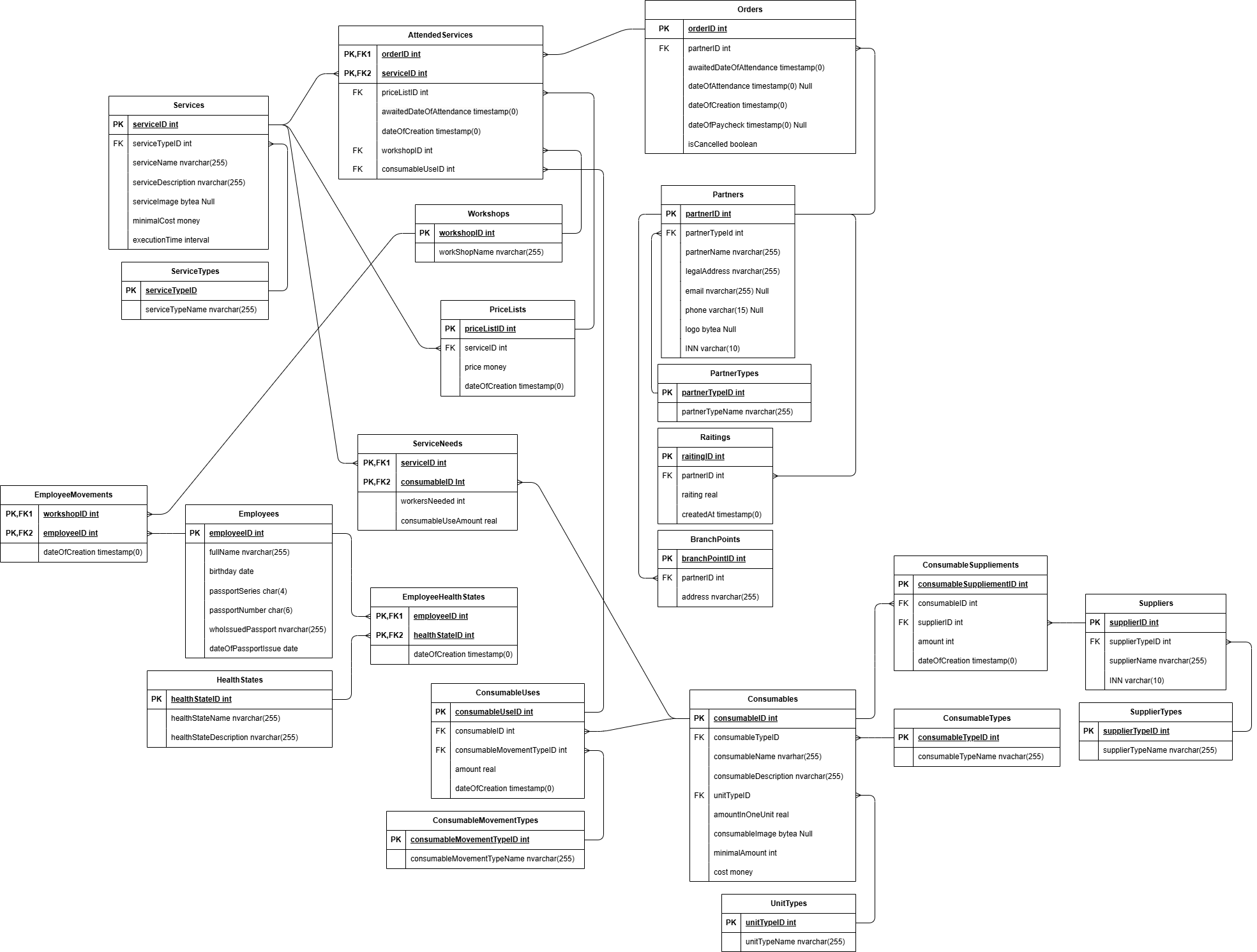
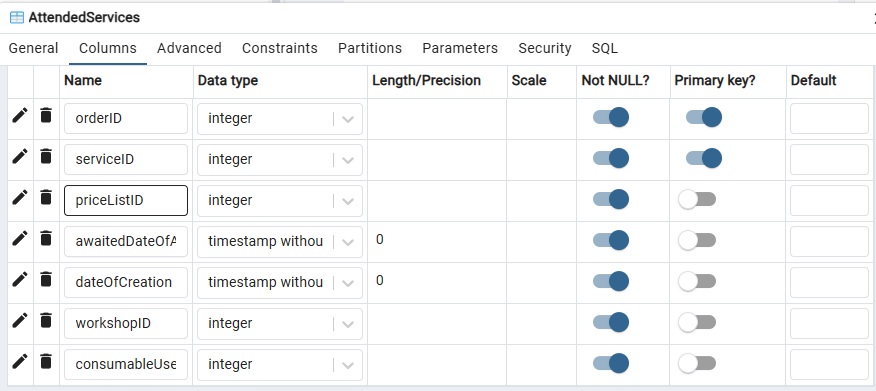
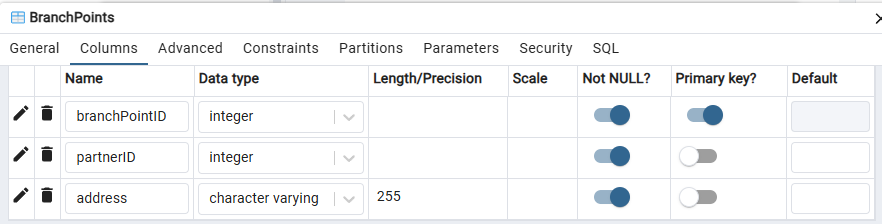
**ER диаграмма БД**

