```
create table document (
name text not null,
surname text not null,
lastname text,
id int PRIMARY KEY
)

create table passport (
eye_color int not null,
nationality text not null
) inherits (document)

create table work_permission (
company_id int
) inherits (document)

create table entry_purpose (
id serial primary key,
name text not null
)

create table entry_permission (
date_gained date,
date_expired date,
purpose serial REFERENCES entry_purpose (id)
) inherits (document)


create table declaration (
weight int not null,
size int not null,
```

category_id int

) inherits (document)

```
create table document_pack (
                               id int primary key,
                               passport_id int,
                               entry_permission_id int,
                               work_permission_id int,
                               declaration_id int
)
```

```
create table person (
                        name text not null,
                        surname text not null,
                        lastname text,
                        nationality text not null,
                        eye_color int not null,
                        fingerprint bigint not null,
                        document_pack_id int REFERENCES document_pack (id),
                        luggage_id int,
                        check(eye_color > 0),
                        check(fingerprint > 0)
)
```

create table country (

id serial primary key,

name text not null,

is_borders_open Boolean not null

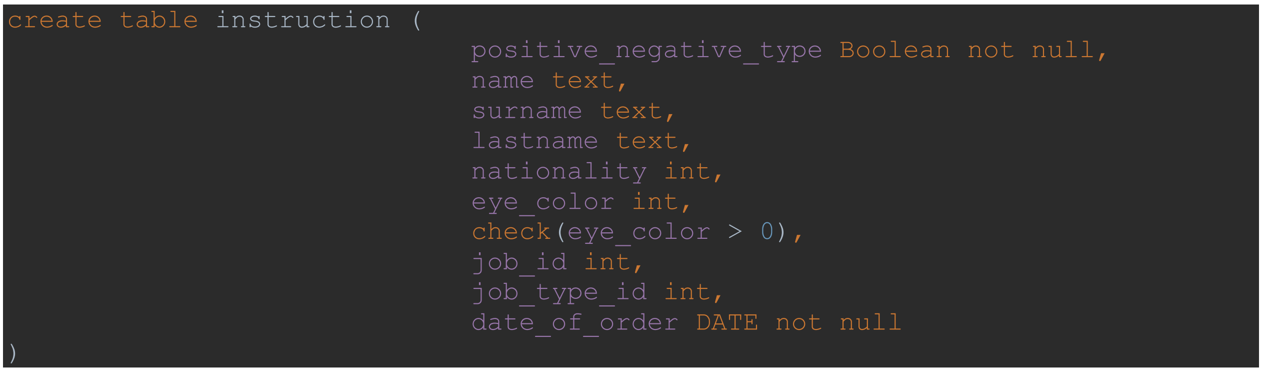
)

create table criminal (

```
          name text,
          surname text,
          lastname text,
          nationality int,
          eye_color int,
          fingerprint bigint,
          check(eye_color > 0),
          check(fingerprint > 0)
```

)

```
create table job_type (

id serial primary key,

name text not null

)


create table company (

id serial primary key,

name text not null

)


create table job (

id serial primary key,

type_id serial REFERENCES job_type (id),

capacity int not null,

company_id serial REFERENCES company (id),

check (capacity > -1)

)
```



```
create table instruction (
                             positive_negative_type Boolean not null,
                             name text,
                             surname text,
                             lastname text,
                             nationality int,
                             eye_color int,
                             check(eye_color > 0),
                             job_id int,
                             job_type_id int,
                             date_of_order DATE not null
)
```

```
create table customs_category (

id serial primary key,

name text not null,

is_importable Boolean not null

)


create table customs_param (
max_weight int,

max_size int,

category_id serial REFERENCES customs_category (id),

check (max_weight > 0),

check (max_size > 0)

)


create table luggage (

id int primary key,

size int NOT NULL,

weight int NOT NULL,

category_id serial REFERENCES customs_category (id)

)


create table entry_list (

id serial primary key,

passport_id int not null

)
```

```
CREATE SEQUENCE "passport_id_seq";
CREATE SEQUENCE "entry_permission_id_seq";
CREATE SEQUENCE "work_permission_id_seq";
CREATE SEQUENCE "luggage_id_seq";
CREATE SEQUENCE "declaration_id_seq";
CREATE SEQUENCE "document_pack_id_seq";
```

```
SELECT next_one('Misha', 'Griboff', null, 'Арстоцка', 1234, 12341234, 'Misha', 'Griboff', null, 1234, 'Арстоцка', 'Misha', 'Griboff', null, CURRENT_DATE, CURRENT_DATE, 1, null, null, null, null, null, null, null, null, null, null, null, null, null)
```

```
SELECT next_one('Misha', 'Griboff', null, 'Ривенделл', 1234, 12341234,
'Misha', 'Griboff', null, 1234, 'Ривенделл', 'Misha', 'Griboff', null,
CURRENT_DATE, CURRENT_DATE, 3, 'Misha', 'Griboff', null, 45, null, null,
null, null, null, null, null, null, null)
```

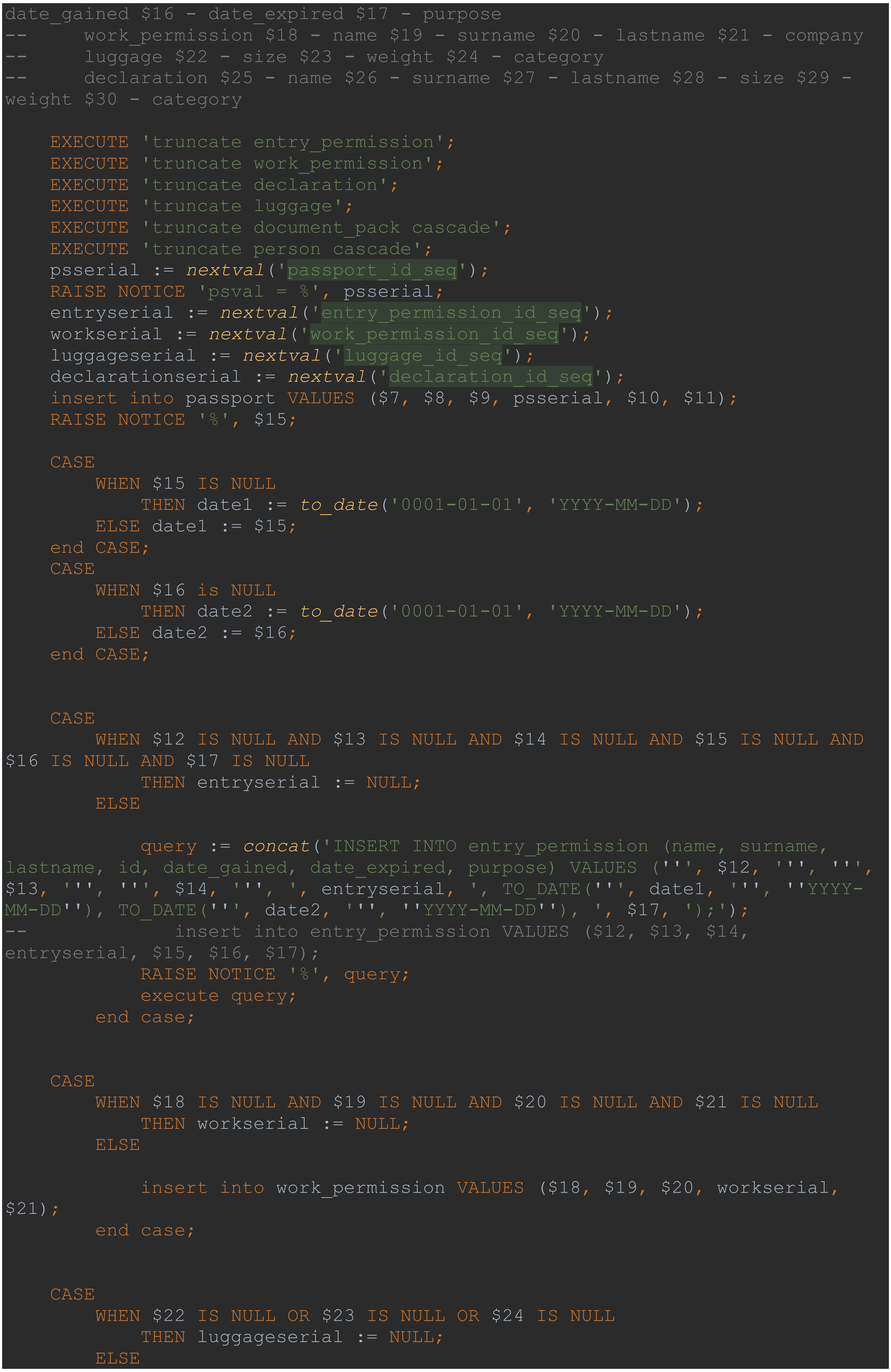
```
SELECT next_one('Misha', 'Griboff', null, 'Калечия', 1234, 12341234, 'Misha',
'Griboff', null, 1234, 'Арстоцка', 'Misha', 'Griboff', null, CURRENT_DATE,
CURRENT_DATE, 1, null, null, null, null, null, null, null, null, null, null,
null, null, null)
```

```
SELECT next_one('Misha', 'Griboff', null, 'Калечия', 1234, 12341234, 'Misha', 'Griboff', null, 1234, 'Калечия', 'Misha', 'Griboff', null, CURRENT_DATE, CURRENT_DATE, 1, null, null, null, null, null, null, null, null, null, null, null, null, null)
```

```
create function passport_check() returns text
    language plpgsql
as
$$
DECLARE
    result_text text;
BEGIN
    SELECT INTO result_text
        CASE
            WHEN EXISTS(SELECT * FROM entry_list WHERE entry_list.passport_id = passport.id)
                THEN 'ACCEPTED'
            ELSE CASE
            WHEN passport.name != person.name OR passport.surname != person.surname OR passport.lastname != person.lastname OR passport.eye_color != person.eye_color OR passport.nationality != person.nationality
                THEN 'DECLINED: MISTAKE IN PASSPORT'
            WHEN NOT EXISTS(select * from country where country.name = passport.nationality)
                THEN 'DECLINED: NO SUCH COUNTRY'
            WHEN NOT (SELECT is_borders_open FROM country WHERE name = passport.nationality)
                THEN 'DECLINED: BORDERS ARE CLOSED NOW'
            WHEN EXISTS(SELECT * FROM instruction
                        JOIN country c2 on instruction.nationality = c2.id
                        WHERE date_of_order = CURRENT_DATE AND NOT positive_negative_type AND (instruction.name = passport.name OR instruction.surname = passport.surname OR instruction.lastname = passport.lastname OR instruction.eye_color = passport.eye_color OR c2.name = passport.nationality))
                THEN 'DECLINED: NO REASON'
            WHEN EXISTS(SELECT * FROM instruction
                                          JOIN country c3 on instruction.nationality = c3.id
                        WHERE date_of_order = CURRENT_DATE AND positive_negative_type AND (instruction.name != passport.name OR instruction.surname != passport.surname OR instruction.lastname != passport.lastname OR instruction.eye_color != passport.eye_color OR c3.name != passport.nationality))
                THEN 'DECLINED: NO REASON'
            WHEN NOT EXISTS(SELECT * FROM instruction WHERE date_of_order = CURRENT_DATE)
                THEN 'ACCEPTED'
            ELSE 'ACCEPTED'
            END
    END
    FROM person
             JOIN document_pack ON person.document_pack_id = document_pack.id
             JOIN passport ON document_pack.passport_id = passport.id;
    RETURN result_text;
END
$$;
```

```
create function entry_check() returns text
    language plpgsql
```

```
as
$$
DECLARE
    result_text text;
BEGIN
    SELECT INTO result_text
        CASE
            WHEN entry_permission.name != person.name OR
entry_permission.surname != person.surname OR entry_permission.lastname !=
person.lastname
                THEN 'DECLINED: MISTAKE IN ENTRY PERMISSION'
            WHEN entry_permission.date_gained = to_date('0001-01-01', 'YYYY-
MM-DD') OR entry_permission.date_expired = to_date('0001-01-01', 'YYYY-MM-
DD')
                THEN 'DECLINED: BLANK DATES'
            WHEN entry_permission.date_gained > CURRENT_DATE
                THEN 'DECLINED: GAINED DATE IS FORGED'
            WHEN entry_permission.date_expired < CURRENT_DATE
                THEN 'DECLINED: PERMISSION EXPIRED'
            WHEN entry_permission.purpose = 3
                THEN CASE
                         WHEN document_pack.work_permission_id IS NULL
                             THEN 'DECLINED: WORK PERMISSION IS ABSENT'
                         ELSE 'ACCEPTED. SHOW YOUR WORK PERMISSION'
                END
            ELSE 'ACCEPTED'
            END
    FROM person
             JOIN document_pack ON person.document_pack_id = document_pack.id
             JOIN entry_permission ON document_pack.entry_permission_id =
entry_permission.id;

    RETURN result_text;
END
$$;


create function next_one(name text, surname text, lastname text, nationality
text, eye_color integer, fingerprint bigint, psname text, pssurname text,
pslastname text, pseye_color integer, psnationality text, enname text,
ensurname text, enlastname text, endate_ganed date, endate_expired date,
enpurpose integer, workname text, worksurname text, worklastname text,
workcompany integer, lugsize integer, lugweight integer, lugcategory integer,
decname text, decsurname text, declastname text, decsize integer, decweight
integer, deccat integer) returns void
    language plpgsql
as
$$
DECLARE
    psserial int;
    entryserial int;
    workserial int;
    luggageserial int;
    declarationserial int;
    docpack int;
    query text;
    date1 date;
    date2 date;
BEGIN
    --     $1 - name $2 - surname $3 - lastname $4 - nationality $5 -
eye_color $6 - fingerprint
--     passport $7 - name $8 - surname $9 - lastname $10 - eye_color $11 -
nationality
--     entry_permission $12 - name $13 - surname $14 - lastname $15 -
```



