountID INT;
    age INT;
BEGIN
    age := EXTRACT(year FROM
age(current_date,birthdate));
    discountID := 5;

    IF age < 24  THEN
        discountID := 1;
    END IF;
    IF age > 60 THEN
        discountID := 2;
    END IF;
    RETURN discountID;
END;
$$;


ALTER FUNCTION public.getdiscountid(birthdate date)
OWNER TO postgres;
```

```
--
-- Name: salaryChangesTR(); Type: FUNCTION; Schema:
public; Owner: postgres
--

CREATE FUNCTION public."salaryChangesTR"() RETURNS
trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    IF NEW."salary" <> OLD."salary" THEN
        INSERT INTO "SalaryChanges"("driverID",
"oldSalary", "newSalary", "updatedOn")
        VALUES(OLD."driverID", OLD."salary",
NEW."salary", CURRENT_TIMESTAMP::TIMESTAMP);
    END IF;

    RETURN NEW;
END;
$$;


ALTER FUNCTION public."salaryChangesTR"() OWNER TO
postgres;

--
-- Name: searchdistricts(integer); Type: FUNCTION;
Schema: public; Owner: postgres
--
```

```
CREATE FUNCTION public.searchdistricts(district
integer) RETURNS TABLE("districtID" integer, "Name"
character varying)
    LANGUAGE plpgsql
    AS $$
BEGIN
    RETURN QUERY SELECT "districtNo", "name" FROM
"District"
                WHERE "districtNo" = district;
END;
$$;


ALTER FUNCTION public.searchdistricts(district integer)
OWNER TO postgres;

--
-- Name: setStatus(); Type: FUNCTION; Schema: public;
Owner: postgres
--

CREATE FUNCTION public."setStatus"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    UPDATE "Driver"
    SET "status" = 'Busy'
    WHERE "Driver"."driverID" = NEW."driver";

    UPDATE "Bus"
    SET "status" = 'Busy'
    WHERE "Bus"."plateNo" = NEW."bus";
```

```
    RETURN NEW;
END;
$$;


ALTER FUNCTION public."setStatus"() OWNER TO postgres;

--
-- Name: updateTrip(); Type: FUNCTION; Schema: public;
Owner: postgres
--

CREATE FUNCTION public."updateTrip"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    UPDATE "Trip"
    SET "status" = 'Booked'
    WHERE "Trip"."tripID" = NEW."tripID";
           RETURN NEW;

END;
$$;


ALTER FUNCTION public."updateTrip"() OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
```

```
-- Name: Bus; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Bus" (
    "plateNo" integer NOT NULL,
    "brandName" character varying(40) NOT NULL,
    model character varying(40) NOT NULL,
    "numberOfSeats" smallint DEFAULT 20,
    status character varying(10) DEFAULT
'Available'::character varying,
    company integer DEFAULT 1
);


ALTER TABLE public."Bus" OWNER TO postgres;

--
-- Name: Bus_plateNo_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Bus_plateNo_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Bus_plateNo_seq" OWNER TO postgres;
```

```
--
-- Name: Bus_plateNo_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Bus_plateNo_seq" OWNED BY
public."Bus"."plateNo";


--
-- Name: City; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."City" (
    "cityID" integer NOT NULL,
    name character varying(20) NOT NULL,
    "districtNo" integer NOT NULL
);


ALTER TABLE public."City" OWNER TO postgres;

--
-- Name: City_cityID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."City_cityID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
```

```
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."City_cityID_seq" OWNER TO postgres;

--
-- Name: City_cityID_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE public."City_cityID_seq" OWNED BY
public."City"."cityID";


--
-- Name: Company; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Company" (
    "companyID" integer NOT NULL,
    name character varying(40) NOT NULL,
    "districtNo" integer NOT NULL
);


ALTER TABLE public."Company" OWNER TO postgres;

--
-- Name: Company_companyID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--
```

```
CREATE SEQUENCE public."Company_companyID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Company_companyID_seq" OWNER TO
postgres;

--
-- Name: Company_companyID_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Company_companyID_seq" OWNED BY
public."Company"."companyID";


--
-- Name: ContactInformation; Type: TABLE; Schema:
public; Owner: postgres
--

CREATE TABLE public."ContactInformation" (
    "contactID" integer NOT NULL,
    "phoneNo" character varying(20) NOT NULL,
    email character varying(30),
    "districtNo" integer,
    company integer
);
```

```
ALTER TABLE public."ContactInformation" OWNER TO
postgres;

--
-- Name: ContactInformation_contactID_seq; Type:
SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE
public."ContactInformation_contactID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."ContactInformation_contactID_seq"
OWNER TO postgres;

--
-- Name: ContactInformation_contactID_seq; Type:
SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE
public."ContactInformation_contactID_seq" OWNED BY
public."ContactInformation"."contactID";


--
```

```
-- Name: Discount; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Discount" (
    "discountID" integer NOT NULL,
    name character varying(30) NOT NULL,
    "discountRate" real NOT NULL
);


ALTER TABLE public."Discount" OWNER TO postgres;

--
-- Name: Discount_discountID_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Discount_discountID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Discount_discountID_seq" OWNER TO
postgres;

--
-- Name: Discount_discountID_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
```

```
--

ALTER SEQUENCE public."Discount_discountID_seq" OWNED
BY public."Discount"."discountID";


--
-- Name: District; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."District" (
    "districtNo" integer NOT NULL,
    name character varying(20) NOT NULL
);


ALTER TABLE public."District" OWNER TO postgres;

--
-- Name: District_districtNo_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE public."District_districtNo_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
```

```
ALTER TABLE public."District_districtNo_seq" OWNER TO
postgres;

--
-- Name: District_districtNo_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."District_districtNo_seq" OWNED
BY public."District"."districtNo";


--
-- Name: Driver; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Driver" (
    "driverID" integer NOT NULL,
    name character varying(30) NOT NULL,
    surname character varying(30) NOT NULL,
    salary integer NOT NULL,
    status character varying(10) DEFAULT
'Available'::character varying,
    "plateNo" integer NOT NULL,
    contact integer,
    company integer NOT NULL
);


ALTER TABLE public."Driver" OWNER TO postgres;

--
```

```
-- Name: Driver_driverID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Driver_driverID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Driver_driverID_seq" OWNER TO
postgres;

--
-- Name: Driver_driverID_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Driver_driverID_seq" OWNED BY
public."Driver"."driverID";


--
-- Name: Passenger; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Passenger" (
    "passengerID" integer NOT NULL,
    name character varying(30) NOT NULL,
```

```
    surname character varying(30) NOT NULL,
    "birthDate" date NOT NULL,
    contact integer NOT NULL,
    "userID" integer
);


ALTER TABLE public."Passenger" OWNER TO postgres;

--
-- Name: Passenger_passengerID_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Passenger_passengerID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Passenger_passengerID_seq" OWNER TO
postgres;

--
-- Name: Passenger_passengerID_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Passenger_passengerID_seq" OWNED
BY public."Passenger"."passengerID";
```

```
--
-- Name: Payment; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Payment" (
    "paymentID" integer NOT NULL,
    amount character varying(30) NOT NULL,
    passenger integer NOT NULL,
    reservation integer NOT NULL
);


ALTER TABLE public."Payment" OWNER TO postgres;

--
-- Name: Payment_paymentID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Payment_paymentID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Payment_paymentID_seq" OWNER TO
postgres;
```

```
--
-- Name: Payment_paymentID_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Payment_paymentID_seq" OWNED BY
public."Payment"."paymentID";


--
-- Name: Reservation; Type: TABLE; Schema: public;
Owner: postgres
--

CREATE TABLE public."Reservation" (
    "reservationID" integer NOT NULL,
    "journeyDate" date NOT NULL,
    "seatNo" smallint NOT NULL,
    "passengerID" integer NOT NULL,
    "tripID" integer NOT NULL,
    "ticketID" integer NOT NULL,
    discount integer
);


ALTER TABLE public."Reservation" OWNER TO postgres;

--
-- Name: Reservation_reservationID_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Reservation_reservationID_seq"
```

```
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Reservation_reservationID_seq"
OWNER TO postgres;

--
-- Name: Reservation_reservationID_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Reservation_reservationID_seq"
OWNED BY public."Reservation"."reservationID";


--
-- Name: Route; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Route" (
    "routeID" integer NOT NULL,
    departure character varying(30) NOT NULL,
    destination character varying(30) NOT NULL,
    distance character varying(20),
    station integer
);
```

```
ALTER TABLE public."Route" OWNER TO postgres;

--
-- Name: Route_routeID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Route_routeID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Route_routeID_seq" OWNER TO
postgres;

--
-- Name: Route_routeID_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Route_routeID_seq" OWNED BY
public."Route"."routeID";


--
-- Name: SalaryChanges; Type: TABLE; Schema: public;
Owner: postgres
--
```

```
CREATE TABLE public."SalaryChanges" (
    "recordNo" integer NOT NULL,
    "driverID" smallint NOT NULL,
    "oldSalary" real NOT NULL,
    "newSalary" real NOT NULL,
    "updatedOn" timestamp without time zone NOT NULL
);


ALTER TABLE public."SalaryChanges" OWNER TO postgres;

--
-- Name: SalaryChanges_recordNo_seq; Type: SEQUENCE;
Schema: public; Owner: postgres
--

CREATE SEQUENCE public."SalaryChanges_recordNo_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."SalaryChanges_recordNo_seq" OWNER
TO postgres;

--
-- Name: SalaryChanges_recordNo_seq; Type: SEQUENCE
OWNED BY; Schema: public; Owner: postgres
--
```

```
ALTER SEQUENCE public."SalaryChanges_recordNo_seq"
OWNED BY public."SalaryChanges"."recordNo";


--
-- Name: Station; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Station" (
    "stationID" integer NOT NULL,
    name character varying(30) NOT NULL,
    location character varying(30) NOT NULL
);


ALTER TABLE public."Station" OWNER TO postgres;

--
-- Name: Station_stationID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Station_stationID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Station_stationID_seq" OWNER TO
postgres;
```

```
--
-- Name: Station_stationID_seq; Type: SEQUENCE OWNED
BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Station_stationID_seq" OWNED BY
public."Station"."stationID";


--
-- Name: Ticket; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Ticket" (
    "ticketID" integer NOT NULL,
    price money NOT NULL,
    trip integer
);


ALTER TABLE public."Ticket" OWNER TO postgres;

--
-- Name: Ticket_ticketID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Ticket_ticketID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
```

```
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Ticket_ticketID_seq" OWNER TO
postgres;

--
-- Name: Ticket_ticketID_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Ticket_ticketID_seq" OWNED BY
public."Ticket"."ticketID";