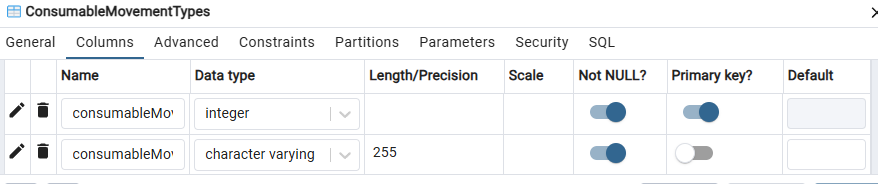


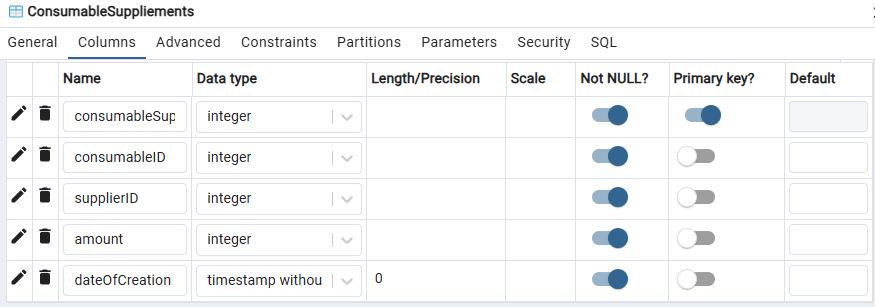
**Словарь данных**

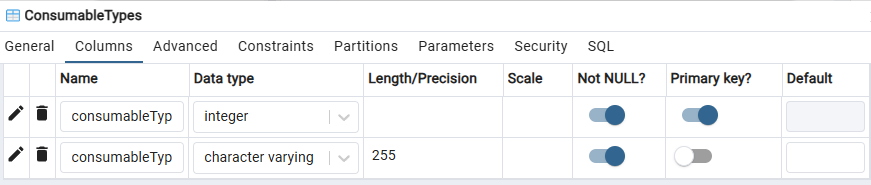
Скриншоты из pgAdmin

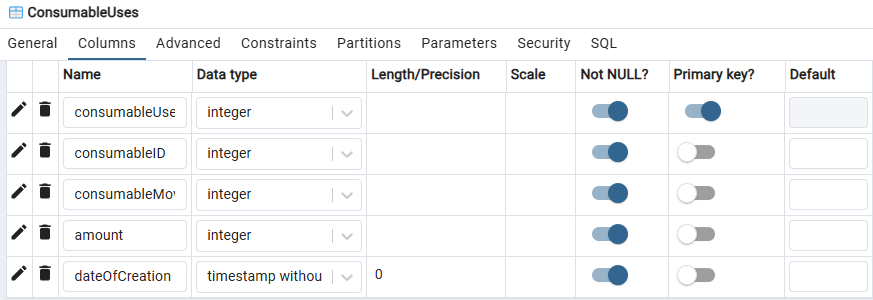


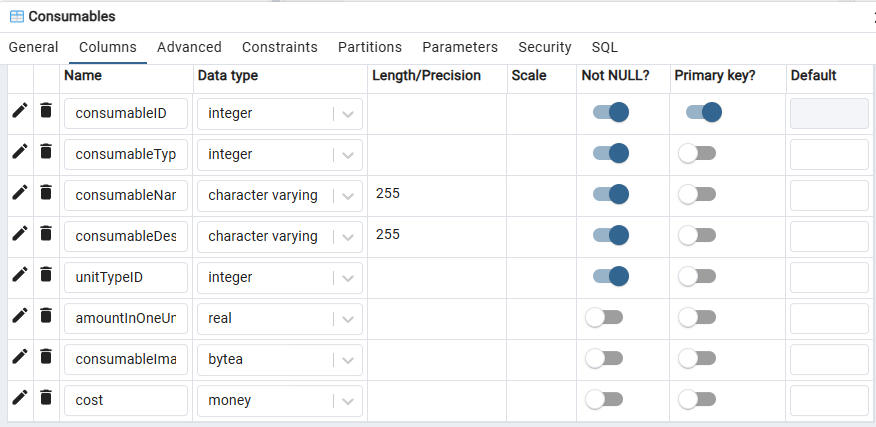


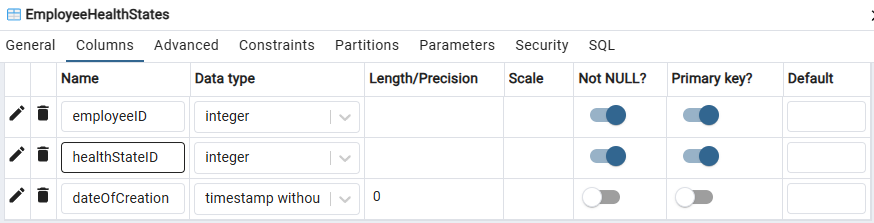


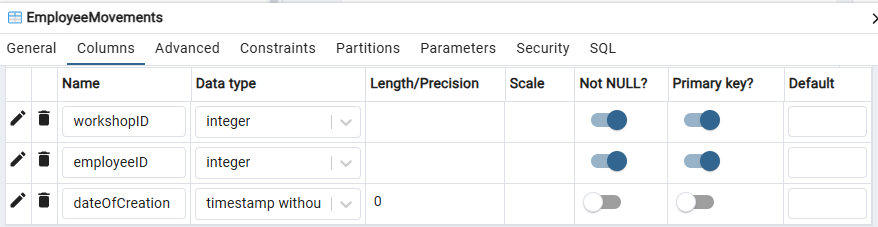


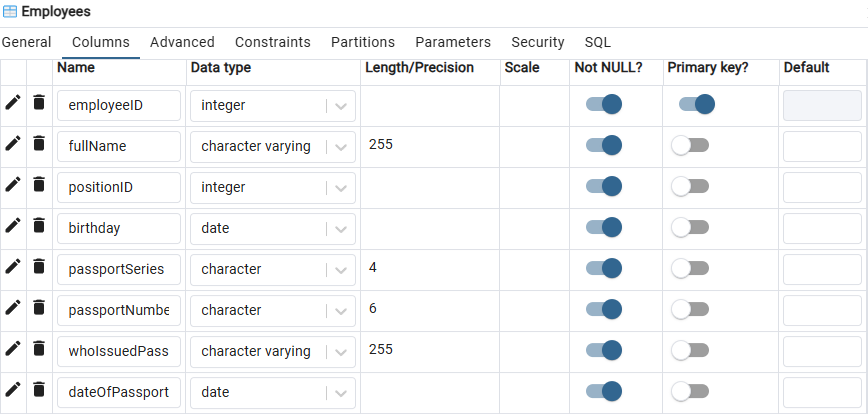


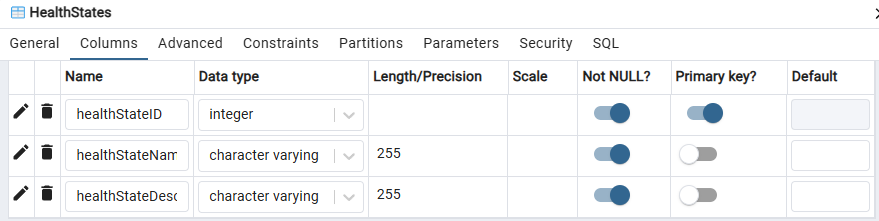