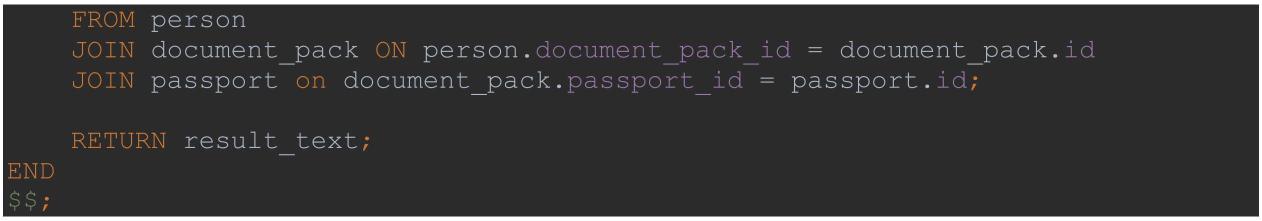
```
date_gained $16 - date_expired $17 - purpose
--      work_permission $18 - name $19 - surname $20 - lastname $21 - company
--      luggage $22 - size $23 - weight $24 - category
--      declaration $25 - name $26 - surname $27 - lastname $28 - size $29 -
weight $30 - category

    EXECUTE 'truncate entry_permission';
    EXECUTE 'truncate work_permission';
    EXECUTE 'truncate declaration';
    EXECUTE 'truncate luggage';
    EXECUTE 'truncate document_pack cascade';
    EXECUTE 'truncate person cascade';
    psserial := nextval('passport_id_seq');
    RAISE NOTICE 'psval = %', psserial;
    entryserial := nextval('entry_permission_id_seq');
    workserial := nextval('work_permission_id_seq');
    luggageserial := nextval('luggage_id_seq');
    declarationserial := nextval('declaration_id_seq');
    insert into passport VALUES ($7, $8, $9, psserial, $10, $11);
    RAISE NOTICE '%', $15;

    CASE
        WHEN $15 IS NULL
            THEN date1 := to_date('0001-01-01', 'YYYY-MM-DD');
        ELSE date1 := $15;
    end CASE;
    CASE
        WHEN $16 is NULL
            THEN date2 := to_date('0001-01-01', 'YYYY-MM-DD');
        ELSE date2 := $16;
    end CASE;


    CASE
        WHEN $12 IS NULL AND $13 IS NULL AND $14 IS NULL AND $15 IS NULL AND
$16 IS NULL AND $17 IS NULL
            THEN entryserial := NULL;
        ELSE

            query := concat('INSERT INTO entry_permission (name, surname,
lastname, id, date_gained, date_expired, purpose) VALUES (''', $12, ''', ''',
$13, ''', ''', $14, ''', ', entryserial, ', TO_DATE(''', date1, ''', ''YYYY-
MM-DD''), TO_DATE(''', date2, ''', ''YYYY-MM-DD''), ', $17, ');');
--              insert into entry_permission VALUES ($12, $13, $14,
entryserial, $15, $16, $17);
            RAISE NOTICE '%', query;
            execute query;
        end case;


    CASE
        WHEN $18 IS NULL AND $19 IS NULL AND $20 IS NULL AND $21 IS NULL
            THEN workserial := NULL;
        ELSE

            insert into work_permission VALUES ($18, $19, $20, workserial,
$21);
        end case;


    CASE
        WHEN $22 IS NULL OR $23 IS NULL OR $24 IS NULL
            THEN luggageserial := NULL;
        ELSE
```

```
            insert into luggage VALUES (luggageserial, $22, $23, $24);
        end case;


    CASE
        WHEN $25 IS NULL AND $26 IS NULL AND $27 IS NULL AND $28 IS NULL AND
$29 IS NULL AND $30 IS NULL
            THEN declarationserial := NULL;
        ELSE

            insert into declaration VALUES ($25, $26, $27, declarationserial,
$28, $29, $30);
        end case;

    docpack := nextval('document_pack_id_seq');
    insert into document_pack VALUES (docpack, psserial, entryserial,
workserial, declarationserial);

    insert into person VALUES ($1, $2, $3, $4, $5, $6, docpack,
luggageserial);
END
$$;



create function work_check() returns text
    language plpgsql
as
$$
DECLARE
    result_text text;

BEGIN
    SELECT INTO result_text
        CASE
            WHEN work_permission.name != person.name OR
work_permission.surname != person.surname OR work_permission.lastname !=
person.lastname
                THEN 'DECLINED: MISTAKE IN WORK PERMISSION'
            WHEN NOT EXISTS(SELECT * FROM company WHERE id =
work_permission.company_id)
                THEN 'DECLINED: NO SUCH COMPANY'
            WHEN NOT EXISTS(SELECT * FROM job WHERE company_id =
work_permission.company_id AND capacity > 0)
                THEN 'DECLINED: THIS COMPANY CANCELLED SEARCHING'
            WHEN (SELECT work_place_reduce(work_permission.company_id))
                then 'ACCEPTED'
            END
    FROM person
             JOIN document_pack ON person.document_pack_id = document_pack.id
             JOIN work_permission ON document_pack.work_permission_id =
work_permission.id;

    RETURN result_text;
END
$$;
```