--
-- Name: Trip; Type: TABLE; Schema: public; Owner:
postgres
--

CREATE TABLE public."Trip" (
    "tripID" integer NOT NULL,
    "departureTime" time without time zone NOT NULL,
    "arrivalTime" time without time zone NOT NULL,
    "travelDuration" character varying(30),
    status character varying(10) DEFAULT
'Unbooked'::character varying,
    bus integer NOT NULL,
    driver integer NOT NULL,
    route integer NOT NULL
);
```

```
ALTER TABLE public."Trip" OWNER TO postgres;

--
-- Name: Trip_tripID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Trip_tripID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Trip_tripID_seq" OWNER TO postgres;

--
-- Name: Trip_tripID_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Trip_tripID_seq" OWNED BY
public."Trip"."tripID";


--
-- Name: Users; Type: TABLE; Schema: public; Owner:
postgres
--
```

```
CREATE TABLE public."Users" (
    "userID" integer NOT NULL,
    username character varying(20) NOT NULL,
    password character varying(10) NOT NULL,
    email character varying(30) NOT NULL,
    type character(1) DEFAULT 'P'::bpchar,
    status character varying(10) DEFAULT
'Passive'::character varying
);


ALTER TABLE public."Users" OWNER TO postgres;

--
-- Name: Users_userID_seq; Type: SEQUENCE; Schema:
public; Owner: postgres
--

CREATE SEQUENCE public."Users_userID_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Users_userID_seq" OWNER TO
postgres;

--
-- Name: Users_userID_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres
```

```
--

ALTER SEQUENCE public."Users_userID_seq" OWNED BY
public."Users"."userID";


--
-- Name: Bus plateNo; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Bus" ALTER COLUMN "plateNo"
SET DEFAULT
nextval('public."Bus_plateNo_seq"'::regclass);


--
-- Name: City cityID; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."City" ALTER COLUMN "cityID"
SET DEFAULT
nextval('public."City_cityID_seq"'::regclass);


--
-- Name: Company companyID; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Company" ALTER COLUMN
"companyID" SET DEFAULT
nextval('public."Company_companyID_seq"'::regclass);
```

```
--
-- Name: ContactInformation contactID; Type: DEFAULT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."ContactInformation" ALTER
COLUMN "contactID" SET DEFAULT
nextval('public."ContactInformation_contactID_seq"'::re
gclass);


--
-- Name: Discount discountID; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Discount" ALTER COLUMN
"discountID" SET DEFAULT
nextval('public."Discount_discountID_seq"'::regclass);


--
-- Name: District districtNo; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."District" ALTER COLUMN
"districtNo" SET DEFAULT
nextval('public."District_districtNo_seq"'::regclass);


--
```

```
-- Name: Driver driverID; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Driver" ALTER COLUMN
"driverID" SET DEFAULT
nextval('public."Driver_driverID_seq"'::regclass);


--
-- Name: Passenger passengerID; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Passenger" ALTER COLUMN
"passengerID" SET DEFAULT
nextval('public."Passenger_passengerID_seq"'::regclass)
;


--
-- Name: Payment paymentID; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Payment" ALTER COLUMN
"paymentID" SET DEFAULT
nextval('public."Payment_paymentID_seq"'::regclass);


--
-- Name: Reservation reservationID; Type: DEFAULT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Reservation" ALTER COLUMN
"reservationID" SET DEFAULT
nextval('public."Reservation_reservationID_seq"'::regcl
ass);


--
-- Name: Route routeID; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Route" ALTER COLUMN "routeID"
SET DEFAULT
nextval('public."Route_routeID_seq"'::regclass);


--
-- Name: SalaryChanges recordNo; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."SalaryChanges" ALTER COLUMN
"recordNo" SET DEFAULT
nextval('public."SalaryChanges_recordNo_seq"'::regclass
);


--
-- Name: Station stationID; Type: DEFAULT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Station" ALTER COLUMN
"stationID" SET DEFAULT
nextval('public."Station_stationID_seq"'::regclass);


--
-- Name: Ticket ticketID; Type: DEFAULT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Ticket" ALTER COLUMN
"ticketID" SET DEFAULT
nextval('public."Ticket_ticketID_seq"'::regclass);


--
-- Name: Trip tripID; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Trip" ALTER COLUMN "tripID"
SET DEFAULT
nextval('public."Trip_tripID_seq"'::regclass);


--
-- Name: Users userID; Type: DEFAULT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Users" ALTER COLUMN "userID"
SET DEFAULT
nextval('public."Users_userID_seq"'::regclass);
```

```
--
-- Data for Name: Bus; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Bus" VALUES
    (789456, 'Volks Wagen', 'VW34', 16, 'Available',
1),
    (1111, '2134', '4124', 20, 'Busy', 1),
    (12, '134', '124', 20, 'Busy', 1),
    (124241124, '12', '12', 20, 'Busy', 1),
    (333333, 'Eurotracker', 'ER345', 14, 'Busy', 1),
    (895623, 'Isuzu', 'IS3234', 20, 'Busy', 1),
    (21, '214', 'rrr241', 20, 'Busy', 1),
    (123456, 'Volvo', 'V3ER', 20, 'Available', 1),
    (789789, 'test', 'ER32', 10, 'Available', 1),
    (456789, 'Mercedes', 'MW49', 18, 'Available', 1),
    (784512, 'Toyota', 'TER234', 20, 'Available', 1),
    (12444, '124', '123412', 20, 'Available', 1),
    (444, '234', '2352', 20, 'Available', 1),
    (2312111, 'eewew', 'qqq', 20, 'Available', 1),
    (24, 'dddd', '123', 20, 'Available', 1);


--
-- Data for Name: City; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."City" VALUES
    (1, 'Sakarya', 1),
    (2, 'Sakarya', 7),
    (3, 'Sakarya', 8),
```

```
    (4, 'Sakarya', 9),
    (5, 'Istanbul', 2),
    (6, 'Istanbul', 3),
    (7, 'Bursa', 10),
    (8, 'Bursa', 11),
    (9, 'Istanbul', 4),
    (10, 'Bursa', 12),
    (11, 'Istanbul', 5),
    (12, 'Bursa', 13),
    (13, 'Istanbul', 6),
    (14, 'Izmir', 17),
    (15, 'Izmir', 16),
    (16, 'Ankara', 20),
    (17, 'Ankara', 19),
    (18, 'Izmir', 14),
    (19, 'Ankara', 18),
    (21, 'Izmir', 15),
    (22, 'Ankara', 21);


--
-- Data for Name: Company; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Company" VALUES
    (1, 'Epix Travel Ltd', 1);


--
-- Data for Name: ContactInformation; Type: TABLE DATA;
Schema: public; Owner: postgres
--
```

```
INSERT INTO public."ContactInformation" VALUES
    (1, '5539501268', 'testting@epix.com', 1, 1),
    (9, '8494916986', 'driver@epixtravels.com', 1, 1),
    (10, '475788878', NULL, 1, 1),
    (11, '1241312413', 'wow@epixtravels.com', 1, 1),
    (12, '05547894512', 'New@epixtravels.com', 2, 1),
    (13, '055784512', 'Testing@epixtravels.com', 4, 1),
    (16, '05578451212', 'Testinga@epixtravels.com', 4,
1),
    (17, '4556422246', 'sofor@epixtravels.com', 20, 1),
    (18, '111111111', 'Check@epixtravels.com', 11, 1),
    (19, '222222222222222', 'Checking@epixtravels.com',
17, 1),
    (29, '054687121584', 'NewPassenger@gmail.com@ ',
NULL, NULL),
    (30, '05784512281', 'Izmir@gmail.com@ ', NULL,
NULL),
    (31, '78577852778', 'dw@hotmail.com@ ', NULL,
NULL),
    (32, '78513848', 'cool@yahoo.com@ ', NULL, NULL),
    (33, '12412', 're@gmail.com@ ', NULL, NULL),
    (34, '984518', 'fe@gmail.com@ ', NULL, NULL),
    (35, '1241241', 'check@hotmalci.com@ ', NULL,
NULL),
    (36, '12413412', 'jamy@gmail.com@ ', NULL, NULL),
    (37, '214241341', 'ozil@gmail.com@ ', NULL, NULL),
    (38, '1241242', 'resul@gmaul.com@ ', NULL, NULL),
    (39, '21412', 'deejoong@gamil.com@ ', NULL, NULL),
    (41, '2141141231423', 'Jota@gmail.com@ ', NULL,
NULL),
    (42, '12421412412', 'alonso@gamilk..com@ ', NULL,
NULL),
```

```
    (43, '0548131984 35', 'Elon@genius.com@ ', NULL,
NULL),
    (44, '0654238329', 'burak@gmail.com@ ', NULL,
NULL),
    (45, '0548213218', 'Umer@epixtravels.com', 17, 1),
    (46, '06518158165', 'Mohammed@epixtravels.com', 19,
1);


--
-- Data for Name: Discount; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Discount" VALUES
    (2, 'Above 60', 0.2),
    (1, 'Student Under 24', 0.3),
    (5, 'Normal', 0);


--
-- Data for Name: District; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."District" VALUES
    (1, 'Serdivan'),
    (2, 'Üsküdar'),
    (3, 'Kadiköy'),
    (4, 'Fatih'),
    (5, 'Beşiktaş'),
    (6, 'Pendik'),
    (7, 'Arifiye'),
```

```
    (8, 'Adapazarı'),
    (9, 'Erenler'),
    (10, 'Osmangazi'),
    (11, 'Gürsu'),
    (12, 'Orhaneli '),
    (13, 'Karacabey'),
    (14, 'Bayraklı'),
    (15, 'Güzelbahçe'),
    (16, 'Bornova '),
    (17, 'Gaziemir'),
    (18, 'Çankaya'),
    (19, 'Beypazarı'),
    (20, 'Kahramankazan '),
    (21, 'Kalecik'),
    (22, 'Kadiköy'),
    (23, 'Kadiköy'),
    (24, 'Kadiköy'),
    (25, 'Kadiköy'),
    (26, 'Kadiköy'),
    (27, 'Kadiköy'),
    (28, 'Kadiköy'),
    (29, 'Kadiköy'),
    (30, 'Kadiköy'),
    (31, 'Üsküdar'),
    (32, 'Fatih'),
    (33, 'Fatih'),
    (34, 'Fatih'),
    (35, 'Fatih'),
    (36, 'Kahramankazan '),
    (37, 'Gürsu'),
    (38, 'Gürsu'),
    (39, 'Gaziemir'),
    (40, 'Gaziemir'),
```

```
    (41, 'Gaziemir'),
    (42, 'Gaziemir'),
    (43, 'Gaziemir'),
    (44, 'Beypazarı');