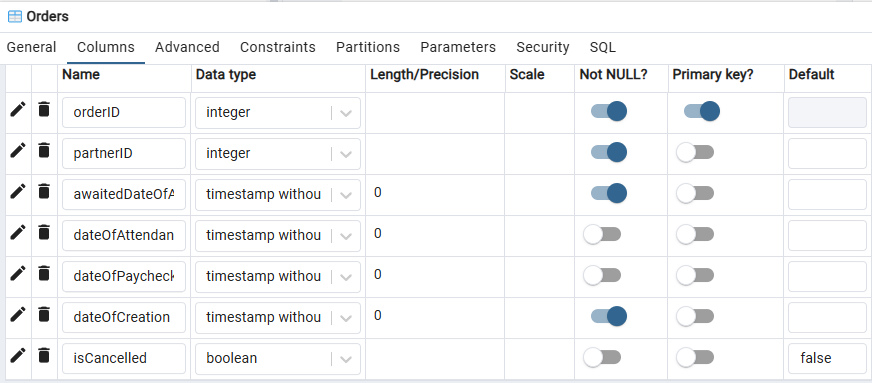


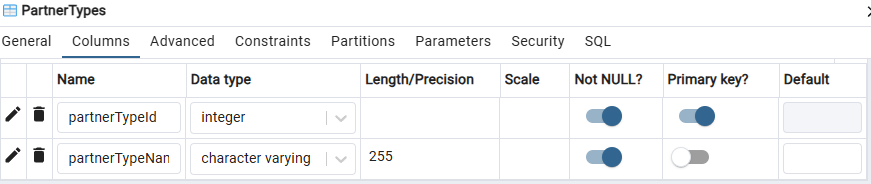


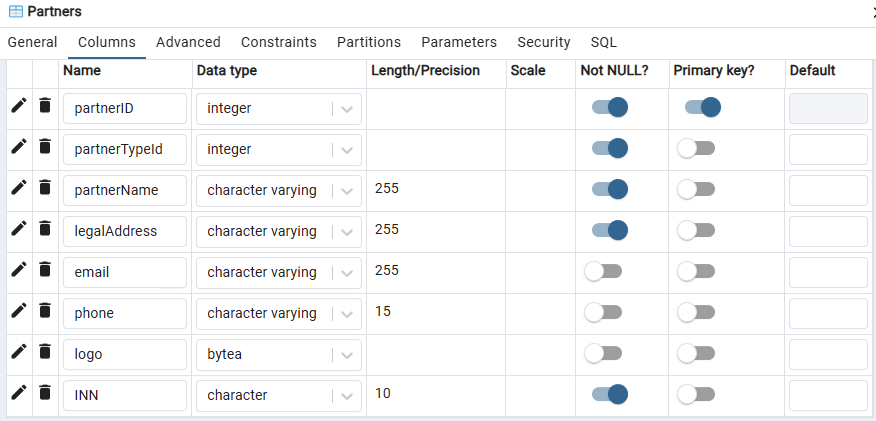


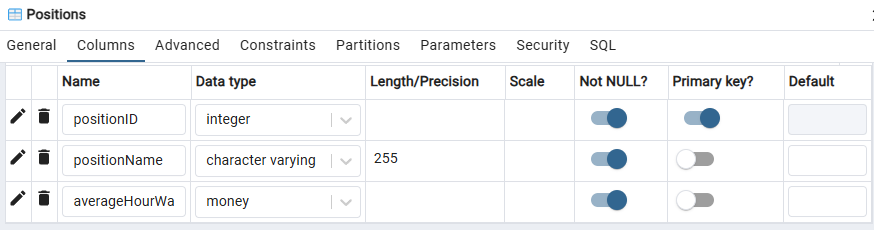


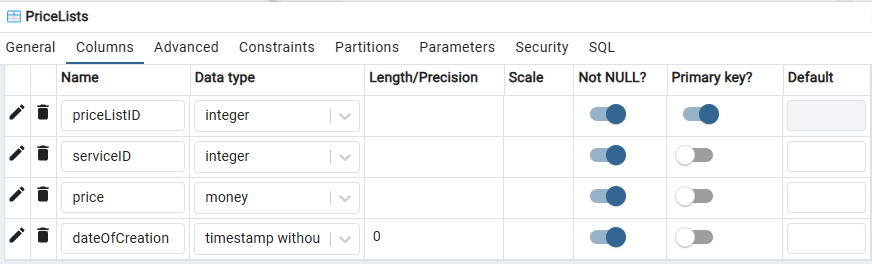


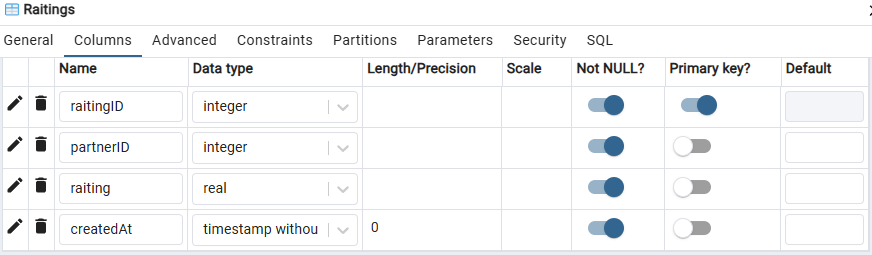


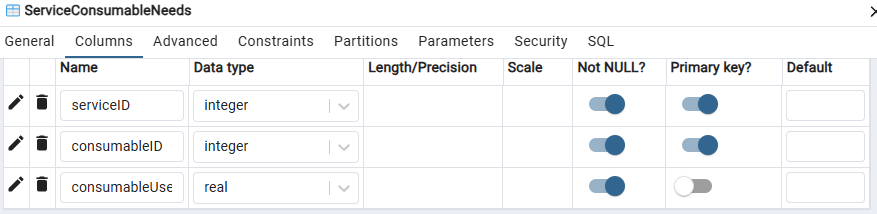


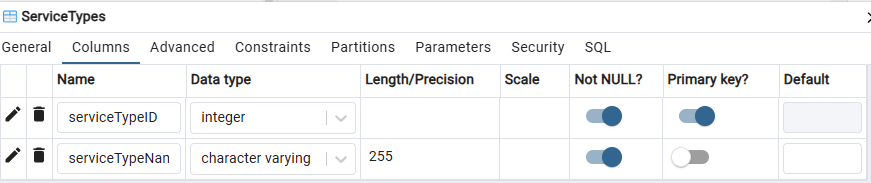


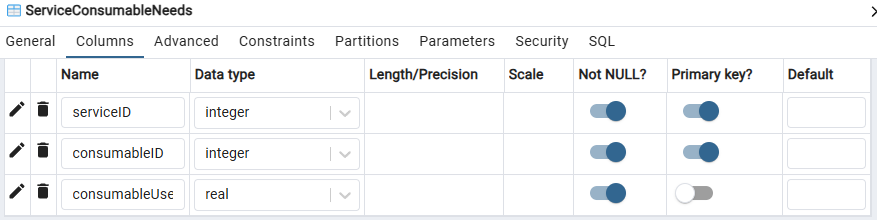