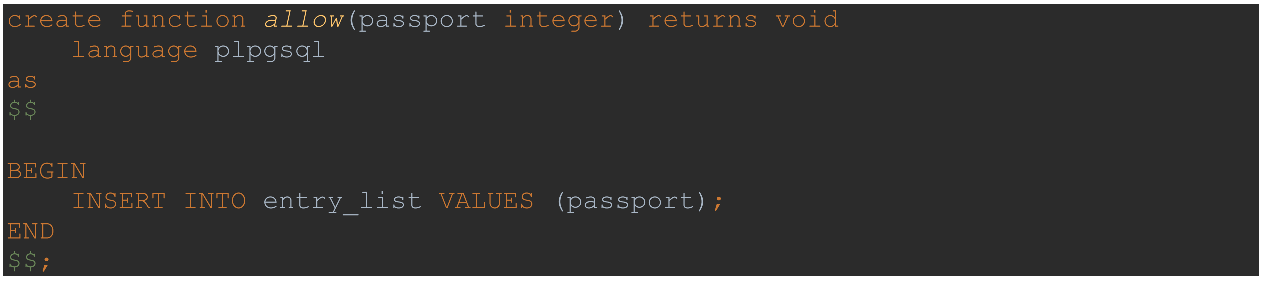
```
create function declaration_check() returns text
    language plpgsql
as
```

```
$$
DECLARE
    result_text text;

BEGIN
    SELECT INTO result_text
        CASE
            WHEN declaration.name != person.name OR declaration.surname !=
person.surname OR declaration.lastname != person.lastname
                THEN 'DECLINED: MISTAKE IN DECLARATION'
            WHEN NOT EXISTS(SELECT * FROM customs_category WHERE id =
declaration.category_id)
                THEN 'DECLINED: NO SUCH CATEGORY'
            WHEN declaration.size > customs_param.max_size OR
declaration.weight > customs_param.max_weight
                THEN 'DECLINED: DECLARATION FORGERY'
            WHEN luggage.size != declaration.size OR luggage.weight !=
declaration.weight OR luggage.category_id != declaration.category_id
                THEN 'DECLINED: WRONG DECLARATION'
            WHEN NOT (SELECT is_importable FROM customs_category WHERE id =
declaration_id)
                THEN 'DECLINED: FORBIDDEN LUGGAGE'
            ELSE
                'ACCEPTED'
            END
    FROM person
             JOIN document_pack ON person.document_pack_id = document_pack.id
             JOIN declaration ON document_pack.declaration_id =
declaration.id
             JOIN luggage ON person.luggage_id = luggage.id
             JOIN customs_param ON customs_param.category_id =
declaration.category_id;

    RETURN result_text;
END
$$;
```

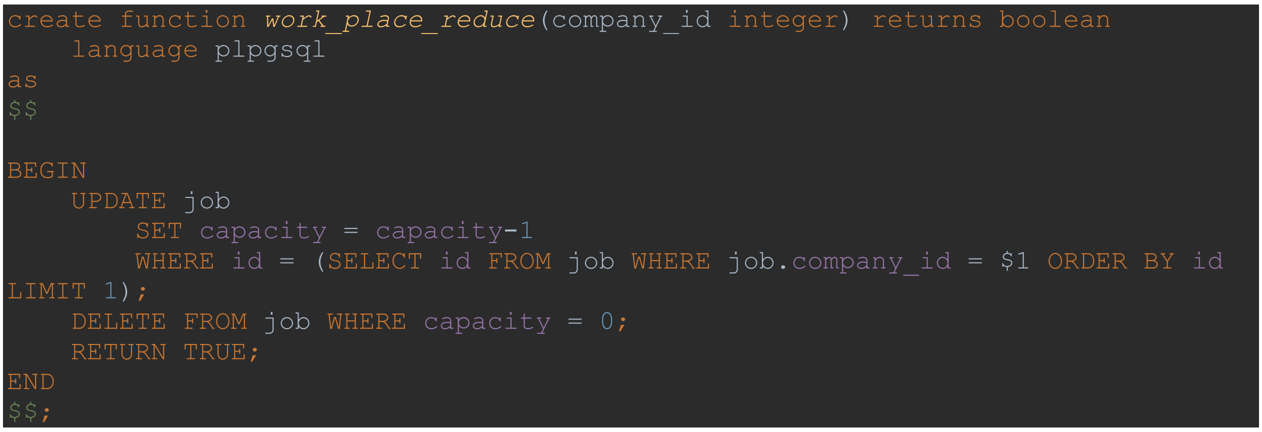
```
create function criminal_check() returns text
    language plpgsql
as
$$
DECLARE
    result_text text;

BEGIN
    SELECT INTO result_text
        CASE
            WHEN EXISTS(SELECT * FROM criminal WHERE criminal.fingerprint =
person.fingerprint)
                THEN 'DECLINED: YOU ARE CRIMINAL'
            WHEN EXISTS(SELECT * FROM criminal WHERE criminal.name =
person.name AND criminal.surname = person.surname AND criminal.lastname =
person.lastname AND criminal.eye_color = person.eye_color)
                THEN 'DECLINED: YOU MIGHT BE A CRIMINAL'
            WHEN EXISTS(SELECT * FROM criminal WHERE criminal.name =
passport.name AND criminal.surname = passport.surname AND criminal.lastname =
passport.lastname AND criminal.eye_color = passport.eye_color)
                THEN 'DECLINED: YOU SHOWED A PASSPORT OF POTENTIAL CRIMINAL'
            ELSE
                'ACCEPTED'
            END
```



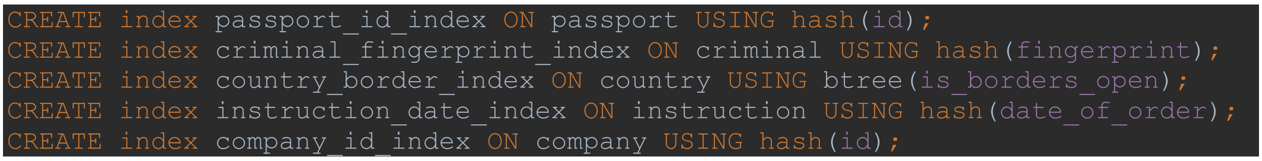
```
    FROM person
    JOIN document_pack ON person.document_pack_id = document_pack.id
    JOIN passport on document_pack.passport_id = passport.id;

    RETURN result_text;
END
$$;
```



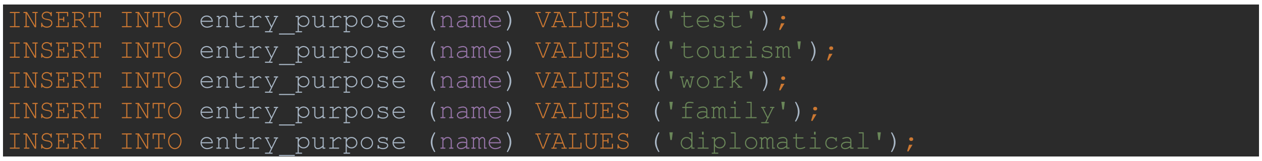
```
create function allow(passport integer) returns void
    language plpgsql
as
$$

BEGIN
    INSERT INTO entry_list VALUES (passport);
END
$$;
```



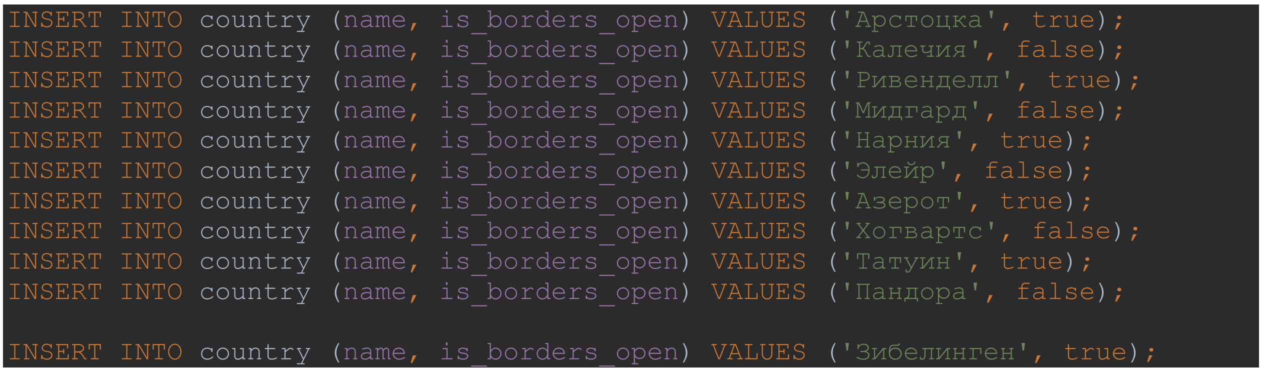
```
create function work_place_reduce(company_id integer) returns boolean
    language plpgsql
as
$$

BEGIN
    UPDATE job
        SET capacity = capacity-1
        WHERE id = (SELECT id FROM job WHERE job.company_id = $1 ORDER BY id
LIMIT 1);
    DELETE FROM job WHERE capacity = 0;
    RETURN TRUE;
END
$$;
```