--
-- Data for Name: Driver; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Driver" VALUES
    (5, 'triggertest', 'SMALLLETTERS', 5000, 'Busy',
123456, 1, 1),
    (10, 'Check', 'AGAIN', 6000, 'Busy', 1111, NULL,
1),
    (12, 'Checking', 'AGAAAIN', 4500, 'Busy', 12, 19,
1),
    (3, 'SMALLLETTERS', 'SMALLLETTERS', 2500, 'Busy',
123456, 1, 1),
    (13, 'Checking', 'AGAAAIN', 4500, 'Busy', 12, NULL,
1),
    (8, 'sofor', 'YENI', 5000, 'Busy', 124241124, 17,
1),
    (7, 'Testinga', 'NEW', 5222, 'Busy', 333333, 16,
1),
    (15, 'Umer', 'FATIH', 3200, 'Busy', 895623, 45, 1),
    (16, 'Mohammed', 'BEYLUL', 5000, 'Busy', 21, 46,
1),
    (4, 'SMALLLETTERS', 'SMALLLETTERS', 4000,
'Available', 123456, 1, 1);
```

```
--
-- Data for Name: Passenger; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Passenger" VALUES
    (5, 'New', 'Passenger', '2000-03-16', 29, 1),
    (6, 'Izmire', 'Gidiyorum', '1960-12-31', 30, 1),
    (7, 'hi', 'hi', '2000-07-04', 31, 1),
    (8, 'young', 'Boy', '2000-07-12', 32, 1),
    (9, 'hi', 'juyf', '1990-12-31', 33, 1),
    (10, 'Henry', 'Theo', '1960-12-31', 34, 1),
    (11, 'theo', 'walcott', '2001-06-01', 35, 1),
    (12, 'jamy', 'vardy', '1960-12-31', 36, 1),
    (13, 'mesut', 'ozil', '1940-12-31', 37, 1),
    (14, 'resul', 'daspinar', '2000-07-04', 38, 1),
    (15, 'Frank', 'de Jong', '1990-01-31', 39, 1),
    (17, 'Joa', 'Jota', '1991-12-31', 41, 1),
    (18, 'alonso', 'dee', '2000-12-31', 42, 1),
    (19, 'Elon', 'Musk', '1975-03-12', 43, 1),
    (20, 'Burak', 'Koca', '2000-07-12', 44, 1);


--
-- Data for Name: Payment; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Payment" VALUES
    (5, '210', 5, 5),
    (6, '175', 6, 6),
    (7, '175', 7, 7),
    (8, '175', 8, 8),
```

```
    (9, '210', 9, 9),
    (11, '280', 11, 11),
    (13, '240', 13, 13),
    (14, '210', 14, 14),
    (15, '300', 15, 15),
    (17, '250', 17, 17),
    (18, '175', 18, 18),
    (19, '90', 19, 19),
    (20, '175', 20, 20);


--
-- Data for Name: Reservation; Type: TABLE DATA;
Schema: public; Owner: postgres
--

INSERT INTO public."Reservation" VALUES
    (5, '2021-12-31', 5, 5, 42, 8, 1),
    (6, '2021-12-30', 4, 6, 44, 10, 1),
    (7, '2021-12-18', 0, 7, 43, 9, 1),
    (8, '2021-12-13', 0, 8, 44, 10, 1),
    (9, '2021-12-13', 5, 9, 42, 8, 1),
    (11, '2021-12-13', 7, 11, 41, 7, 1),
    (13, '2021-12-13', 3, 13, 42, 8, 2),
    (14, '2021-12-13', 7, 14, 42, 8, 1),
    (15, '2021-12-13', 9, 15, 42, 8, 5),
    (17, '2021-12-13', 1, 17, 44, 10, 5),
    (18, '2021-12-13', 5, 18, 44, 10, 1),
    (19, '2021-12-13', 0, 19, 46, 12, 5),
    (20, '2021-12-13', 8, 20, 44, 10, 1);


--
```

```
-- Data for Name: Route; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Route" VALUES
    (6, 'Ankara', 'Istanbul', '380', 17),
    (7, 'Bursa', 'Istanbul', '250Km', 17),
    (8, 'Bursa', 'Izmir', '350Km', 21),
    (9, 'Sakarya', 'Izmir', '340Km', 21),
    (10, 'Sakarya', 'Istanbul', '120Km', 17),
    (11, 'Ankara', 'Sakarya', '450Km', 22),
    (12, 'Ankara', 'Izmir', '450Km', 21),
    (13, 'Izmir', 'Istanbul', '450Km', 17),
    (14, 'Sakarya', 'Ankara', '340Km', 16),
    (15, 'Bursa', 'Sakarya', '400Km', 22);


--
-- Data for Name: SalaryChanges; Type: TABLE DATA;
Schema: public; Owner: postgres
--

INSERT INTO public."SalaryChanges" VALUES
    (1, 4, 3000, 4000, '2021-12-09 17:24:23.13356'),
    (2, 8, 7800, 6000, '2021-12-10 18:50:33.531276'),
    (3, 5, 3000, 5000, '2021-12-10 18:52:25.737664'),
    (4, 8, 6000, 5000, '2021-12-10 19:02:48.920452'),
    (5, 7, 3500, 5222, '2021-12-10 20:20:47.700011'),
    (6, 3, 3000, 2500, '2021-12-11 13:27:09.41792');


--
```

```
-- Data for Name: Station; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Station" VALUES
    (16, 'Ankara Terminal', 'Ankara'),
    (17, 'Harem', 'Istanbul'),
    (18, 'Bursa Terminal`', 'Bursa'),
    (19, 'Esenler', 'Istanbul'),
    (20, 'Bursa Terminal`', 'Bursa'),
    (21, 'Izmir Terminal', 'Izmir'),
    (22, 'Sakarya Terminal', 'Sakarya'),
    (23, 'Izmir Terminal', 'Izmir'),
    (24, 'Sakarya Terminal', 'Sakarya'),
    (25, 'Esenler', 'Istanbul'),
    (26, 'Ankara Terminal', 'Ankara'),
    (27, 'Sakarya Terminal', 'Sakarya'),
    (28, 'Ankara Terminal', 'Ankara'),
    (29, 'Izmir Terminal', 'Izmir'),
    (30, 'Izmir Terminal', 'Izmir'),
    (31, 'Harem', 'Istanbul'),
    (32, 'sakarya Terminal', 'Sakarya'),
    (33, 'Ankara Terminal', 'Ankara'),
    (34, 'Bursa Terminal', 'Bursa'),
    (35, 'Sakarya Terminal', 'Sakarya');


--
-- Data for Name: Ticket; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Ticket" VALUES
```

```
    (1, '$75.00', NULL),
    (2, '$120.00', NULL),
    (3, '$55.00', NULL),
    (4, '$120.00', NULL),
    (5, '$120.00', NULL),
    (6, '$450.00', NULL),
    (10, '$250.00', 44),
    (7, '$125.00', 41),
    (8, '$75.00', 42),
    (9, '$200.00', 43),
    (12, '$90.00', 46),
    (14, '$80.00', 48),
    (15, '$90.00', 49);


--
-- Data for Name: Trip; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Trip" VALUES
    (40, '02:54:00', '04:00:00', '01:06:00',
'Unbooked', 1111, 10, 9),
    (41, '03:00:00', '05:00:00', '02:00:00',
'Unbooked', 12, 12, 10),
    (42, '06:00:00', '08:00:00', '02:00:00',
'Unbooked', 123456, 3, 10),
    (43, '07:00:00', '09:00:00', '02:00:00',
'Unbooked', 12, 13, 10),
    (46, '09:00:00', '12:30:00', '03:30:00', 'Booked',
333333, 7, 13),
    (44, '09:06:00', '11:00:00', '01:54:00', 'Booked',
124241124, 8, 8),
```

```
    (47, '08:30:00', '12:00:00', '03:30:00',
'Unbooked', 123456, 4, 15),
    (48, '10:55:00', '10:55:00', '00:00:00',
'Unbooked', 895623, 15, 15),
    (49, '10:55:00', '10:55:00', '00:00:00',
'Unbooked', 21, 16, 15);


--
-- Data for Name: Users; Type: TABLE DATA; Schema:
public; Owner: postgres
--

INSERT INTO public."Users" VALUES
    (2, 'admin', '456', 'admin@epix.com', 'A',
'Passive'),
    (3, '123', 'signup', 'signup@gmail.com', 'P',
'Passive'),
    (4, '123', 'signup', 'signup@gmail.com', 'P',
'Passive'),
    (5, '123', 'signup', 'signup@gmail.com', 'P',
'Passive'),
    (6, 'check', '1234', 'check@gmail.com', 'P',
'Passive'),
    (7, 'deneme', '963', 'deneme@gmail.com', 'P',
'Passive'),
    (1, 'test', '123', 'test@gmail.com', 'P',
'Passive');


--
-- Name: Bus_plateNo_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
```

```
--

SELECT pg_catalog.setval('public."Bus_plateNo_seq"', 1,
false);


--
-- Name: City_cityID_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."City_cityID_seq"',
22, true);


--
-- Name: Company_companyID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Company_companyID_seq"', 1,
true);


--
-- Name: ContactInformation_contactID_seq; Type:
SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."ContactInformation_contactID
_seq"', 46, true);
```

```
--
-- Name: Discount_discountID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Discount_discountID_seq"',
5, true);


--
-- Name: District_districtNo_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."District_districtNo_seq"',
44, true);


--
-- Name: Driver_driverID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Driver_driverID_seq"', 16,
true);


--
-- Name: Passenger_passengerID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--
```

```
SELECT
pg_catalog.setval('public."Passenger_passengerID_seq"',
20, true);


--
-- Name: Payment_paymentID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Payment_paymentID_seq"', 20,
true);


--
-- Name: Reservation_reservationID_seq; Type: SEQUENCE
SET; Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Reservation_reservationID_se
q"', 20, true);


--
-- Name: Route_routeID_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Route_routeID_seq"',
15, true);
```

```
--
-- Name: SalaryChanges_recordNo_seq; Type: SEQUENCE
SET; Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."SalaryChanges_recordNo_seq"'
, 6, true);


--
-- Name: Station_stationID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Station_stationID_seq"', 35,
true);


--
-- Name: Ticket_ticketID_seq; Type: SEQUENCE SET;
Schema: public; Owner: postgres
--

SELECT
pg_catalog.setval('public."Ticket_ticketID_seq"', 15,
true);


--
-- Name: Trip_tripID_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('public."Trip_tripID_seq"',
49, true);


--
-- Name: Users_userID_seq; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."Users_userID_seq"',
7, true);


--
-- Name: Driver DriverPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Driver"
    ADD CONSTRAINT "DriverPK" PRIMARY KEY ("driverID");


--
-- Name: SalaryChanges PK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."SalaryChanges"
    ADD CONSTRAINT "PK" PRIMARY KEY ("recordNo");


--
```

```
-- Name: Bus busPK; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Bus"
    ADD CONSTRAINT "busPK" PRIMARY KEY ("plateNo");


--
-- Name: City cityPK; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."City"
    ADD CONSTRAINT "cityPK" PRIMARY KEY ("cityID");