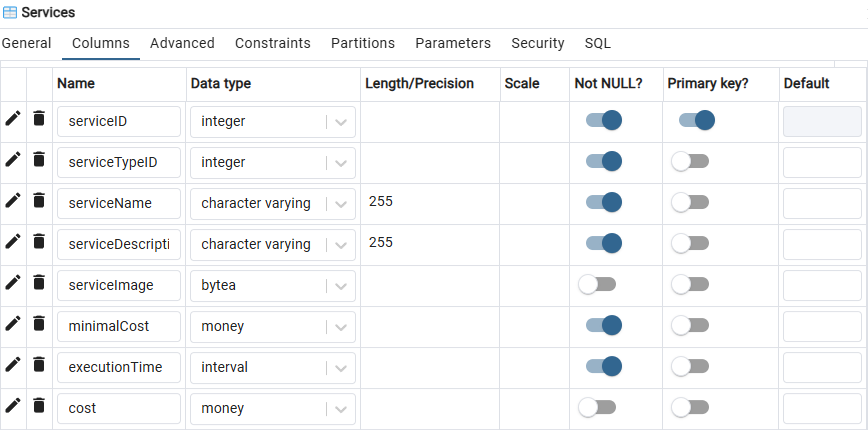








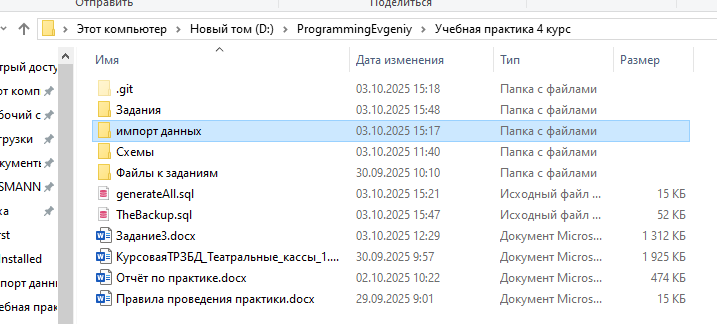




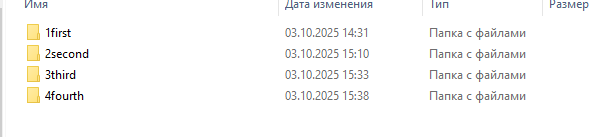
**Файлы для импорта**

(в репозитории)

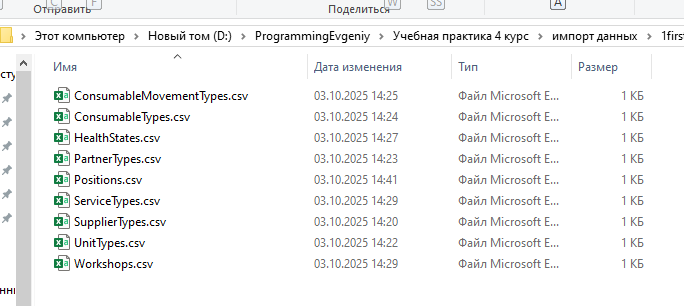
В папке импорт данных



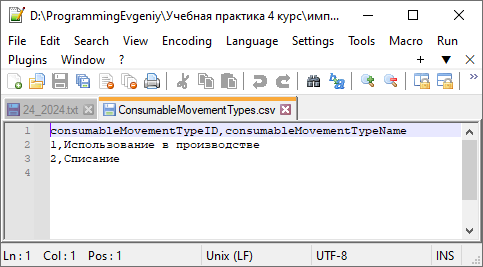
Есть 4 папки. При загрузке данных в БД нужно будет идти по порядку папок, чтоб соблюдать ограничения внешних ключей.