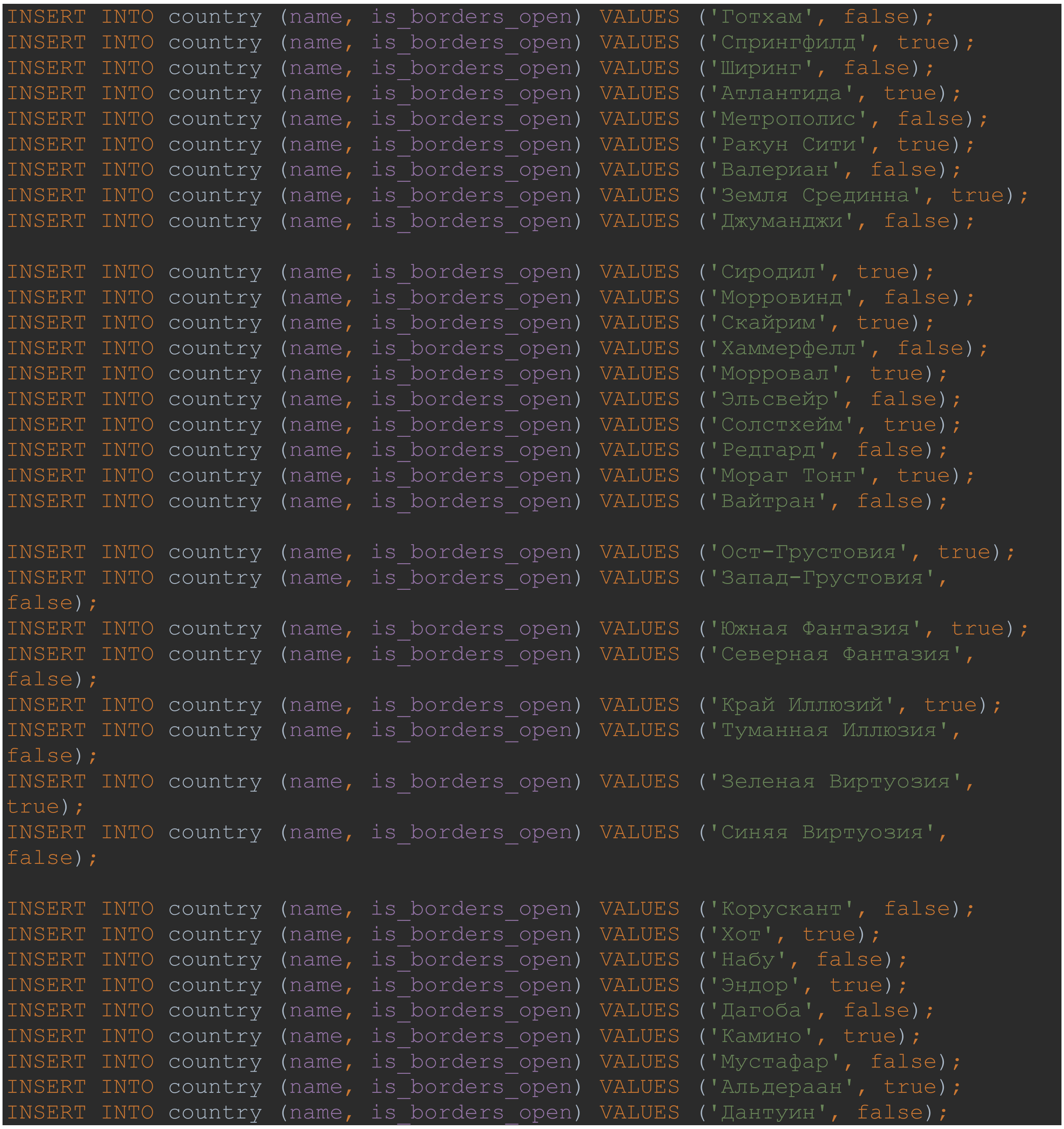
```
CREATE index passport_id_index ON passport USING hash(id);
CREATE index criminal_fingerprint_index ON criminal USING hash(fingerprint);
CREATE index country_border_index ON country USING btree(is_borders_open);
CREATE index instruction_date_index ON instruction USING hash(date_of_order);
CREATE index company_id_index ON company USING hash(id);
```



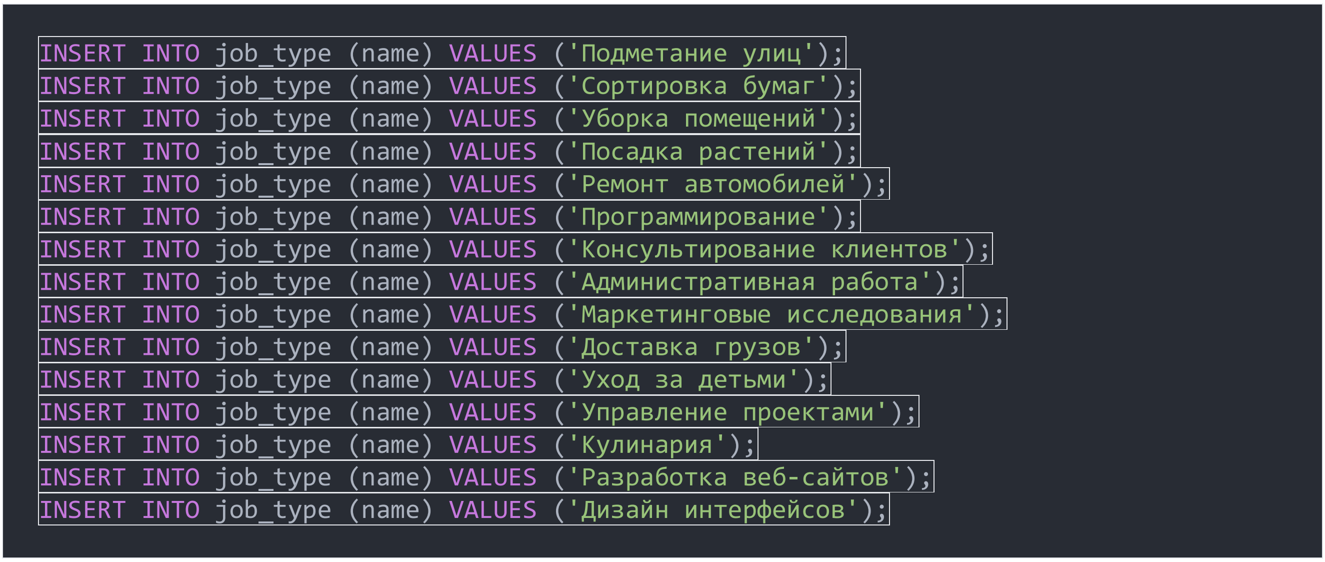
```
INSERT INTO entry_purpose (name) VALUES ('test');
INSERT INTO entry_purpose (name) VALUES ('tourism');
INSERT INTO entry_purpose (name) VALUES ('work');
INSERT INTO entry_purpose (name) VALUES ('family');
INSERT INTO entry_purpose (name) VALUES ('diplomatical');
```



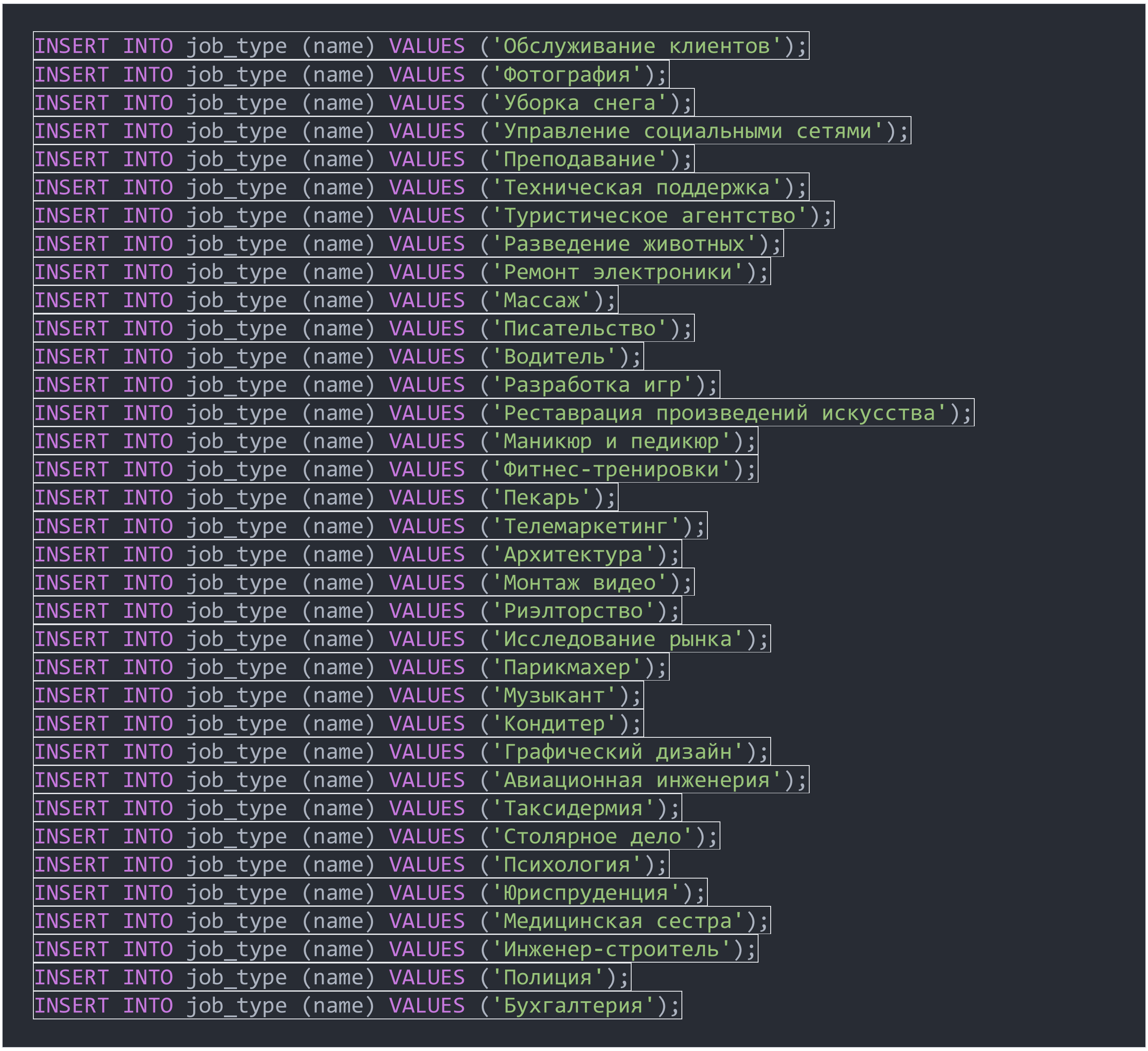
```
INSERT INTO country (name, is_borders_open) VALUES ('Арстоцка', true);
INSERT INTO country (name, is_borders_open) VALUES ('Калечия', false);
INSERT INTO country (name, is_borders_open) VALUES ('Ривенделл', true);
INSERT INTO country (name, is_borders_open) VALUES ('Мидгард', false);
INSERT INTO country (name, is_borders_open) VALUES ('Нарния', true);
INSERT INTO country (name, is_borders_open) VALUES ('Элейр', false);
INSERT INTO country (name, is_borders_open) VALUES ('Азерот', true);
INSERT INTO country (name, is_borders_open) VALUES ('Хогвартс', false);
INSERT INTO country (name, is_borders_open) VALUES ('Татуин', true);
INSERT INTO country (name, is_borders_open) VALUES ('Пандора', false);

INSERT INTO country (name, is_borders_open) VALUES ('Зибелинген', true);
```