--
-- Name: Company companyPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Company"
    ADD CONSTRAINT "companyPK" PRIMARY KEY
("companyID");


--
-- Name: ContactInformation contactInformationPK; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."ContactInformation"
```

```
    ADD CONSTRAINT "contactInformationPK" PRIMARY KEY
("contactID");


--
-- Name: Discount discountPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Discount"
    ADD CONSTRAINT "discountPK" PRIMARY KEY
("discountID");


--
-- Name: District districtPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."District"
    ADD CONSTRAINT "districtPK" PRIMARY KEY
("districtNo");


--
-- Name: Passenger passengerPK; Type: CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Passenger"
    ADD CONSTRAINT "passengerPK" PRIMARY KEY
("passengerID");
```

```
--
-- Name: Payment paymentPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Payment"
    ADD CONSTRAINT "paymentPK" PRIMARY KEY
("paymentID");


--
-- Name: Reservation reservationPK; Type: CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Reservation"
    ADD CONSTRAINT "reservationPK" PRIMARY KEY
("reservationID");


--
-- Name: Route routePK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Route"
    ADD CONSTRAINT "routePK" PRIMARY KEY ("routeID");


--
-- Name: Station stationPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Station"
    ADD CONSTRAINT "stationPK" PRIMARY KEY
("stationID");


--
-- Name: Ticket ticketPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Ticket"
    ADD CONSTRAINT "ticketPK" PRIMARY KEY ("ticketID");


--
-- Name: Trip tripPK; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."Trip"
    ADD CONSTRAINT "tripPK" PRIMARY KEY ("tripID");


--
-- Name: Users userPK; Type: CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Users"
    ADD CONSTRAINT "userPK" PRIMARY KEY ("userID");


--
```

```
-- Name: ContactInformation usersUnique1; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."ContactInformation"
    ADD CONSTRAINT "usersUnique1" UNIQUE (email);


--
-- Name: ContactInformation usersUnique2; Type:
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."ContactInformation"
    ADD CONSTRAINT "usersUnique2" UNIQUE ("phoneNo");


--
-- Name: Driver addingControl; Type: TRIGGER; Schema:
public; Owner: postgres
--

CREATE TRIGGER "addingControl" BEFORE INSERT OR UPDATE
ON public."Driver" FOR EACH ROW EXECUTE FUNCTION
public."addDriverTR1"();


--
-- Name: ContactInformation createEmail; Type: TRIGGER;
Schema: public; Owner: postgres
--
```

```
CREATE TRIGGER "createEmail" BEFORE INSERT ON
public."ContactInformation" FOR EACH ROW EXECUTE
FUNCTION public."addContactInformation"();


--
-- Name: Trip driverTR3; Type: TRIGGER; Schema: public;
Owner: postgres
--

CREATE TRIGGER "driverTR3" AFTER INSERT ON
public."Trip" FOR EACH ROW EXECUTE FUNCTION
public."setStatus"();


--
-- Name: Driver onSalaryUpdate; Type: TRIGGER; Schema:
public; Owner: postgres
--

CREATE TRIGGER "onSalaryUpdate" BEFORE UPDATE ON
public."Driver" FOR EACH ROW EXECUTE FUNCTION
public."salaryChangesTR"();


--
-- Name: Reservation reservationTR; Type: TRIGGER;
Schema: public; Owner: postgres
--

CREATE TRIGGER "reservationTR" AFTER INSERT ON
public."Reservation" FOR EACH ROW EXECUTE FUNCTION
public."updateTrip"();
```

```
--
-- Name: Trip tripTR1; Type: TRIGGER; Schema: public;
Owner: postgres
--

CREATE TRIGGER "tripTR1" BEFORE DELETE ON public."Trip"
FOR EACH ROW EXECUTE FUNCTION public."controlStatus"();


--
-- Name: Bus busFK; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Bus"
    ADD CONSTRAINT "busFK" FOREIGN KEY (company)
REFERENCES public."Company"("companyID") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: City cityFK; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."City"
    ADD CONSTRAINT "cityFK" FOREIGN KEY ("districtNo")
REFERENCES public."District"("districtNo");


--
-- Name: Company companyFK; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Company"
    ADD CONSTRAINT "companyFK" FOREIGN KEY
("districtNo") REFERENCES
public."District"("districtNo") ON UPDATE CASCADE ON
DELETE CASCADE;


--
-- Name: ContactInformation contactInformationFK1;
Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."ContactInformation"
    ADD CONSTRAINT "contactInformationFK1" FOREIGN KEY
("districtNo") REFERENCES
public."District"("districtNo");


--
-- Name: ContactInformation contactInformationFK2;
Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."ContactInformation"
    ADD CONSTRAINT "contactInformationFK2" FOREIGN KEY
(company) REFERENCES public."Company"("companyID");


--
-- Name: Driver driverFK1; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--
```

```
ALTER TABLE ONLY public."Driver"
    ADD CONSTRAINT "driverFK1" FOREIGN KEY ("plateNo")
REFERENCES public."Bus"("plateNo") ON UPDATE CASCADE ON
DELETE CASCADE;


--
-- Name: Driver driverFK2; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Driver"
    ADD CONSTRAINT "driverFK2" FOREIGN KEY (contact)
REFERENCES public."ContactInformation"("contactID") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Driver driverFK3; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Driver"
    ADD CONSTRAINT "driverFK3" FOREIGN KEY (company)
REFERENCES public."Company"("companyID") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Passenger passengerFK; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Passenger"
```

```
    ADD CONSTRAINT "passengerFK" FOREIGN KEY (contact)
REFERENCES public."ContactInformation"("contactID") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Payment paymentFK1; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Payment"
    ADD CONSTRAINT "paymentFK1" FOREIGN KEY (passenger)
REFERENCES public."Passenger"("passengerID") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Payment paymentFK2; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Payment"
    ADD CONSTRAINT "paymentFK2" FOREIGN KEY
(reservation) REFERENCES
public."Reservation"("reservationID") ON UPDATE CASCADE
ON DELETE CASCADE;


--
-- Name: Reservation reservationFK1; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Reservation"
```

```
    ADD CONSTRAINT "reservationFK1" FOREIGN KEY
("passengerID") REFERENCES
public."Passenger"("passengerID") ON UPDATE CASCADE ON
DELETE CASCADE;


--
-- Name: Reservation reservationFK2; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Reservation"
    ADD CONSTRAINT "reservationFK2" FOREIGN KEY
("tripID") REFERENCES public."Trip"("tripID") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Reservation reservationFK3; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Reservation"
    ADD CONSTRAINT "reservationFK3" FOREIGN KEY
("ticketID") REFERENCES public."Ticket"("ticketID") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Reservation reservationFK4; Type: FK
CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Reservation"
```

```
    ADD CONSTRAINT "reservationFK4" FOREIGN KEY
(discount) REFERENCES public."Discount"("discountID")
ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Route routeFK; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Route"
    ADD CONSTRAINT "routeFK" FOREIGN KEY (station)
REFERENCES public."Station"("stationID") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Ticket ticketFK1; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Ticket"
    ADD CONSTRAINT "ticketFK1" FOREIGN KEY (trip)
REFERENCES public."Trip"("tripID") ON UPDATE CASCADE ON
DELETE CASCADE;


--
-- Name: Trip tripFK; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Trip"
```

```
    ADD CONSTRAINT "tripFK" FOREIGN KEY (bus)
REFERENCES public."Bus"("plateNo") ON UPDATE CASCADE ON
DELETE CASCADE;


--
-- Name: Trip tripFK2; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Trip"
    ADD CONSTRAINT "tripFK2" FOREIGN KEY (driver)
REFERENCES public."Driver"("driverID") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Trip tripFK3; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."Trip"
    ADD CONSTRAINT "tripFK3" FOREIGN KEY (route)
REFERENCES public."Route"("routeID") ON UPDATE CASCADE
ON DELETE CASCADE;


--
-- PostgreSQL database dump complete
--
```

Fonksiyonlar(4)

1. Şehirlerin sayısı bulan fonksiyon

```
CREATE OR REPLACE FUNCTION countCity(cityName TEXT, OUT counts INT)
AS
$$
BEGIN
  counts:= (SELECT COUNT(*) FROM "City"
        WHERE "name" = cityName);
END;
$$
LANGUAGE "plpgsql";
```

2. Müsait olan otobüsleri getiren fonksiyon

```
CREATE OR REPLACE FUNCTION bringAvailableBuses()
RETURNS TEXT
AS
$$
DECLARE
        buses "Bus"%ROWTYPE;
        busPlateNo Text;
BEGIN
        busPlateNo := '';
        FOR buses IN SELECT * FROM "Bus"
                LEFT JOIN "Driver" ON "Bus"."plateNo" = "Driver"."plateNo"
                WHERE "driverID" IS NULL LOOP
                busPlateNo := busPlateNo || buses."plateNo" || E'\r\n';
        END LOOP;
        RETURN busPlateNo;
END;
$$
```

```
LANGUAGE "plpgsql";
```

3. Girilen döğüm tarihinin İndirim kodun getiren fonksiyon

```
CREATE OR REPLACE FUNCTION getDiscountID(birthdate DATE)
RETURNS INT
AS
$$
DECLARE
        discountID INT;
        age INT;
BEGIN
        age := EXTRACT(year FROM age(current_date,birthdate));
        discountID := 5;

        IF age < 24  THEN
    discountID := 1;
        END IF;
        IF age > 60 THEN
                discountID := 2;
  END IF;
  RETURN discountID;
END;
$$
LANGUAGE 'plpgsql';
```

4. Yolcunun yaşına göre indirimi hesaplayıp son fiyatı getiren fonksiyon

```
CREATE OR REPLACE FUNCTION calculatePrice(basePrice INT, birthdate DATE)
RETURNS INT
AS
$$
DECLARE
        age INT;
        finalPrice INT;
BEGIN
        age := EXTRACT(year FROM age(current_date,birthdate));
        finalPrice := baseprice;

        IF getDiscountID(birthdate) = 1 THEN
                finalPrice := baseprice * 0.7;
        END IF;
        IF getDiscountID(birthdate) = 2 THEN
                finalPrice := basePrice * 0.8;
        END IF;
  RETURN finalPrice;
END;
$$
LANGUAGE 'plpgsql';
```

Trigger fonksiyonlar(6)

1. **Yeni şoför eklendiği zaman ad ve soyadı biçimlendirilir ve ayrıca maaşı askeri ücretten az olmaması sağlanır aksi takdirde hata fırlatılır.**

```
CREATE OR REPLACE FUNCTION "addDriverTR1"()
RETURNS TRIGGER
AS
$$
BEGIN
        NEW."name" = LTRIM(NEW."name");
        NEW."surname" = LTRIM(NEW."surname");
         NEW."surname" = UPPER(NEW."surname");
         IF NEW."salary" < 4250 THEN
                RAISE EXCEPTION 'Salary should be greater than minimum wage - 4250 ₺';
        END IF;
        RETURN NEW;
END;
$$
LANGUAGE "plpgsql";