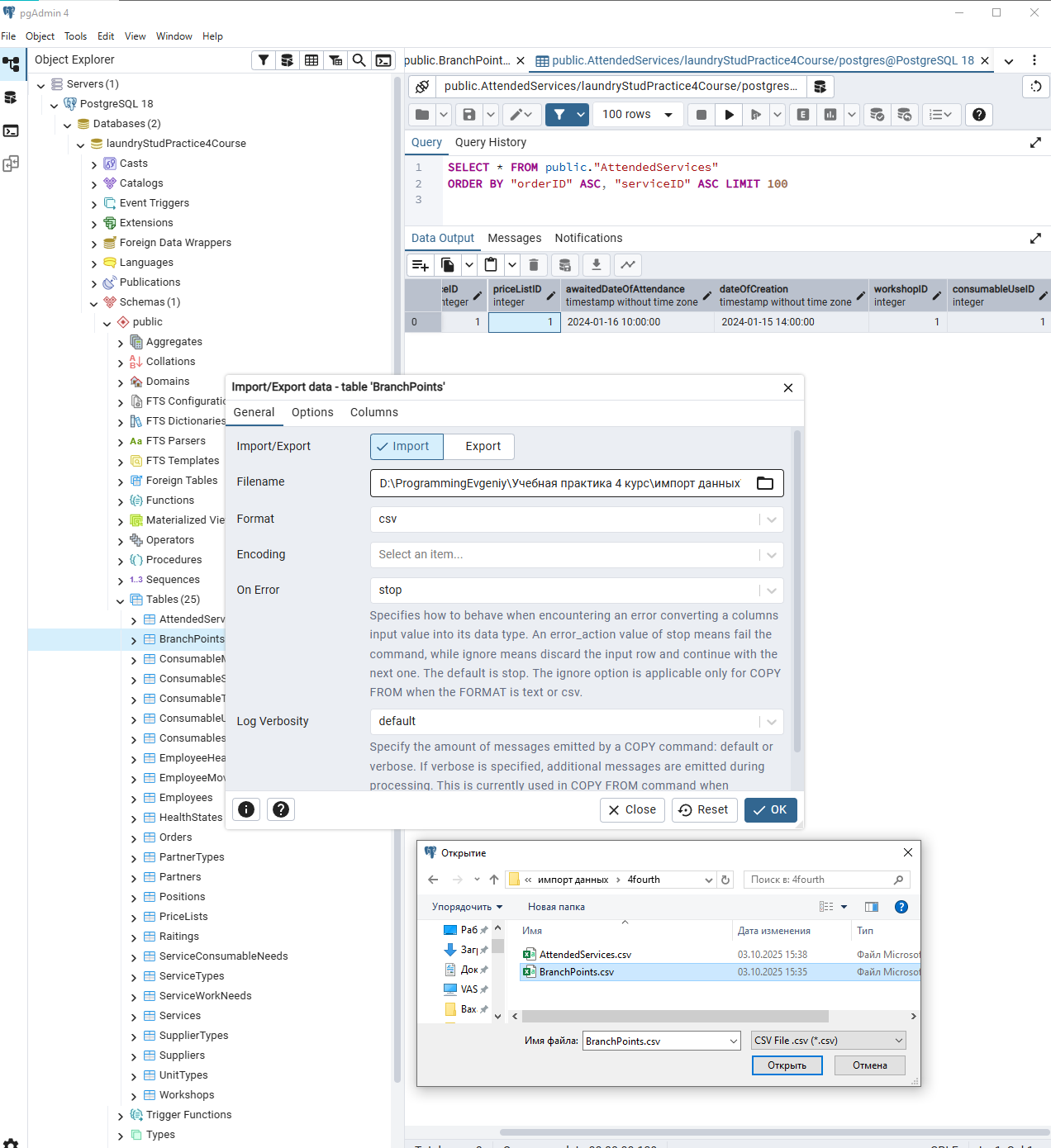
Соответсвующие данные для импорта названы так же, как и таблицы

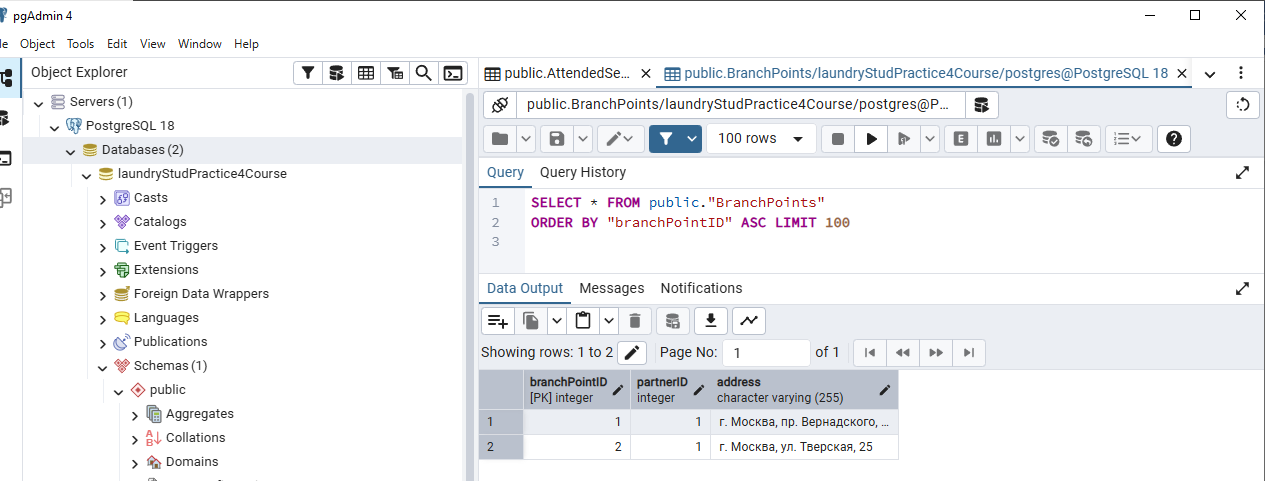
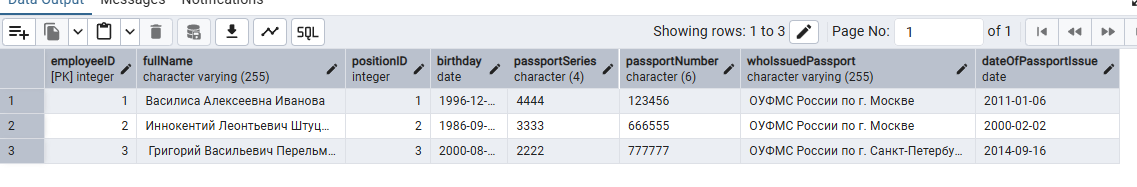