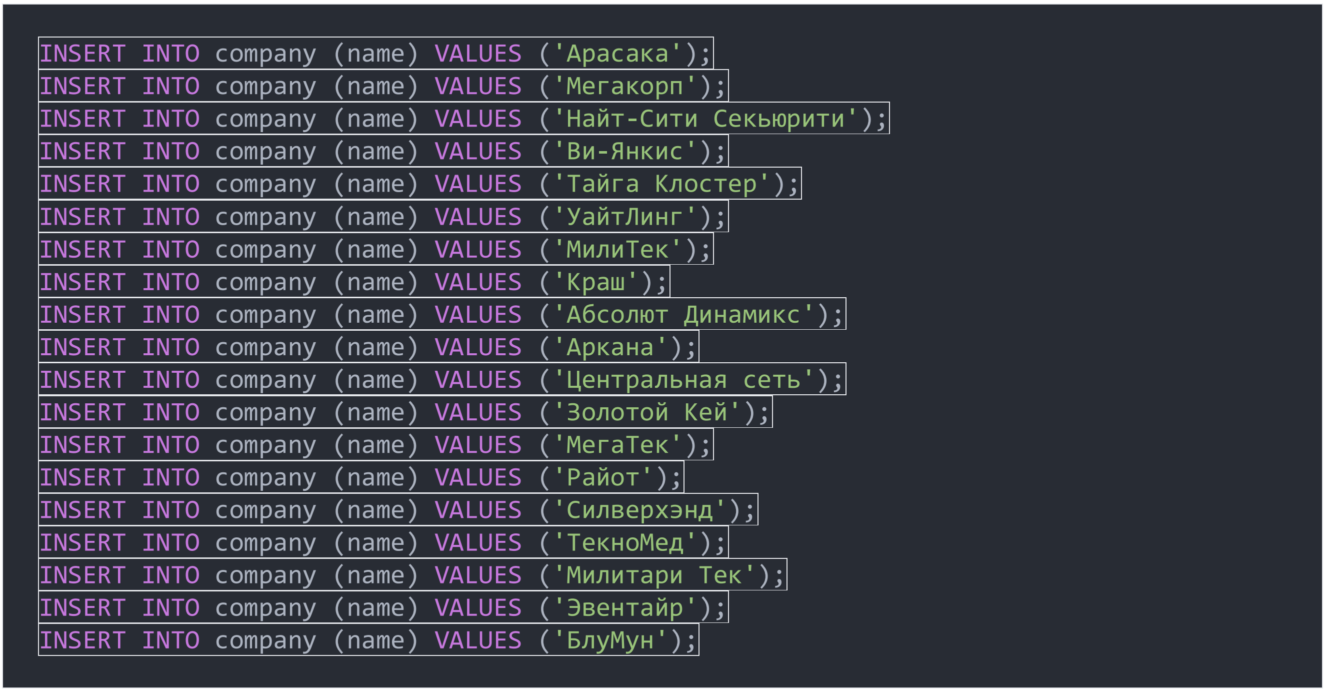
```
INSERT INTO country (name, is_borders_open) VALUES ('Готхам', false);
INSERT INTO country (name, is_borders_open) VALUES ('Спрингфилд', true);
INSERT INTO country (name, is_borders_open) VALUES ('Ширинг', false);
INSERT INTO country (name, is_borders_open) VALUES ('Атлантида', true);
INSERT INTO country (name, is_borders_open) VALUES ('Метрополис', false);
INSERT INTO country (name, is_borders_open) VALUES ('Ракун Сити', true);
INSERT INTO country (name, is_borders_open) VALUES ('Валериан', false);
INSERT INTO country (name, is_borders_open) VALUES ('Земля Срединна', true);
INSERT INTO country (name, is_borders_open) VALUES ('Джуманджи', false);

INSERT INTO country (name, is_borders_open) VALUES ('Сиродил', true);
INSERT INTO country (name, is_borders_open) VALUES ('Морровинд', false);
INSERT INTO country (name, is_borders_open) VALUES ('Скайрим', true);
INSERT INTO country (name, is_borders_open) VALUES ('Хаммерфелл', false);
INSERT INTO country (name, is_borders_open) VALUES ('Морровал', true);
INSERT INTO country (name, is_borders_open) VALUES ('Эльсвейр', false);
INSERT INTO country (name, is_borders_open) VALUES ('Солстхейм', true);
INSERT INTO country (name, is_borders_open) VALUES ('Редгард', false);
INSERT INTO country (name, is_borders_open) VALUES ('Мораг Тонг', true);
INSERT INTO country (name, is_borders_open) VALUES ('Вайтран', false);

INSERT INTO country (name, is_borders_open) VALUES ('Ост-Грустовия', true);
INSERT INTO country (name, is_borders_open) VALUES ('Запад-Грустовия',
false);
INSERT INTO country (name, is_borders_open) VALUES ('Южная Фантазия', true);
INSERT INTO country (name, is_borders_open) VALUES ('Северная Фантазия',
false);
INSERT INTO country (name, is_borders_open) VALUES ('Край Иллюзий', true);
INSERT INTO country (name, is_borders_open) VALUES ('Туманная Иллюзия',
false);
INSERT INTO country (name, is_borders_open) VALUES ('Зеленая Виртуозия',
true);
INSERT INTO country (name, is_borders_open) VALUES ('Синяя Виртуозия',
false);

INSERT INTO country (name, is_borders_open) VALUES ('Корускант', false);
INSERT INTO country (name, is_borders_open) VALUES ('Хот', true);
INSERT INTO country (name, is_borders_open) VALUES ('Набу', false);
INSERT INTO country (name, is_borders_open) VALUES ('Эндор', true);
INSERT INTO country (name, is_borders_open) VALUES ('Дагоба', false);
INSERT INTO country (name, is_borders_open) VALUES ('Камино', true);
INSERT INTO country (name, is_borders_open) VALUES ('Мустафар', false);
INSERT INTO country (name, is_borders_open) VALUES ('Альдераан', true);
INSERT INTO country (name, is_borders_open) VALUES ('Дантуин', false);
```



```
INSERT INTO job_type (name) VALUES ('Подметание улиц');
INSERT INTO job_type (name) VALUES ('Сортировка бумаг');
INSERT INTO job_type (name) VALUES ('Уборка помещений');
INSERT INTO job_type (name) VALUES ('Посадка растений');
INSERT INTO job_type (name) VALUES ('Ремонт автомобилей');
INSERT INTO job_type (name) VALUES ('Программирование');
INSERT INTO job_type (name) VALUES ('Консультирование клиентов');
INSERT INTO job_type (name) VALUES ('Административная работа');
INSERT INTO job_type (name) VALUES ('Маркетинговые исследования');
INSERT INTO job_type (name) VALUES ('Доставка грузов');
INSERT INTO job_type (name) VALUES ('Уход за детьми');
INSERT INTO job_type (name) VALUES ('Управление проектами');
INSERT INTO job_type (name) VALUES ('Кулинария');
INSERT INTO job_type (name) VALUES ('Разработка веб-сайтов');
INSERT INTO job_type (name) VALUES ('Дизайн интерфейсов');
```



```
INSERT INTO job_type (name) VALUES ('Обслуживание клиентов');
INSERT INTO job_type (name) VALUES ('Фотография');
INSERT INTO job_type (name) VALUES ('Уборка снега');
INSERT INTO job_type (name) VALUES ('Управление социальными сетями');
INSERT INTO job_type (name) VALUES ('Преподавание');
INSERT INTO job_type (name) VALUES ('Техническая поддержка');
INSERT INTO job_type (name) VALUES ('Туристическое агентство');
INSERT INTO job_type (name) VALUES ('Разведение животных');
INSERT INTO job_type (name) VALUES ('Ремонт электроники');
INSERT INTO job_type (name) VALUES ('Массаж');
INSERT INTO job_type (name) VALUES ('Писательство');
INSERT INTO job_type (name) VALUES ('Водитель');
INSERT INTO job_type (name) VALUES ('Разработка игр');
INSERT INTO job_type (name) VALUES ('Реставрация произведений искусства');
INSERT INTO job_type (name) VALUES ('Маникюр и педикюр');
INSERT INTO job_type (name) VALUES ('Фитнес-тренировки');
INSERT INTO job_type (name) VALUES ('Пекарь');
INSERT INTO job_type (name) VALUES ('Телемаркетинг');
INSERT INTO job_type (name) VALUES ('Архитектура');
INSERT INTO job_type (name) VALUES ('Монтаж видео');
INSERT INTO job_type (name) VALUES ('Риэлторство');
INSERT INTO job_type (name) VALUES ('Исследование рынка');
INSERT INTO job_type (name) VALUES ('Парикмахер');
INSERT INTO job_type (name) VALUES ('Музыкант');
INSERT INTO job_type (name) VALUES ('Кондитер');
INSERT INTO job_type (name) VALUES ('Графический дизайн');
INSERT INTO job_type (name) VALUES ('Авиационная инженерия');
INSERT INTO job_type (name) VALUES ('Таксидермия');
INSERT INTO job_type (name) VALUES ('Столярное дело');
INSERT INTO job_type (name) VALUES ('Психология');
INSERT INTO job_type (name) VALUES ('Юриспруденция');
INSERT INTO job_type (name) VALUES ('Медицинская сестра');
INSERT INTO job_type (name) VALUES ('Инженер-строитель');
INSERT INTO job_type (name) VALUES ('Полиция');
INSERT INTO job_type (name) VALUES ('Бухгалтерия');
```