CREATE TRIGGER "addingControl"
BEFORE INSERT OR UPDATE ON "Driver"
FOR EACH ROW
EXECUTE PROCEDURE "addDriverTR1"();
```

2. **Yeni şoför eklendiği zaman otobüs şirketinin uzantısıyla yeni eposta hesabı Şoför için oluşturulur.**

```
CREATE OR REPLACE FUNCTION "addContactInformation"()
RETURNS TRIGGER
AS
$$
BEGIN
  IF NEW."email" NOT LIKE '%@%' THEN
        NEW."email"       := CONCAT(NEW.email,
                          '@epixtravels.com');
  END IF;
  RETURN NEW;
END;
$$
LANGUAGE "plpgsql";
```

3. **Yeni bir sefer eklendiği zaman ilgili otobüs ve şoför meşgul edilir**

```
CREATE OR REPLACE FUNCTION "setStatus"()
RETURNS TRIGGER AS
$BODY$
BEGIN
  UPDATE "Driver"
  SET "status" = 'Busy'
  WHERE "Driver"."driverID" = NEW."driver";

   UPDATE "Bus"
   SET "status" = 'Busy'
  WHERE "Bus"."plateNo" = NEW."bus";
```

```
    RETURN NEW;
END;
$BODY$
language plpgsql;
CREATE TRIGGER "driverTR3"
AFTER INSERT ON "Trip"
FOR EACH ROW
EXECUTE PROCEDURE "setStatus"();,
```

4. Bir sefer rezerve edildiği zaman ilgili seferin satın alındığı ilan edilir. (silinmemesi adına)

```
CREATE OR REPLACE FUNCTION "updateTrip"()
RETURNS TRIGGER AS
$BODY$
BEGIN
  UPDATE "Trip"
  SET "status" = 'Booked'
  WHERE "Trip"."tripID" = NEW."tripID";
      RETURN NEW;

END;
$BODY$
language plpgsql;

CREATE TRIGGER "reservationTR"
AFTER INSERT ON "Reservation"
FOR EACH ROW
EXECUTE PROCEDURE "updateTrip"();
```

5. Bir sefer silinmeden önce o sefer satın alınıp alınmadığı kontrol edilir ve ayrıca ilgili şoför ve otobüs serbest bırakılır.

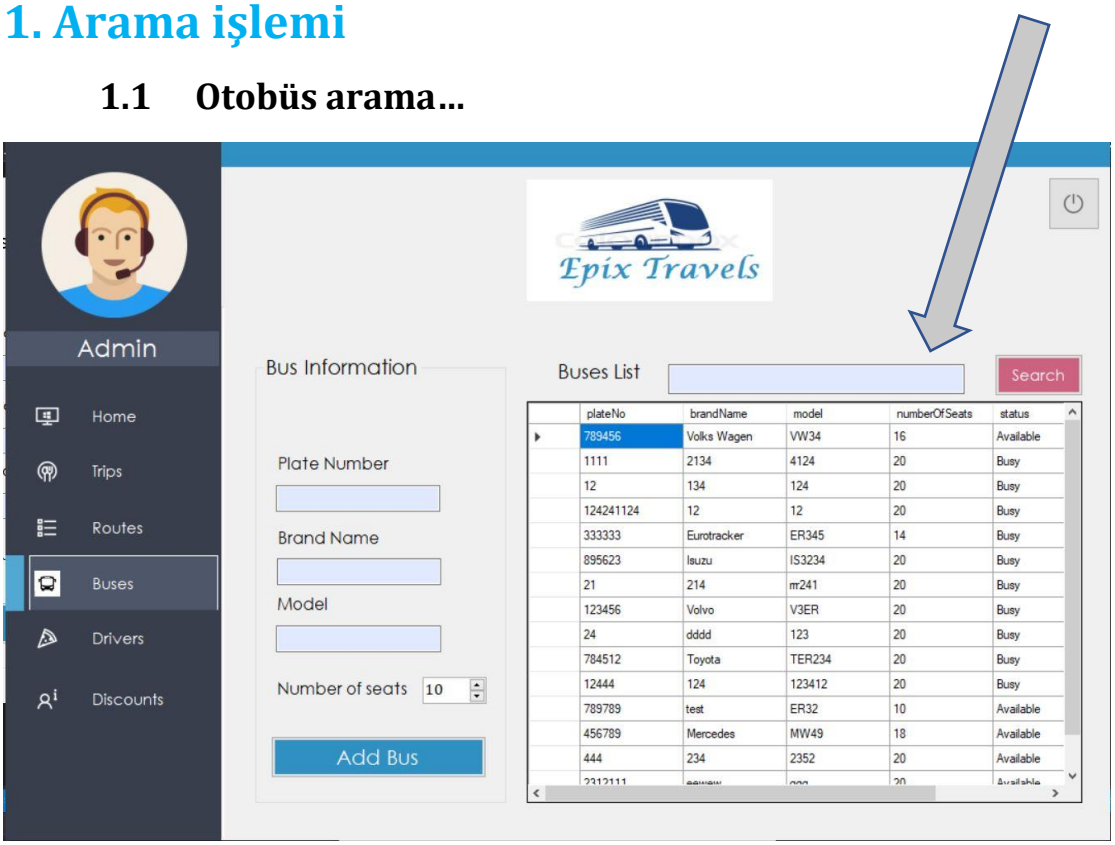
```
CREATE OR REPLACE FUNCTION "controlStatus"()
 RETURNS TRIGGER AS
$BODY$
BEGIN
        IF OLD."status" != 'Unbooked' THEN
                RAISE EXCEPTION 'Trip can not be deleted since it is already booked';
        ELSE
                UPDATE "Driver"
                SET "status" = 'Available'
                WHERE "Driver"."driverID" = OLD."driver";

                UPDATE "Bus"
                SET "status" = 'Available'
                WHERE "Bus"."plateNo" = OLD."bus";
        END IF;
         RETURN OLD;
END;
$BODY$
language plpgsql;
CREATE TRIGGER "tripTR1"
BEFORE DELETE ON "Trip"
FOR EACH ROW
EXECUTE PROCEDURE "controlStatus"();
```

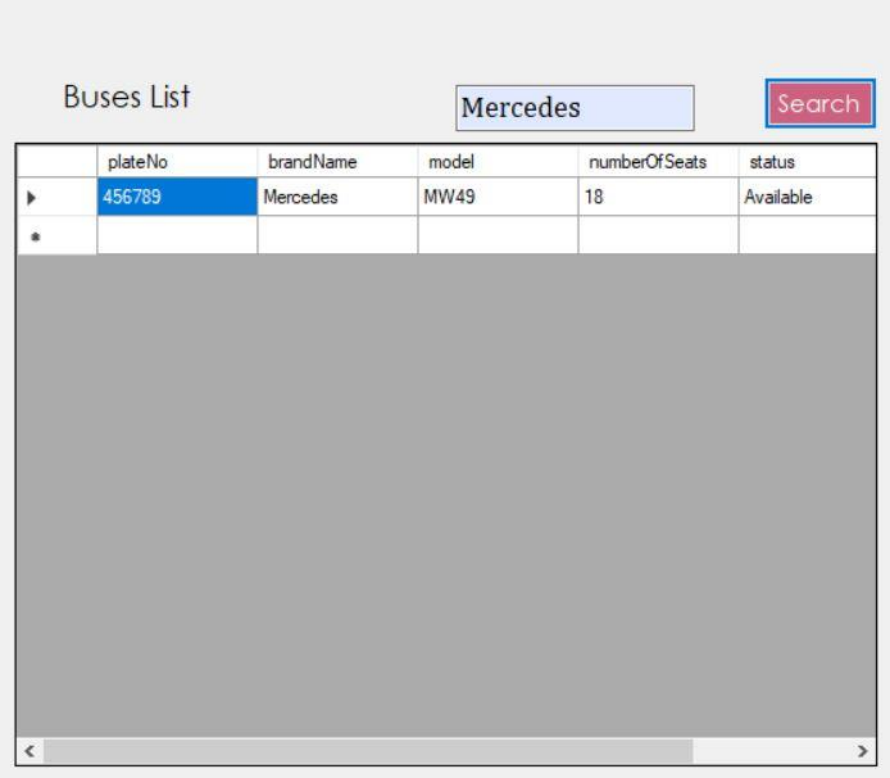
6. Bir şöfürün maaşı güncellendiği zaman değişiklikleri izlemek için bir tabloda bilgileri saklanır .

```
CREATE TABLE "public"."SalaryChanges" (
        "recordNo" serial,
        "driverID" SmallInt NOT NULL,
        "oldSalary" Real NOT NULL,
        "newSalary" Real NOT NULL,
        "updatedOn" TIMESTAMP NOT NULL,
        CONSTRAINT "PK" PRIMARY KEY ("recordNo")
);


CREATE OR REPLACE FUNCTION "salaryChangesTR"()
RETURNS TRIGGER
AS
$$
BEGIN
  IF NEW."salary" <> OLD."salary" THEN
    INSERT INTO "SalaryChanges"("driverID", "oldSalary", "newSalary", "updatedOn")
    VALUES(OLD."driverID", OLD."salary", NEW."salary", CURRENT_TIMESTAMP::TIMESTAMP);
  END IF;