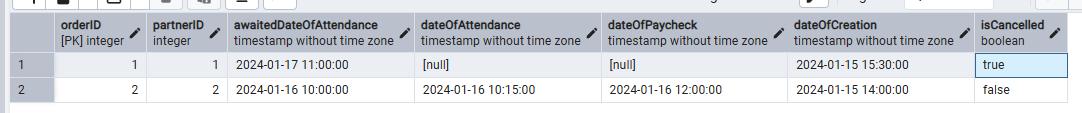
Пример содержания такого файла

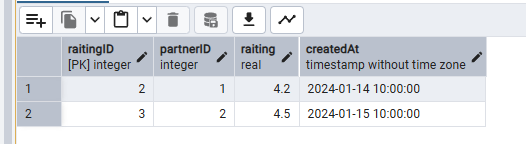


Пример импорта данных через PG admin 4 (клиент для управления БД)



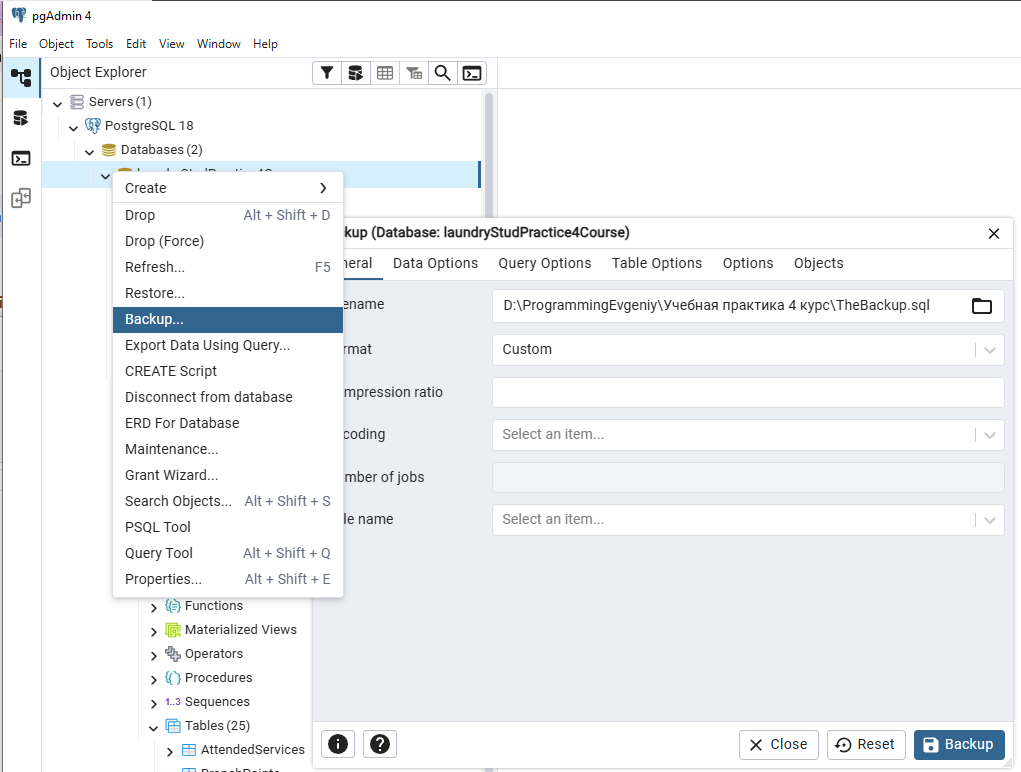
  
Пример заполненных данных:  


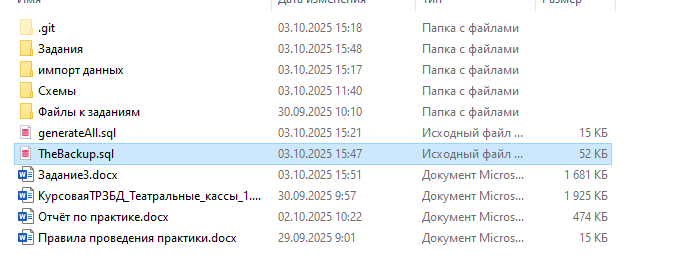




И так далее

**Backup БД**

****

****

**SQL скрипт**

/\*    Создание независимых таблиц   \*/

CREATE TABLE public."SupplierTypes"

(

    "supplierTypeID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "supplierTypeName" character varying(255) NOT NULL,

    PRIMARY KEY ("supplierTypeID")

);

ALTER TABLE IF EXISTS public."SupplierTypes"

    OWNER to postgres;

CREATE TABLE public."ConsumableTypes"

(

    "consumableTypeID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "consumableTypeName" character varying(255) NOT NULL,

    PRIMARY KEY ("consumableTypeID")

);

ALTER TABLE IF EXISTS public."ConsumableTypes"

    OWNER to postgres;

CREATE TABLE public."UnitTypes"

(

    "unitTypeID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "unitTypeName" character varying(255) NOT NULL,

    PRIMARY KEY ("unitTypeID")

);

ALTER TABLE IF EXISTS public."UnitTypes"

    OWNER to postgres;

CREATE TABLE public."PartnerTypes"