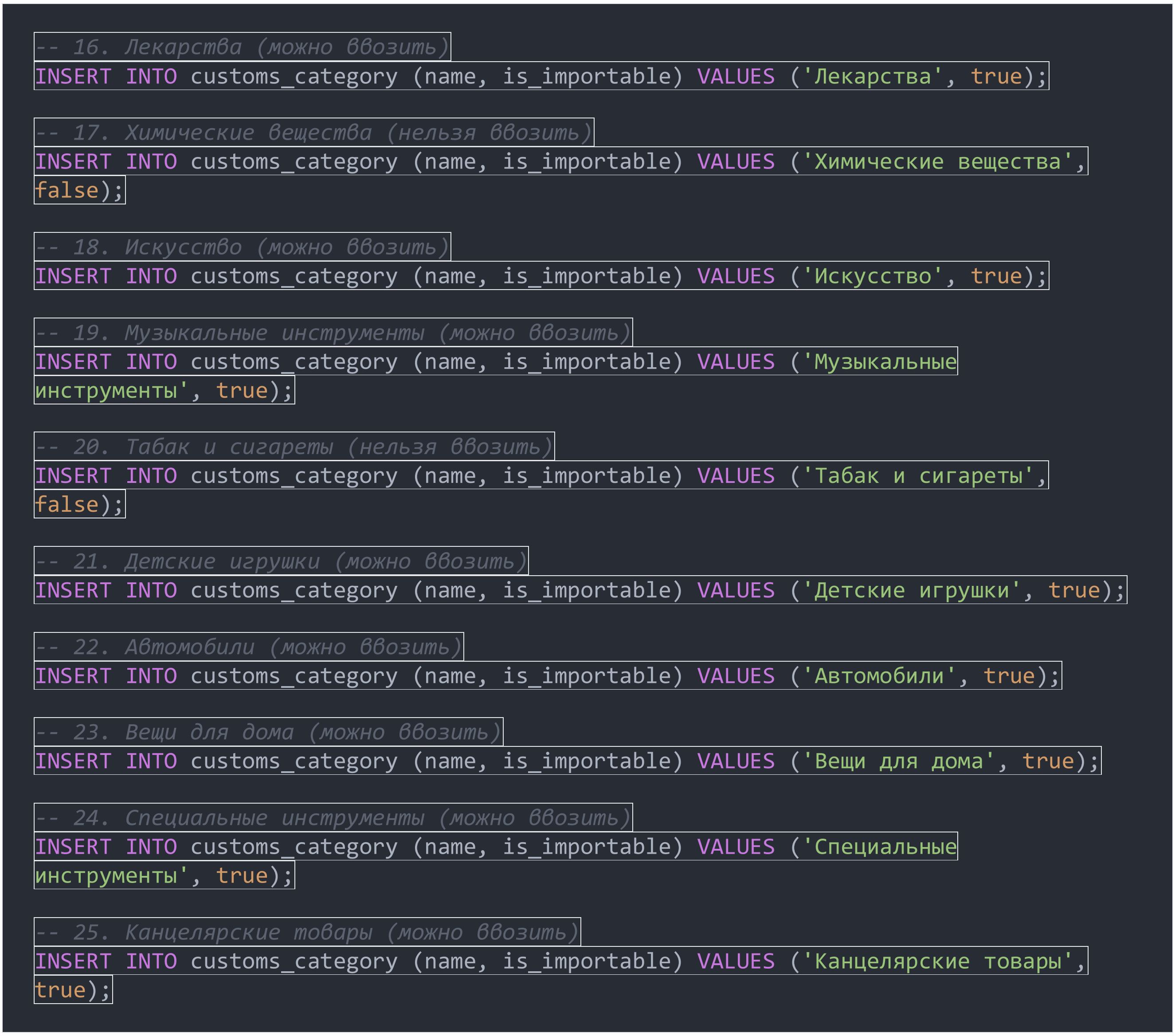
```
INSERT INTO company (name) VALUES ('Арасака');
INSERT INTO company (name) VALUES ('Мегакорп');
INSERT INTO company (name) VALUES ('Найт-Сити Секьюрити');
INSERT INTO company (name) VALUES ('Ви-Янкис');
INSERT INTO company (name) VALUES ('Тайга Клостер');
INSERT INTO company (name) VALUES ('УайтЛинг');
INSERT INTO company (name) VALUES ('МилиТек');
INSERT INTO company (name) VALUES ('Краш');
INSERT INTO company (name) VALUES ('Абсолют Динамикс');
INSERT INTO company (name) VALUES ('Аркана');
INSERT INTO company (name) VALUES ('Центральная сеть');
INSERT INTO company (name) VALUES ('Золотой Кей');
INSERT INTO company (name) VALUES ('МегаТек');
INSERT INTO company (name) VALUES ('Райот');
INSERT INTO company (name) VALUES ('Силверхэнд');
INSERT INTO company (name) VALUES ('ТекноМед');
INSERT INTO company (name) VALUES ('Милитари Тек');
INSERT INTO company (name) VALUES ('Эвентайр');
INSERT INTO company (name) VALUES ('БлуМун');
INSERT INTO company (name) VALUES ('Кристал Динамикс');
INSERT INTO company (name) VALUES ('Армет');
INSERT INTO company (name) VALUES ('Канон Тек');
INSERT INTO company (name) VALUES ('Синтек');
INSERT INTO company (name) VALUES ('Райз');
INSERT INTO company (name) VALUES ('МедТек');
INSERT INTO company (name) VALUES ('Стилето');
INSERT INTO company (name) VALUES ('Хроматика');
INSERT INTO company (name) VALUES ('Самурай');
INSERT INTO company (name) VALUES ('Тайга Концерн');
INSERT INTO company (name) VALUES ('Киберсеть');
INSERT INTO company (name) VALUES ('Армалибер');
INSERT INTO company (name) VALUES ('Найт Корп');
INSERT INTO company (name) VALUES ('Альтернатива');
INSERT INTO company (name) VALUES ('Компьютер Разум');
INSERT INTO company (name) VALUES ('Тайга Секьюрити');
INSERT INTO company (name) VALUES ('Вудманс');
INSERT INTO company (name) VALUES ('Мегафарм');
INSERT INTO company (name) VALUES ('Кибердайн');
INSERT INTO company (name) VALUES ('Ассоциация Киберпанка');
INSERT INTO company (name) VALUES ('Блэйдз');
INSERT INTO company (name) VALUES ('Парагон');
INSERT INTO company (name) VALUES ('Кибервант');
INSERT INTO company (name) VALUES ('Центральная Секьюрити');
INSERT INTO company (name) VALUES ('Армори');
INSERT INTO company (name) VALUES ('МедКорп');
INSERT INTO company (name) VALUES ('МегаФуд');
INSERT INTO company (name) VALUES ('КиберСпорт');
INSERT INTO company (name) VALUES ('Тайга Корп');
INSERT INTO company (name) VALUES ('Квант');
INSERT INTO company (name) VALUES ('КиберНет');
INSERT INTO company (name) VALUES ('Астрал');
```

```
INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 45);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 2);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 15);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 12);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 30);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 28);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 7);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 22);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 3);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 39);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 14);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 18);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 36);
INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 33);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 9);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 47);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 41);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 4);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 8);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 26);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 19);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 25);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 10);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 48);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 11);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 29);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 42);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 40);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 5);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 20);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 37);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 13);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 17);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 38);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 23);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 49);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 34);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 21);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 31);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 6);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 16);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 24);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 27);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 1);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 32);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 1, 35);

INSERT INTO job (type_id, capacity, company_id) VALUES (1, 3, 46);
INSERT INTO job (type_id, capacity, company_id) VALUES (1, 2, 50);
INSERT INTO job (type_id, capacity, company_id) VALUES (1, 4, 43);
INSERT INTO job (type_id, capacity, company_id) VALUES (1, 5, 44);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 20);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 9);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 3);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 49);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 7);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 1, 11);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 34);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 17);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 30);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 19);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 1, 40);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 14);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 48);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 5);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 28);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 1, 23);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 39);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 12);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 41);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 2);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 1, 27);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 13);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 10);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 36);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 8);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 1, 6);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 15);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 16);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 29);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 5, 43);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 1, 25);
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 3, 37);
```