  RETURN NEW;
END;
$$
LANGUAGE "plpgsql";
CREATE TRIGGER "onSalaryUpdate"
BEFORE UPDATE ON "Driver"
FOR EACH ROW
EXECUTE PROCEDURE "salaryChangesTR"();
```

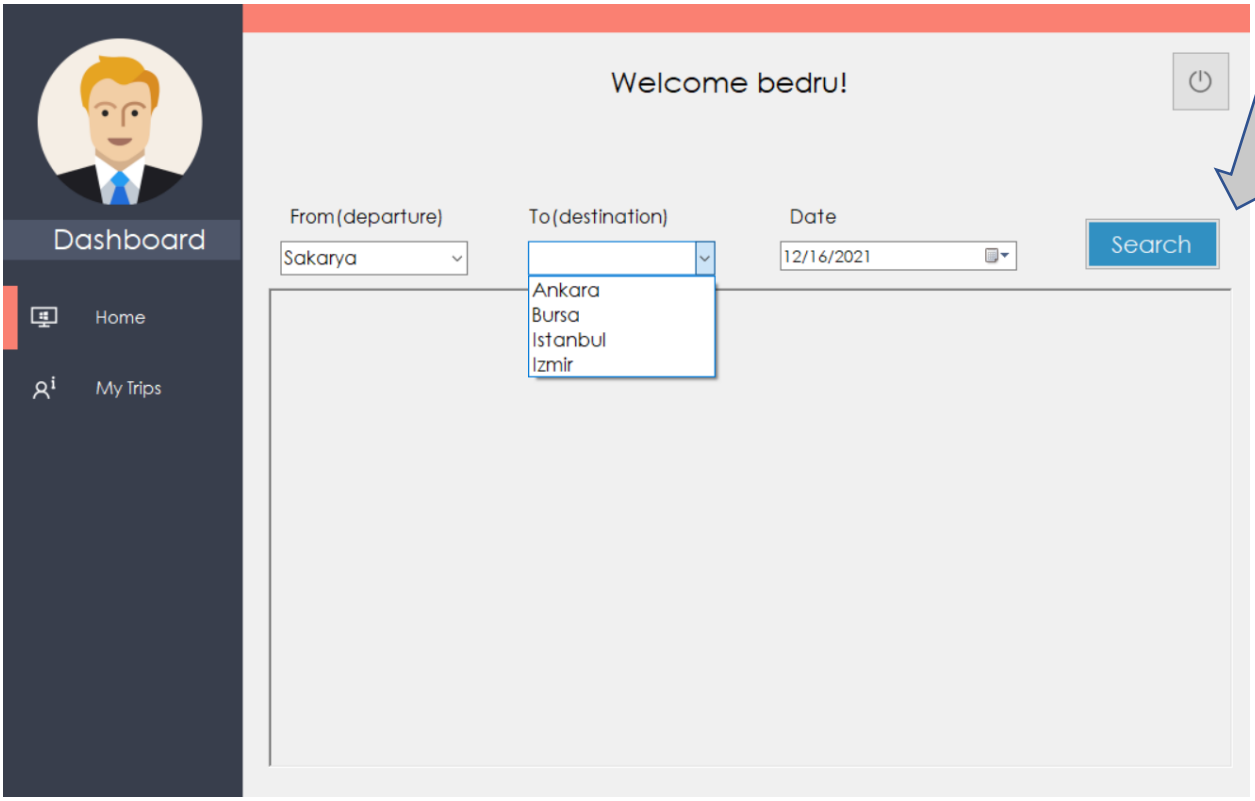
1. Arama işlemi

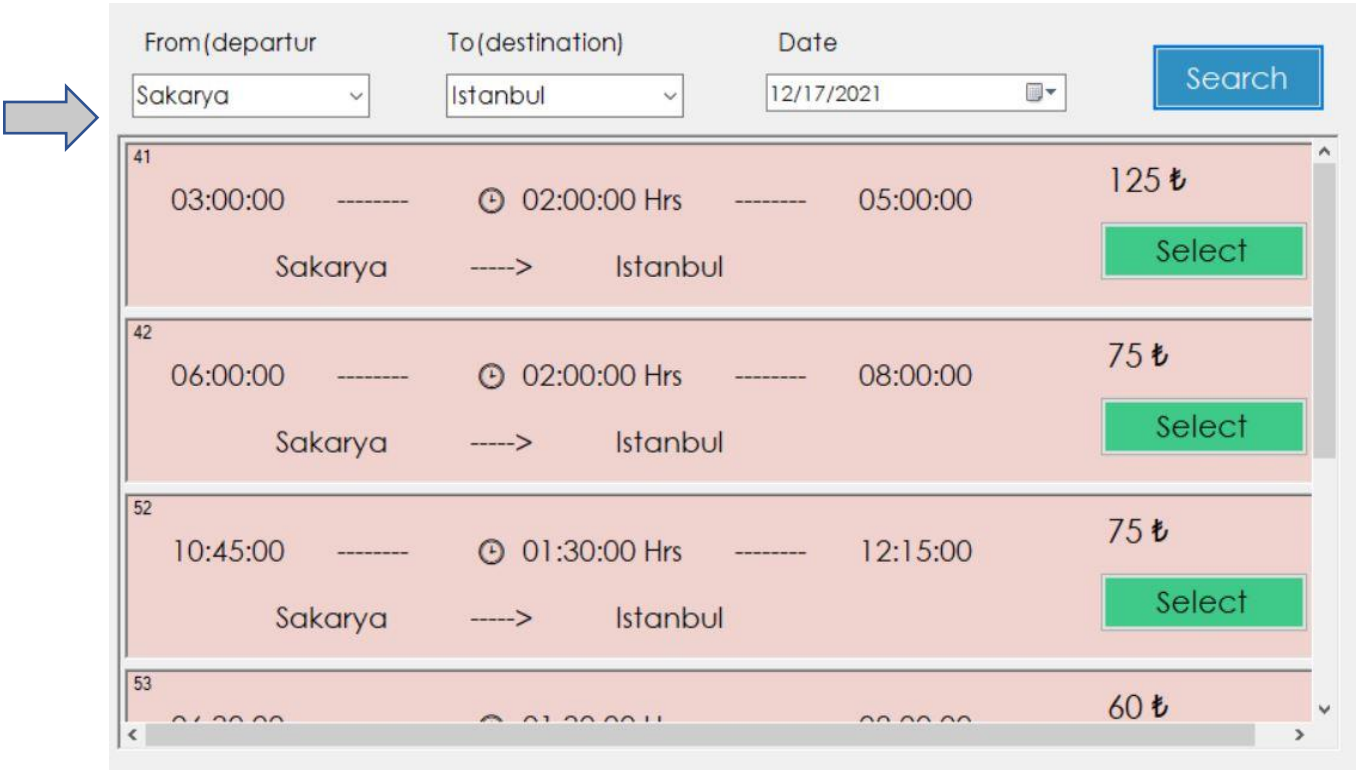
1.1 Otobüs arama…





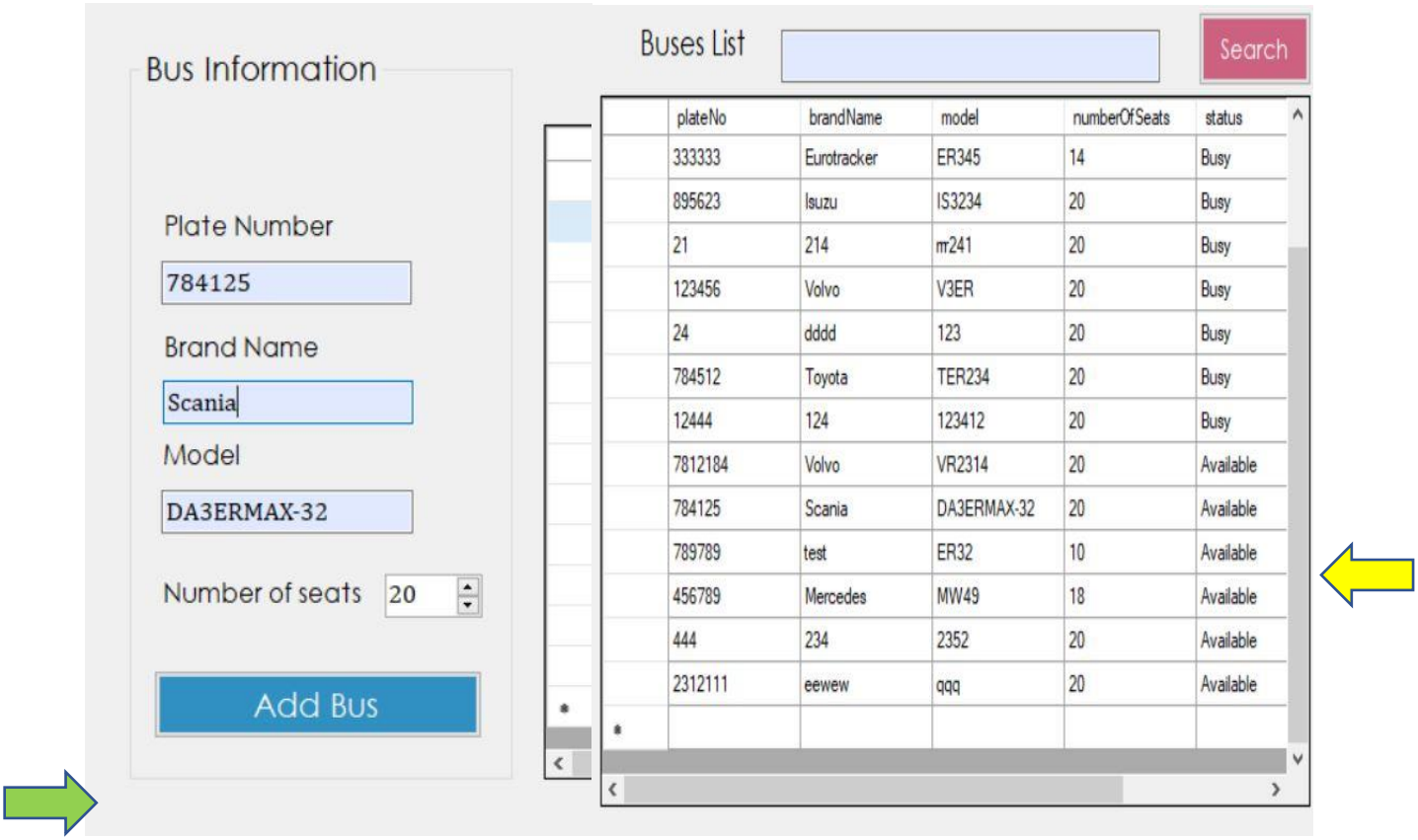
1.2 Seferler arama...



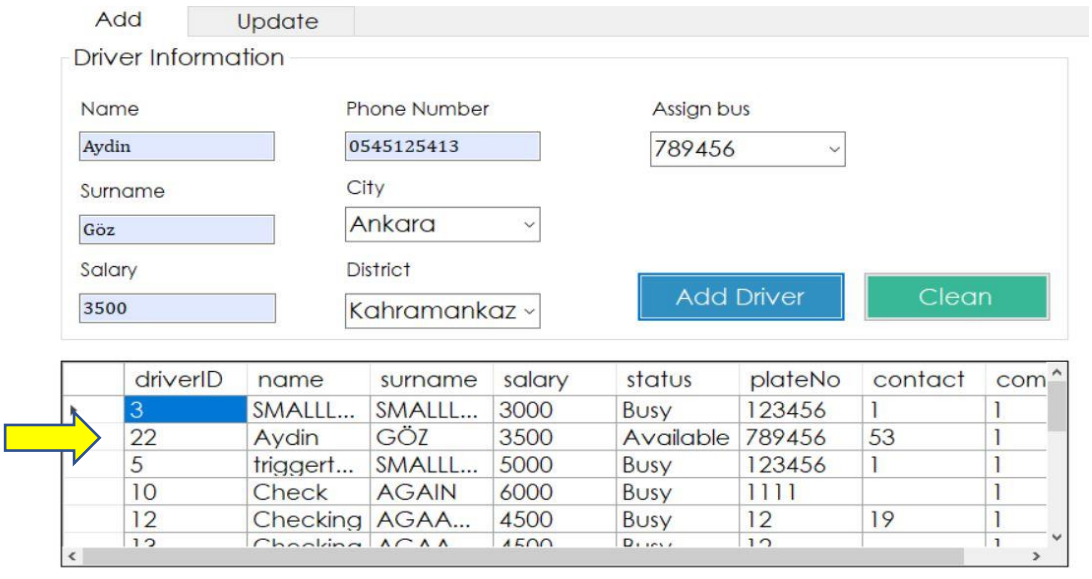


2. Ekleme işlemi

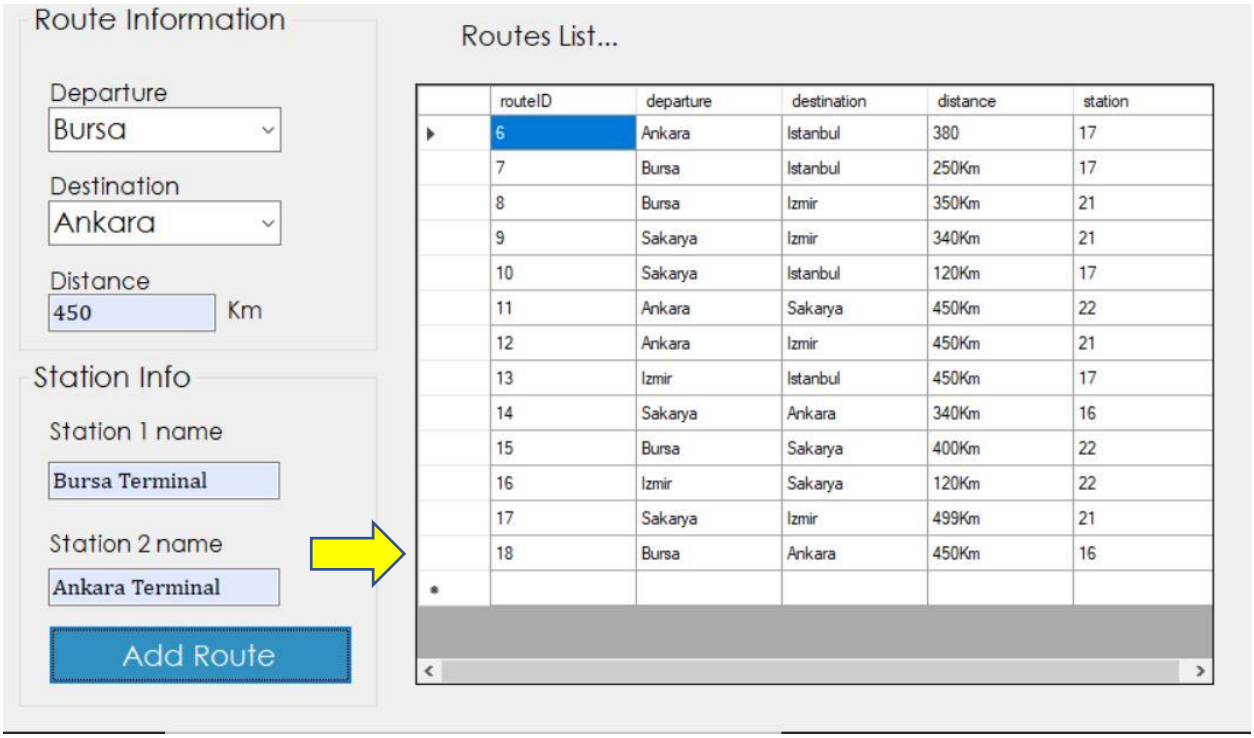
2.1 Otobüs ekleme…



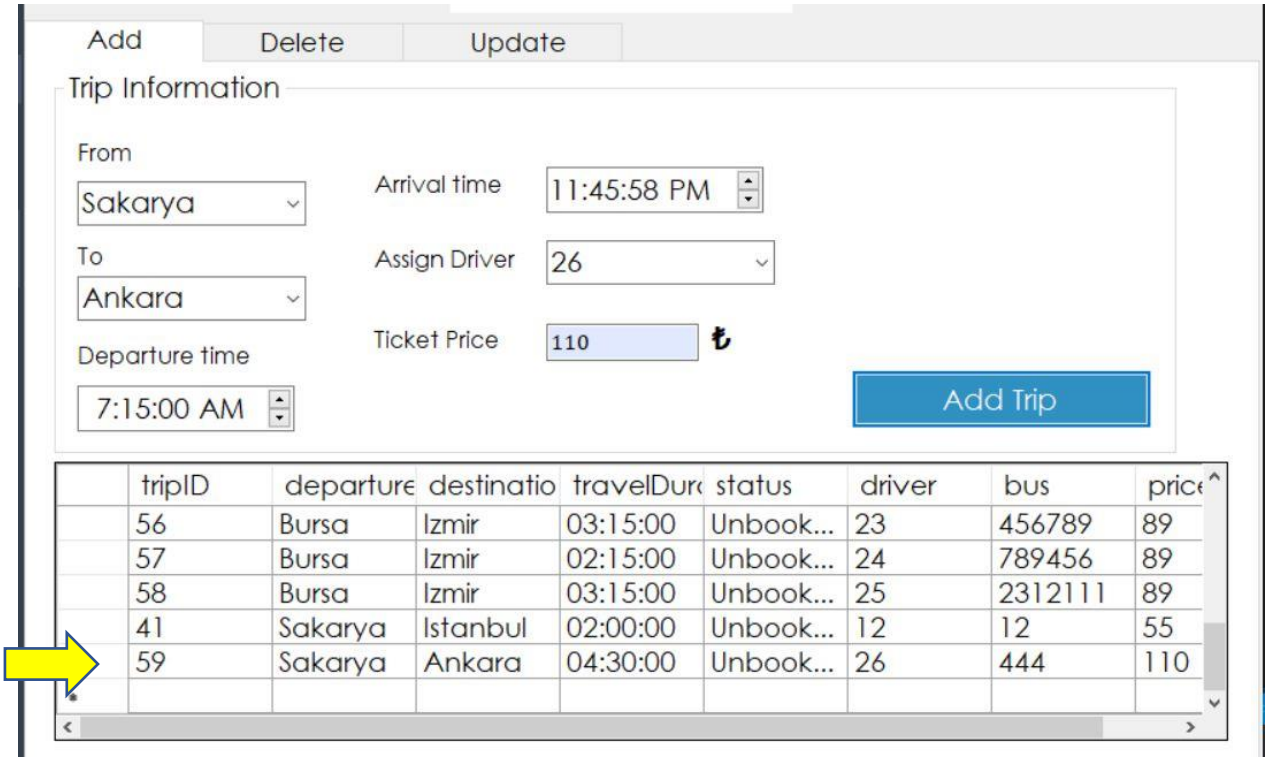
2.2 Şoför ekleme…



2.3 Güzergah ekleme…

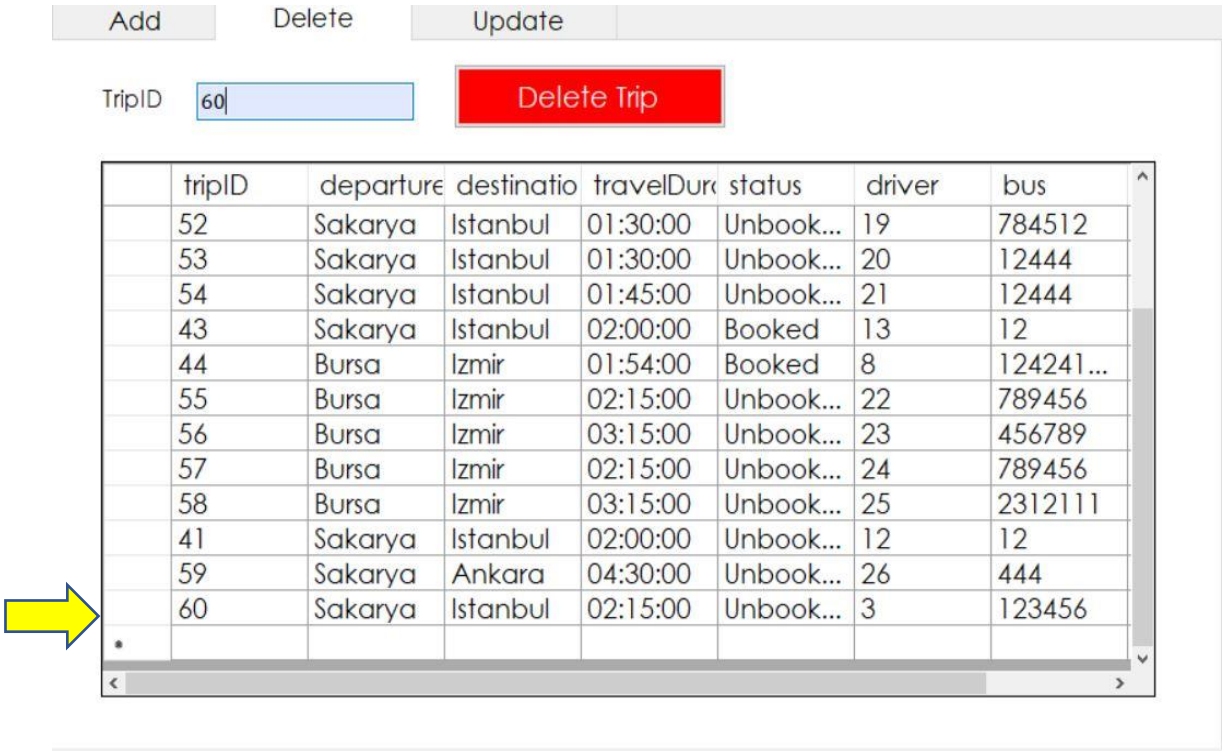


2.4 Sefer ekleme…

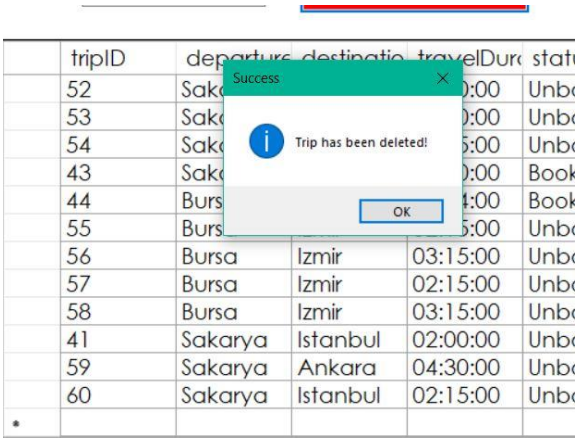


3. Silme işlemi

Sefer silme…



tripID	departure	destinatio	travelDur	status	driver	bus
52	Sakarya	Istanbul	01:30:00	Unbook...	19	784512
53	Sakarya	Istanbul	01:30:00	Unbook...	20	12444
54	Sakarya	Istanbul	01:45:00	Unbook...	21	12444
43	Sakarya	Istanbul	02:00:00	Booked	13	12
44	Bursa	Izmir	01:54:00	Booked	8	124241...