(

    "partnerTypeId" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "partnerTypeName" character varying(255) NOT NULL,

    PRIMARY KEY ("partnerTypeId")

);

ALTER TABLE IF EXISTS public."PartnerTypes"

    OWNER to postgres;

CREATE TABLE public."ConsumableMovementTypes"

(

    "consumableMovementTypeID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "consumableMovementTypeName" character varying(255) NOT NULL,

    PRIMARY KEY ("consumableMovementTypeID")

);

ALTER TABLE IF EXISTS public."ConsumableMovementTypes"

    OWNER to postgres;

CREATE TABLE public."HealthStates"

(

    "healthStateID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "healthStateName" character varying(255) NOT NULL,

    "healthStateDescription" character varying(255) NOT NULL,

    PRIMARY KEY ("healthStateID")

);

ALTER TABLE IF EXISTS public."HealthStates"

    OWNER to postgres;

CREATE TABLE public."ServiceTypes"

(

    "serviceTypeID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "serviceTypeName" character varying(255) NOT NULL,

    PRIMARY KEY ("serviceTypeID")

);

ALTER TABLE IF EXISTS public."ServiceTypes"

    OWNER to postgres;

CREATE TABLE public."Workshops"

(

    "workshopID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "workShopName" character varying(255) NOT NULL,

    PRIMARY KEY ("workshopID")

);

ALTER TABLE IF EXISTS public."Workshops"

    OWNER to postgres;

CREATE TABLE public."Positions"

(

    "positionID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "positionName" character varying(255) NOT NULL,

    "averageHourWage" money NOT NULL,

    PRIMARY KEY ("positionID")

);

ALTER TABLE IF EXISTS public."Positions"

    OWNER to postgres;

/\*    Создание независимых таблиц   \*/

/\*    Создание зависимых таблиц первого порядка   \*/

CREATE TABLE public."Employees"

(

    "employeeID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "fullName" character varying(255) NOT NULL,

    "positionID" integer NOT NULL,

    "birthday" date NOT NULL,

    "passportSeries" character(4) NOT NULL,

    "passportNumber" character(6) NOT NULL,

    "whoIssuedPassport" character varying(255) NOT NULL,

    "dateOfPassportIssue" date NOT NULL,

    PRIMARY KEY ("employeeID"),

    CONSTRAINT "FK\_Positions" FOREIGN KEY ("positionID")

        REFERENCES public."Positions" ("positionID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Employees"

    OWNER to postgres;

CREATE TABLE public."Suppliers"

(

    "supplierID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "supplierTypeID" integer NOT NULL,

    "supplierName" character varying(255) NOT NULL,

    "INN" character(10) NOT NULL,

    PRIMARY KEY ("supplierID"),

    CONSTRAINT "FK\_SupplierTypes" FOREIGN KEY ("supplierTypeID")

        REFERENCES public."SupplierTypes" ("supplierTypeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Suppliers"

    OWNER to postgres;

CREATE TABLE public."Consumables"

(

    "consumableID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "consumableTypeID" integer NOT NULL,

    "consumableName" character varying(255) NOT NULL,

    "consumableDescription" character varying(255) NOT NULL,

    "unitTypeID" integer NOT NULL,

    "amountInOneUnit" real,

    "consumableImage" bytea,

    cost money,

    PRIMARY KEY ("consumableID"),

    CONSTRAINT "FK\_ConsumableTypes" FOREIGN KEY ("consumableTypeID")

        REFERENCES public."ConsumableTypes" ("consumableTypeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_UnitTypes" FOREIGN KEY ("unitTypeID")

        REFERENCES public."UnitTypes" ("unitTypeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Consumables"

    OWNER to postgres;

CREATE TABLE public."Partners"

(

    "partnerID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "partnerTypeId" integer NOT NULL,

    "partnerName" character varying(255) NOT NULL,

    "legalAddress" character varying(255) NOT NULL,

    "email" character varying(255),

    "phone" character varying(15),

    "logo" bytea,

    "INN" character(10) NOT NULL,

    PRIMARY KEY ("partnerID"),

    CONSTRAINT "FK\_PartnerTypes" FOREIGN KEY ("partnerTypeId")

        REFERENCES public."PartnerTypes" ("partnerTypeId") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Partners"

    OWNER to postgres;

CREATE TABLE public."Services"

(

    "serviceID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "serviceTypeID" integer NOT NULL,

    "serviceName" character varying(255) NOT NULL,

    "serviceDescription" character varying(255) NOT NULL,

    "serviceImage" bytea,

    "minimalCost" money NOT NULL,

    "executionTime" interval NOT NULL,

    cost money,

    PRIMARY KEY ("serviceID"),

    CONSTRAINT "FK\_ServiceTypes" FOREIGN KEY ("serviceTypeID")

        REFERENCES public."ServiceTypes" ("serviceTypeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Services"

    OWNER to postgres;

CREATE TABLE public."EmployeeHealthStates"

(

    "employeeID" integer NOT NULL,

    "healthStateID" integer NOT NULL,

    "dateOfCreation" timestamp(0) without time zone,

    PRIMARY KEY ("employeeID", "healthStateID"),

    CONSTRAINT "FK\_Employees" FOREIGN KEY ("employeeID")

        REFERENCES public."Employees" ("employeeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_HealthStates" FOREIGN KEY ("healthStateID")

        REFERENCES public."HealthStates" ("healthStateID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."EmployeeHealthStates"

    OWNER to postgres;

CREATE TABLE public."EmployeeMovements"

(

    "workshopID" integer NOT NULL,

    "employeeID" integer NOT NULL,

    "dateOfCreation" timestamp(0) without time zone,

    PRIMARY KEY ("workshopID", "employeeID"),

    CONSTRAINT "FK\_Workshops" FOREIGN KEY ("workshopID")

        REFERENCES public."Workshops" ("workshopID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_Employees" FOREIGN KEY ("employeeID")

        REFERENCES public."Employees" ("employeeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."EmployeeMovements"

    OWNER to postgres;