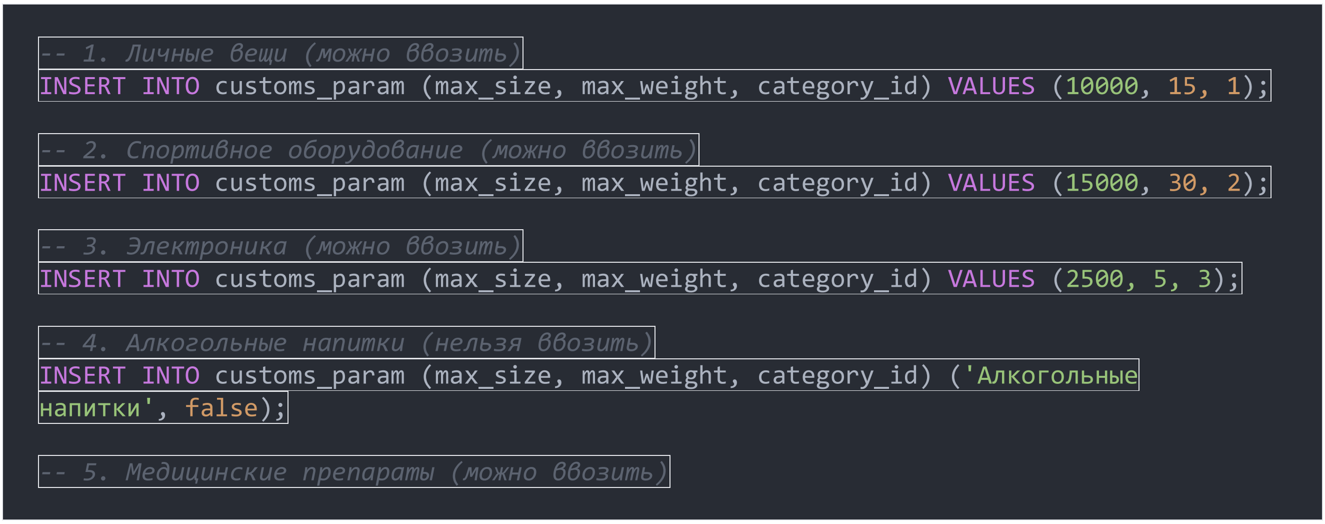
```
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 2, 22);
```

```
INSERT INTO job (type_id, capacity, company_id) VALUES (2, 4, 35);
```

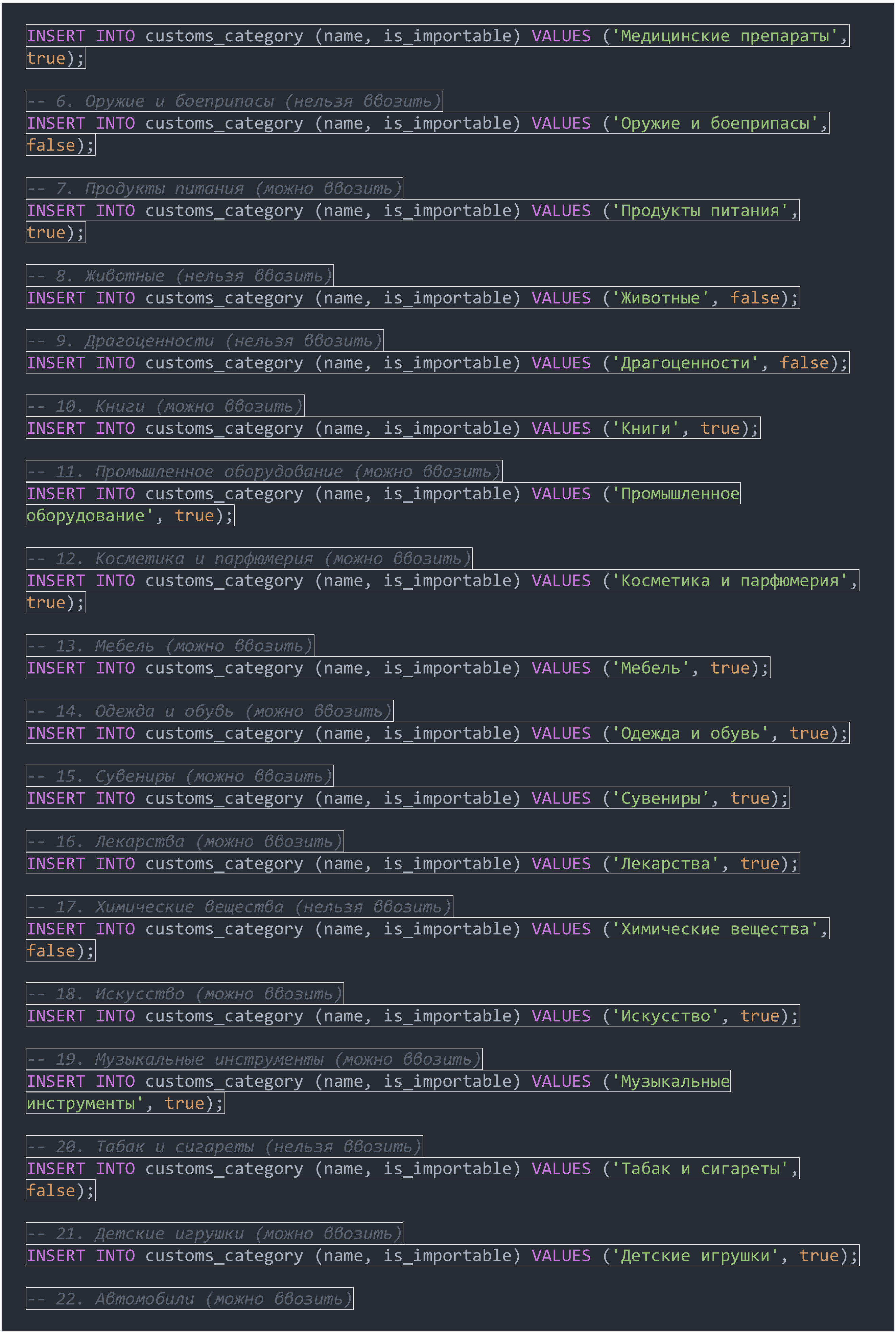
```
-- 1. Личные вещи (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Личные вещи', true);

-- 2. Спортивное оборудование (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Спортивное
оборудование', true);

-- 3. Электроника (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Электроника', true);

-- 4. Алкогольные напитки (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Алкогольные напитки',
false);

-- 5. Медицинские препараты (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Медицинские препараты',
true);

-- 6. Оружие и боеприпасы (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Оружие и боеприпасы',
false);

-- 7. Продукты питания (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Продукты питания',
true);

-- 8. Животные (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Животные', false);

-- 9. Драгоценности (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Драгоценности', false);

-- 10. Книги (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Книги', true);

-- 11. Промышленное оборудование (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Промышленное
оборудование', true);

-- 12. Косметика и парфюмерия (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Косметика и парфюмерия',
true);

-- 13. Мебель (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Мебель', true);

-- 14. Одежда и обувь (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Одежда и обувь', true);

-- 15. Сувениры (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Сувениры', true);
```



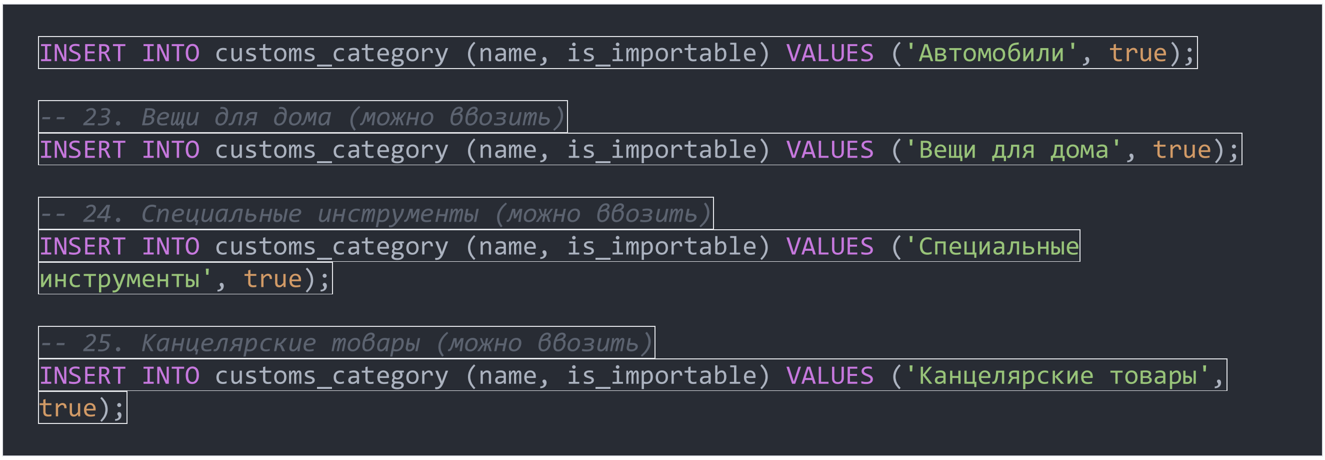
```sql
-- 16. Лекарства (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Лекарства', true);

-- 17. Химические вещества (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Химические вещества', false);

-- 18. Искусство (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Искусство', true);

-- 19. Музыкальные инструменты (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Музыкальные инструменты', true);

-- 20. Табак и сигареты (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Табак и сигареты', false);

-- 21. Детские игрушки (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Детские игрушки', true);

-- 22. Автомобили (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Автомобили', true);

-- 23. Вещи для дома (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Вещи для дома', true);

-- 24. Специальные инструменты (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Специальные инструменты', true);

-- 25. Канцелярские товары (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Канцелярские товары', true);
```



