

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень,
звание

подпись, дата

Н.В Путилова

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы

по дисциплине: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. _____ 4134к

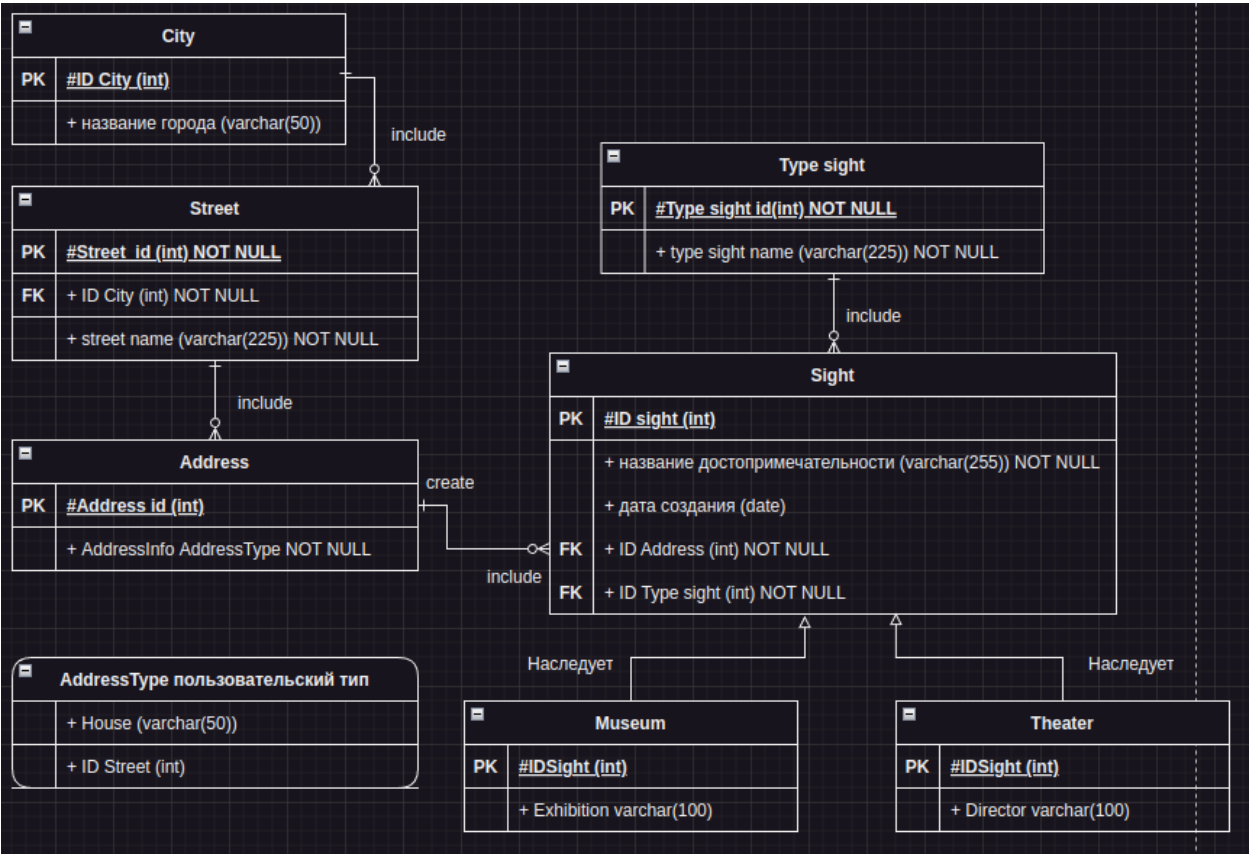
подпись, дата

Столяров Н.С.

инициалы, фамилия

Задание:

- 1)Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 9.
- 2)Придумать (задание) и реализовать запрос выборки с условием к таблицам и предку и потомку.
- Придумать и реализовать запросы выборки с условием к таблице только к предку и только к потомку.
- 3)Придумать и создать пользовательский оператор для своей предметной области
- 4)Придумать и создать пользовательскую агрегатную функцию для своей предметной области



Листинг

```
-- выборка с условием
-- Выборка всех музеев и театров с их адресами
```

```
SELECT
    S.NameSight AS SightName,
    S.CreateDate AS CreateDate,
    A.AddressInfo AS Address,
    TS.TypeSightName AS Type
FROM
    Sight S
JOIN
    Address A ON S.IDAddress = A.IDAddress
JOIN
    TypeSight TS ON S.IDTypeSight = TS.IDTypeSight;
```

```
-- запрос выборки предку
-- Выборка всех объектов Sight
```

```
SELECT
    IDSight,
    NameSight,
    CreateDate
FROM
    Sight;
```

```
SELECT
    T.IDSight,
    T.NameSight,
    T.CreateDate,
```

```
NULL AS Exhibition, -- Для театров выставка не применима
T.Director,
'Theater' AS Type -- Указываем тип как 'Theater'
FROM
Theater T;
```

```
-- Запрос выборки только к предку (Sight)
-- Выборка всех объектов Sight
```

```
SELECT
    IDSight,
    NameSight,
    CreateDate
FROM
    Sight;
```

```
-- Запрос выборки только к потомку (Museum)
-- Выборка всех музеев
```

```
SELECT
    M.IDSight,
    M.NameSight,
    M.CreateDate,
    M.Exhibition
FROM
    Museum M;
```

-- оператор для сравнения названий достопримечательностей.

-- Создание пользовательского оператора

CREATE OR REPLACE FUNCTION sight_name_compare(a VARCHAR, b VARCHAR) RETURNS
BOOLEAN AS \$\$

BEGIN

 RETURN a = b;

END;

\$\$ LANGUAGE plpgsql;

CREATE OPERATOR <=> (

 LEFTARG = VARCHAR,

 RIGHTARG = VARCHAR,

 PROCEDURE = sight_name_compare

);

SELECT * ,sight.NameSight <=> 'Лувр' from sight

-- Создание пользовательской агрегатной функции

CREATE AGGREGATE COUNT_AND_LIST_MUSEUMS (

 SFUNC = count_and_list_museums_sfunc,

 STYPE = TEXT,

 INITCOND = "

);

```
CREATE OR REPLACE FUNCTION count_and_list_museums_sfunc(state TEXT, sight_name  
VARCHAR) RETURNS TEXT AS $$
```

```
BEGIN
```

```
    IF state = '' THEN
```

```
        RETURN sight_name; -- Если это первый элемент, просто возвращаем его
```

```
    ELSE
```

```
        RETURN state || ', ' || sight_name; -- Иначе добавляем к существующему  
состоянию
```

```
    END IF;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
-- Пример использования агрегатной функции
```

```
SELECT COUNT_AND_LIST_MUSEUMS(NameSight) AS MuseumsList FROM Museum;
```

```
-----
```