/\*    Создание зависимых таблиц первого порядка   \*/

/\*    Создание зависимых таблиц второго порядка   \*/

CREATE TABLE public."Raitings"

(

    "raitingID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "partnerID" integer NOT NULL,

    "raiting" real NOT NULL,

    "createdAt" timestamp(0) without time zone NOT NULL,

    PRIMARY KEY ("raitingID"),

    CONSTRAINT "FK\_Partners" FOREIGN KEY ("partnerID")

        REFERENCES public."Partners" ("partnerID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Raitings"

    OWNER to postgres;

CREATE TABLE public."ConsumableSuppliements"

(

    "consumableSuppliementID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "consumableID" integer NOT NULL,

    "supplierID" integer NOT NULL,

    "amount" integer NOT NULL,

    "dateOfCreation" timestamp(0) without time zone NOT NULL,

    PRIMARY KEY ("consumableSuppliementID"),

    CONSTRAINT "FK\_Consumables" FOREIGN KEY ("consumableID")

        REFERENCES public."Consumables" ("consumableID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_Suppliers" FOREIGN KEY ("supplierID")

        REFERENCES public."Suppliers" ("supplierID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."ConsumableSuppliements"

    OWNER to postgres;

CREATE TABLE public."ConsumableUses"

(

    "consumableUseID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "consumableID" integer NOT NULL,

    "consumableMovementTypeID" integer NOT NULL,

    "amount" integer NOT NULL,

    "dateOfCreation" timestamp(0) without time zone NOT NULL,

    PRIMARY KEY ("consumableUseID"),

    CONSTRAINT "FK\_Consumables" FOREIGN KEY ("consumableID")

        REFERENCES public."Consumables" ("consumableID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_ConsumableMovementTypes" FOREIGN KEY ("consumableMovementTypeID")

        REFERENCES public."ConsumableMovementTypes" ("consumableMovementTypeID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."ConsumableUses"

    OWNER to postgres;

CREATE TABLE public."Orders"

(

    "orderID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "partnerID" integer NOT NULL,

    "awaitedDateOfAttendance" timestamp(0) without time zone NOT NULL,

    "dateOfAttendance" timestamp(0) without time zone,

    "dateOfPaycheck" timestamp(0) without time zone,

    "dateOfCreation" timestamp(0) without time zone NOT NULL,

    "isCancelled" boolean DEFAULT false,

    PRIMARY KEY ("orderID"),

    CONSTRAINT "FK\_Partners" FOREIGN KEY ("partnerID")

        REFERENCES public."Partners" ("partnerID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."Orders"

    OWNER to postgres;

CREATE TABLE public."PriceLists"

(

    "priceListID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "serviceID" integer NOT NULL,

    "price" money NOT NULL,

    "dateOfCreation" timestamp(0) without time zone NOT NULL,

    PRIMARY KEY ("priceListID"),

    CONSTRAINT "FK\_Services" FOREIGN KEY ("serviceID")

        REFERENCES public."Services" ("serviceID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."PriceLists"

    OWNER to postgres;

CREATE TABLE public."ServiceWorkNeeds"

(

    "serviceID" integer NOT NULL,

    "positionID" integer NOT NULL,

    "overalWorkTime" interval NOT NULL,

    PRIMARY KEY ("serviceID", "positionID"),

    CONSTRAINT "FK\_Services" FOREIGN KEY ("serviceID")

        REFERENCES public."Services" ("serviceID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_Positions" FOREIGN KEY ("positionID")

        REFERENCES public."Positions" ("positionID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."ServiceWorkNeeds"

    OWNER to postgres;

CREATE TABLE public."ServiceConsumableNeeds"

(

    "serviceID" integer NOT NULL,

    "consumableID" integer NOT NULL,

    "consumableUseAmount" real NOT NULL,

    PRIMARY KEY ("serviceID", "consumableID"),

    CONSTRAINT "FK\_Services" FOREIGN KEY ("serviceID")

        REFERENCES public."Services" ("serviceID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_Consumables" FOREIGN KEY ("consumableID")

        REFERENCES public."Consumables" ("consumableID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."ServiceConsumableNeeds"

    OWNER to postgres;

/\*    Создание зависимых таблиц второго порядка   \*/

/\*    Создание зависимых таблиц третьего порядка   \*/

CREATE TABLE public."AttendedServices"

(

    "orderID" integer NOT NULL,

    "serviceID" integer NOT NULL,

    "priceListID" integer NOT NULL,

    "awaitedDateOfAttendance" timestamp(0) without time zone NOT NULL,

    "dateOfCreation" timestamp(0) without time zone NOT NULL,

    "workshopID" integer NOT NULL,

    "consumableUseID" integer NOT NULL,

    PRIMARY KEY ("serviceID", "orderID"),

    CONSTRAINT "FK\_Orders" FOREIGN KEY ("orderID")

        REFERENCES public."Orders" ("orderID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_Services" FOREIGN KEY ("serviceID")

        REFERENCES public."Services" ("serviceID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_PriceLists" FOREIGN KEY ("priceListID")

        REFERENCES public."PriceLists" ("priceListID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_Workshops" FOREIGN KEY ("workshopID")

        REFERENCES public."Workshops" ("workshopID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID,

    CONSTRAINT "FK\_ConsumableUses" FOREIGN KEY ("consumableUseID")

        REFERENCES public."ConsumableUses" ("consumableUseID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."AttendedServices"

    OWNER to postgres;

CREATE TABLE public."BranchPoints"

(

    "branchPointID" integer NOT NULL GENERATED ALWAYS AS IDENTITY,

    "partnerID" integer NOT NULL,

    "address" character varying(255) NOT NULL,

    PRIMARY KEY ("branchPointID"),

    CONSTRAINT "FK\_Partners" FOREIGN KEY ("partnerID")

        REFERENCES public."Partners" ("partnerID") MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

        NOT VALID

);

ALTER TABLE IF EXISTS public."BranchPoints"

    OWNER to postgres;

/\*    Создание зависимых таблиц третьего порядка   \*/