55	Bursa	Izmir	02:15:00	Unbook...	22	789456
56	Bursa	Izmir	03:15:00	Unbook...	23	456789
57	Bursa	Izmir	02:15:00	Unbook...	24	789456
58	Bursa	Izmir	03:15:00	Unbook...	25	2312111
41	Sakarya	Istanbul	02:00:00	Unbook...	12	12
59	Sakarya	Ankara	04:30:00	Unbook...	26	444
60	Sakarya	Istanbul	02:15:00	Unbook...	3	123456



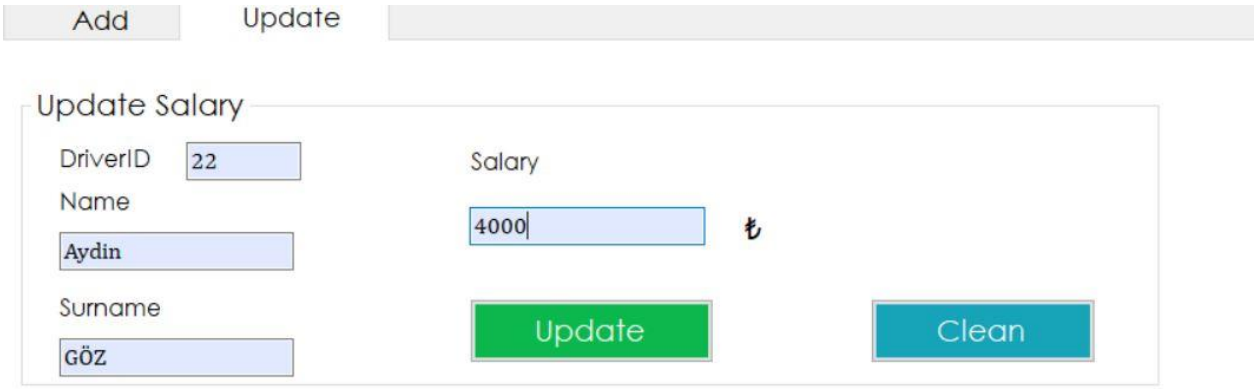
tripID	departure	destinatio	travelDur	stat
52	Sak		:00	Unb
53	Sak		:00	Unb
54	Sak		:00	Unb
43	Sak		:00	Book
44	Burs		:00	Book
55	Burs		:00	Unb
56	Bursa	Izmir	03:15:00	Unb
57	Bursa	Izmir	02:15:00	Unb
58	Bursa	Izmir	03:15:00	Unb
41	Sakarya	Istanbul	02:00:00	Unb
59	Sakarya	Ankara	04:30:00	Unb
60	Sakarya	Istanbul	02:15:00	Unb

Silindi…

58	Bursa	Izmir	03:15:00	Unbook...	25	2312111
41	Sakarya	Istanbul	02:00:00	Unbook...	12	12
59	Sakarya	Ankara	04:30:00	Unbook...	26	444

4. Güncelleme işlemi

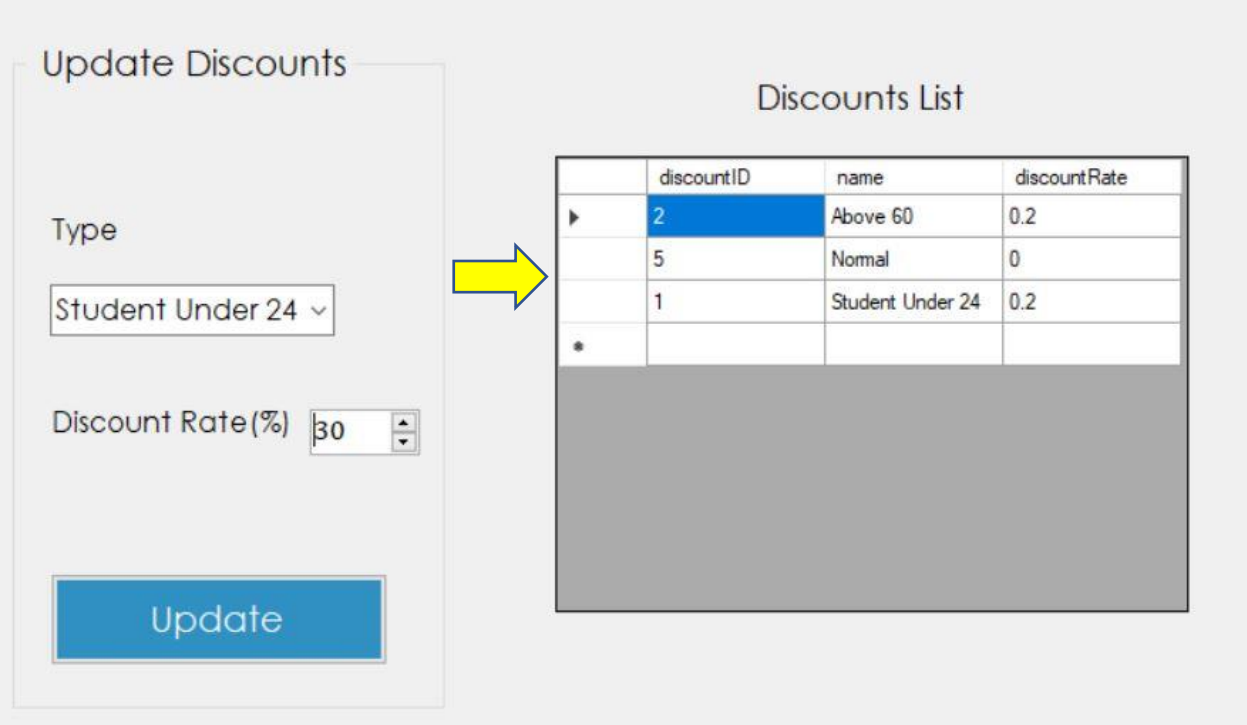
4.1 Şoför maaşı güncelleme…

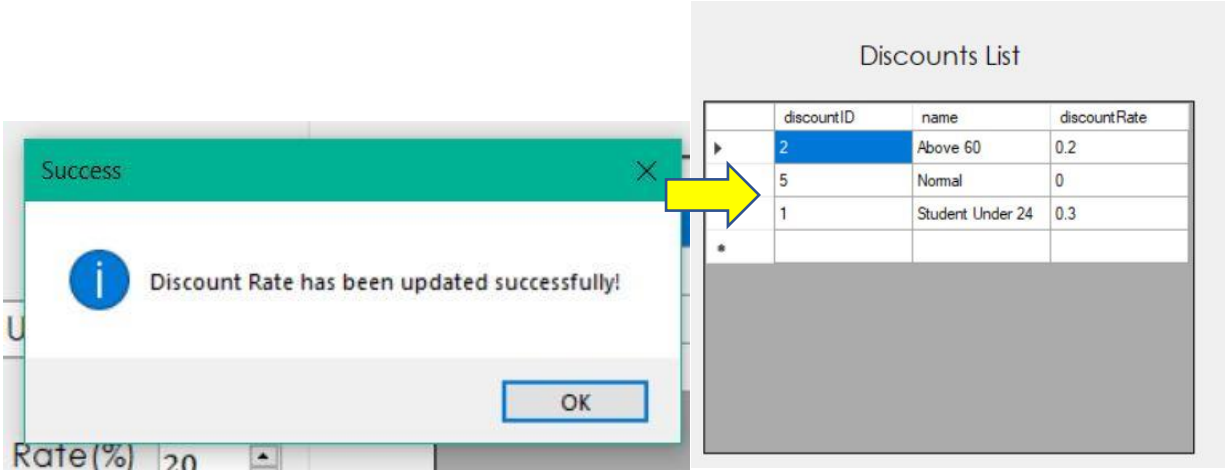


driverID	name	surname	salary	status	plateNo	contact	compc