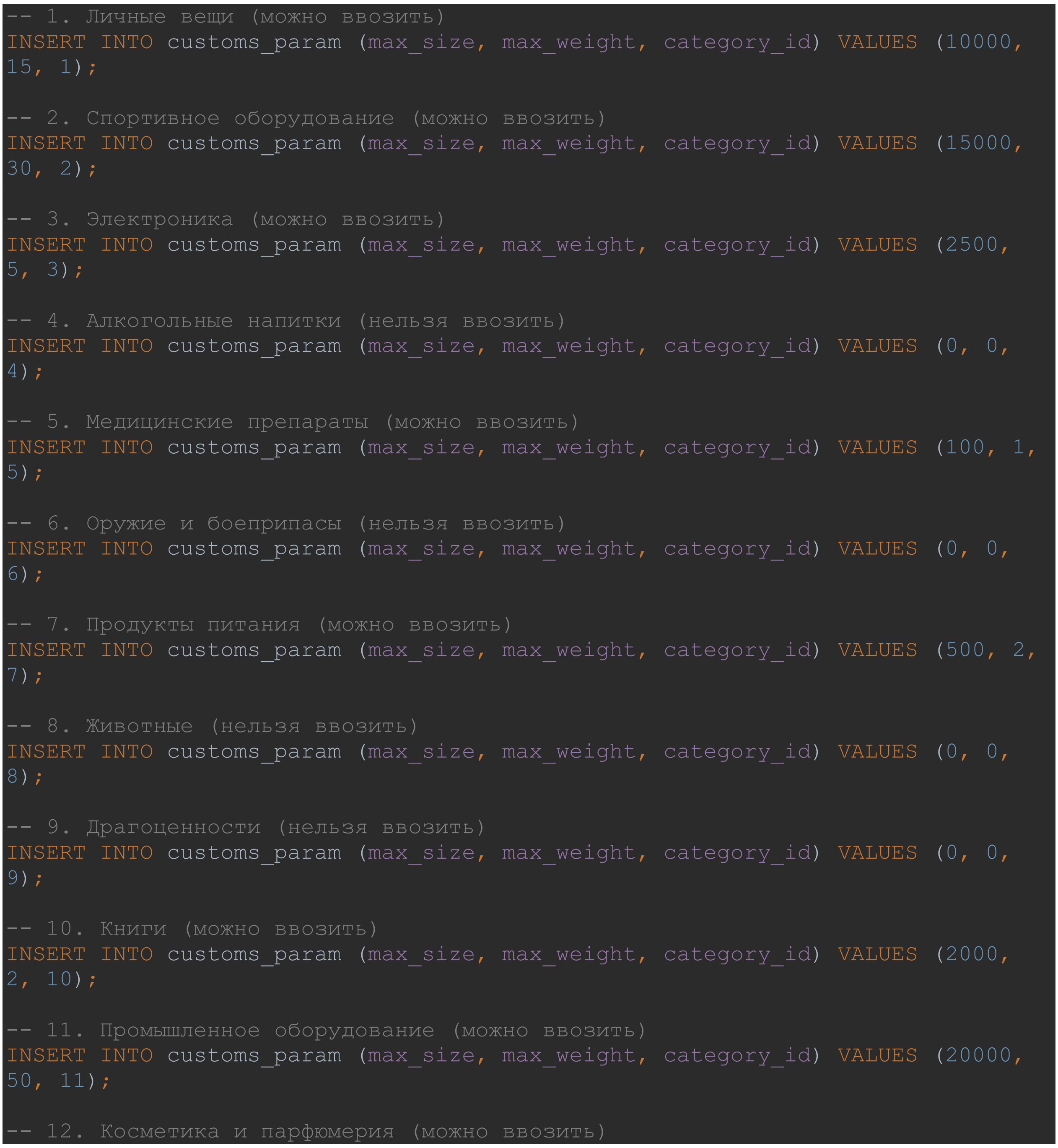
```sql
-- 1. Личные вещи (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (10000, 15, 1);

-- 2. Спортивное оборудование (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (15000, 30, 2);

-- 3. Электроника (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (2500, 5, 3);

-- 4. Алкогольные напитки (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) ('Алкогольные напитки', false);

-- 5. Медицинские препараты (можно ввозить)
```



```sql
INSERT INTO customs_category (name, is_importable) VALUES ('Медицинские препараты',
true);

-- 6. Оружие и боеприпасы (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Оружие и боеприпасы',
false);

-- 7. Продукты питания (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Продукты питания',
true);

-- 8. Животные (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Животные', false);

-- 9. Драгоценности (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Драгоценности', false);

-- 10. Книги (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Книги', true);

-- 11. Промышленное оборудование (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Промышленное
оборудование', true);

-- 12. Косметика и парфюмерия (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Косметика и парфюмерия',
true);

-- 13. Мебель (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Мебель', true);

-- 14. Одежда и обувь (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Одежда и обувь', true);

-- 15. Сувениры (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Сувениры', true);

-- 16. Лекарства (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Лекарства', true);

-- 17. Химические вещества (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Химические вещества',
false);

-- 18. Искусство (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Искусство', true);

-- 19. Музыкальные инструменты (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Музыкальные
инструменты', true);

-- 20. Табак и сигареты (нельзя ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Табак и сигареты',
false);

-- 21. Детские игрушки (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Детские игрушки', true);

-- 22. Автомобили (можно ввозить)
```



```
INSERT INTO customs_category (name, is_importable) VALUES ('Автомобили', true);

-- 23. Вещи для дома (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Вещи для дома', true);

-- 24. Специальные инструменты (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Специальные
инструменты', true);

-- 25. Канцелярские товары (можно ввозить)
INSERT INTO customs_category (name, is_importable) VALUES ('Канцелярские товары',
true);
```



```
-- 1. Личные вещи (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (10000,
15, 1);

-- 2. Спортивное оборудование (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (15000,
30, 2);

-- 3. Электроника (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (2500,
5, 3);

-- 4. Алкогольные напитки (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (0, 0,
4);

-- 5. Медицинские препараты (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (100, 1,
5);

-- 6. Оружие и боеприпасы (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (0, 0,
6);

-- 7. Продукты питания (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (500, 2,
7);

-- 8. Животные (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (0, 0,
8);

-- 9. Драгоценности (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (0, 0,
9);

-- 10. Книги (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (2000,
2, 10);

-- 11. Промышленное оборудование (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (20000,
50, 11);

-- 12. Косметика и парфюмерия (можно ввозить)
```

```
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (100, 1,
12);

-- 13. Мебель (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (30000,
100, 13);

-- 14. Одежда и обувь (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (2000,
2, 14);

-- 15. Сувениры (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (200, 1,
15);

-- 16. Лекарства (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (100, 1,
16);

-- 17. Химические вещества (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (0, 0,
17);

-- 18. Искусство (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (10000,
5, 18);

-- 19. Музыкальные инструменты (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (15000,
10, 19);

-- 20. Табак и сигареты (нельзя ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (0, 0,
20);

-- 21. Детские игрушки (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (2000,
2, 21);

-- 22. Автомобили (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES
(8000000, 3000, 22);

-- 23. Вещи для дома (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (8000,
8, 23);

-- 24. Специальные инструменты (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (2000,
20, 24);

-- 25. Канцелярские товары (можно ввозить)
INSERT INTO customs_param (max_size, max_weight, category_id) VALUES (500, 1,
25);
```

```
create function next_one1(name text, surname text, lastname text, nationality
text, eye_color integer, fingerprint bigint, psname text, pssurname text,
pslastname text, pseye_color integer, psnationality text, enname text,
ensurname text, enlastname text, endate_ganed date, endate_expired date,
enpurpose integer, workname text, worksurname text, worklastname text,
workcompany integer, lugsize integer, lugweight integer, lugcategory integer,
decname text, decsurname text, declastname text, decsize integer, decweight
integer, deccat integer) returns void
    language plpgsql
as
$$
DECLARE
    psserial int;
    entryserial int;
    workserial int;
    luggageserial int;
    declarationserial int;
    docpack int;
BEGIN
    --     $1 - name $2 - surname $3 - lastname $4 - nationality $5 -
eye_color $6 - fingerprint
--     passport $7 - name $8 - surname $9 - lastname $10 - eye_color $11 -
nationality
--     entry_permission $12 - name $13 - surname $14 - lastname $15 -
date_gained $16 - date_expired $17 - purpose
--     work_permission $18 - name $19 - surname $20 - lastname $21 - company
--     luggage $22 - size $23 - weight $24 - category
--     declaration $25 - name $26 - surname $27 - lastname $28 - size $29 -
weight $30 - category

    drop table person;
    drop table entry_permission;
    drop table work_permission;
    drop table declaration;
    drop table luggage;
    psserial := nextval('passport_id_seq');
    entryserial := nextval('entry_permission_id_seq');
    workserial := nextval('work_permission_id_seq');
    luggageserial := nextval('luggage_id_seq');
    declarationserial := nextval('declaration_id_seq');
    insert into passport VALUES ($7, $8, $9, psserial, $10, $11);

    CASE
        WHEN $12 IS NULL AND $13 IS NULL AND $14 IS NULL AND $15 IS NULL AND
$16 IS NULL AND $17 IS NULL
            THEN entryserial := NULL;
        ELSE

            insert into entry_permission VALUES ($12, $13, $14, entryserial,
$15, $16, $17);
        end case;


    CASE
        WHEN $18 IS NULL AND $19 IS NULL AND $20 IS NULL AND $21 IS NULL
            THEN workserial := NULL;
        ELSE

            insert into work_permission VALUES ($18, $19, $20, workserial,
$21);
        end case;


    CASE
        WHEN $22 IS NULL OR $23 IS NULL OR $24 IS NULL
            THEN luggageserial := NULL;
        ELSE

            insert into luggage VALUES (luggageserial, $22, $23, $24);
        end case;
    CASE
        WHEN $25 IS NULL AND $26 IS NULL AND $27 IS NULL AND $28 IS NULL AND
$29 IS NULL AND $30 IS NULL
            THEN declarationserial := NULL;
        ELSE

            insert into declaration VALUES ($25, $26, $27, declarationserial,
$28, $29, $30);
        end case;

    docpack := nextval('document_pack_id_seq');
    insert into document_pack VALUES (docpack, psserial, entryserial,
workserial, declarationserial);

    insert into person VALUES ($1, $2, $3, $4, $5, $6, docpack,
luggageserial);
END
$$;
```