22	Aydin	GÖZ	3500	Busy	789456	53	1
23	Wasiq	MASOOD	5000	Busy	456789	54	1
24	Mehmet	OĞUZ	4500	Busy	789456	55	1
25	Murat	YILMAZ	4700	Busy	2312111	56	1
27	Andy	ROBERTSON	4900	Available	784125	58	1
28	Mert	YILMAZ	5500	Available	4515699	59	1

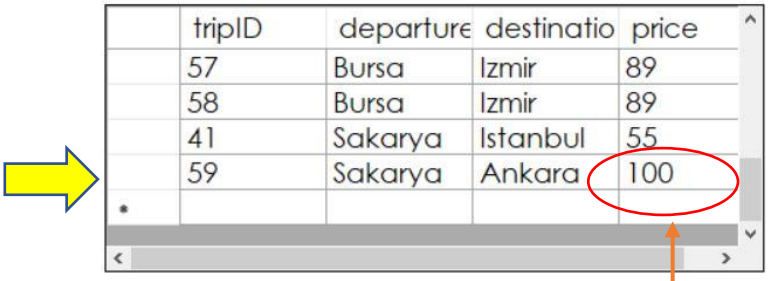
driverID	name	surname	salary	status	plateNo	contact	compc
29	Ibrahim	YAPMAZ	6500	Available	7812184	60	1
26	Alexander	ARNOLD	7800	Busy	444	57	1
3	SMALLLETTERS	SMALLLETTERS	3000	Available	123456	1	1
22	Aydin	GÖZ	4000	Busy	789456	53	1
5	triggertest	SMALLLETTERS	5000	Busy	123456	1	1
10	Check	AGAIN	6000	Busy	1111		1

4.2 İndirim oranı güncelleme…

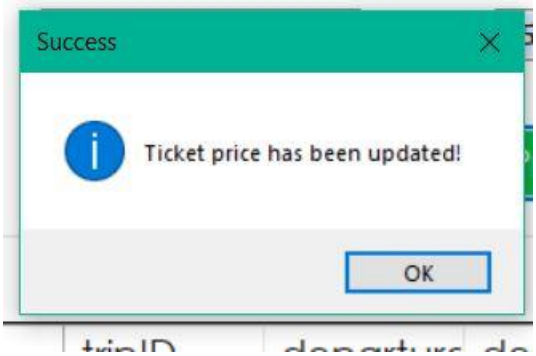




4.3 Seferin fiyatı güncelleme…



tripID	departure	destinatio	price
57	Bursa	Izmir	89
58	Bursa	Izmir	89
41	Sakarya	Istanbul	55
59	Sakarya	Ankara	100



tripID	departure	destinatio	price
57	Bursa	Izmir	89
58	Bursa	Izmir	89
41	Sakarya	Istanbul	55
59	Sakarya	Ankara	150

Github link :

Youtube video link :