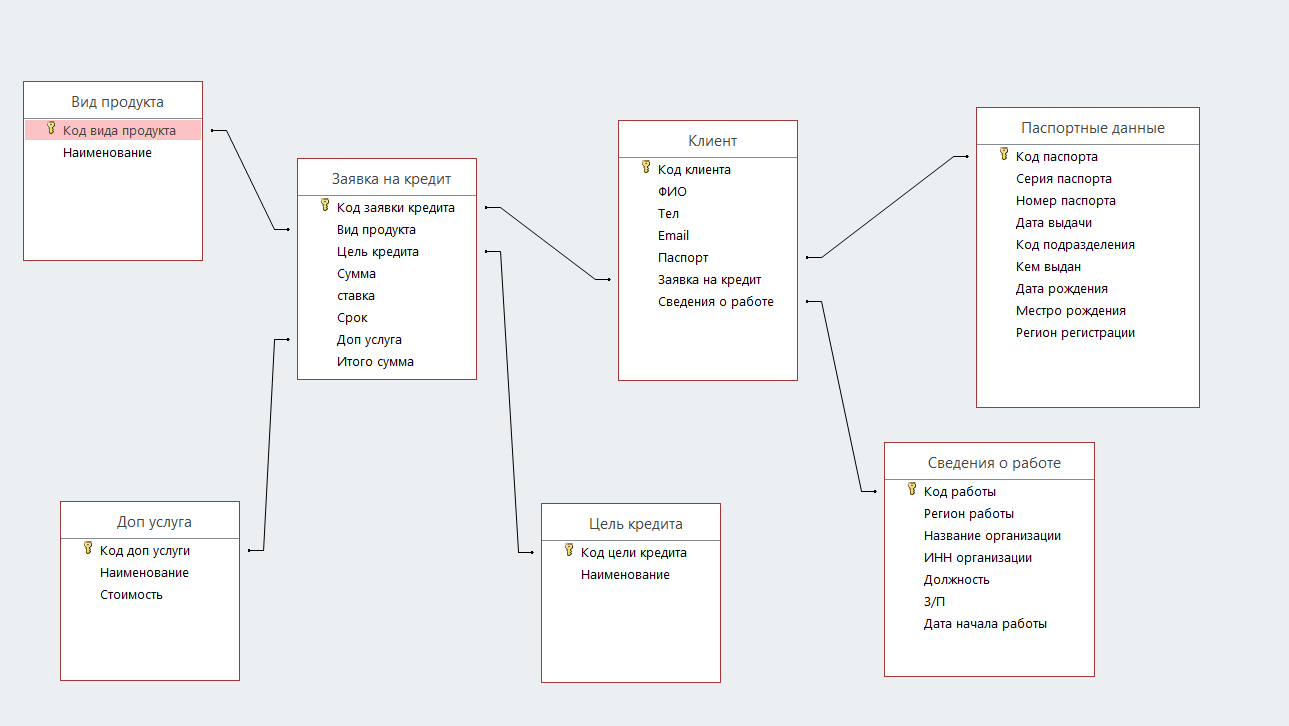
**Решение**

**Задание 1**

Опишите модель данных, которая может лежать в основе данного приложения.

Диаграмма создана в Access.



*ER-диаграмма:*

На диаграмме представлено 7 таблиц с их уникальными ключами:

1. Вид продукта (Например, кредит наличными, ипотека, автокредит).

Здесь код вида продукта – уникальный код определенного продукта (числовой счетчик), так же наименование продукта (длинный тест). Например, первый вид продукта будет иметь код вида продукта = 1, наименование = Кредит наличными).

1. Цель кредита (Кредит на товары и услуги, на жилье, на машину …).

Аналогично виду продукта: уникальный код и наименование цели.

1. Паспортные данные клиента.

Удобнее хранить в отдельных таблицах информацию о паспортных данных. В данной таблице столбцы: уникальный код – код паспорта (числовой счетчик) определенного человека, серия, номер паспорта (длинные цифры) и другая информация о паспорте человека (с типом данных длинный текст). Такая таблица должна быть связана с таблицей клиент.

1. Информация о работе клиента.

Код работы – уникальное число, несущее информацию о данных места работы определенного человека. Таблица связана с таблицей клиент.

1. Дополнительные услуги (Страховка, юр помощь).

Аналогично виду продукта: уникальный код и наименование доп. услуг.

1. Сам клиент. Со связями на паспортные данные, сведения о работе и заявке на кредит.

Здесь ФИО, тел, почта - длинный текст.

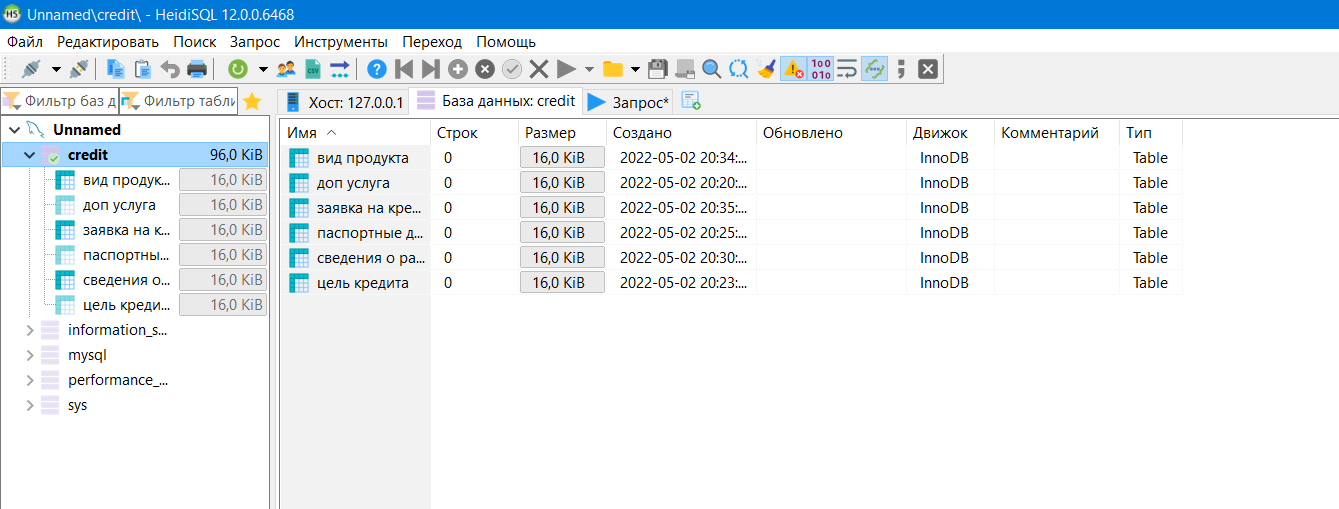
1. Заявка на кредит с уникальным кодом, принадлежащая определенному человеку. Таблица имеет связи с видом продукта, целью кредита и доп. услугами.

Так же здесь есть столбцы с суммой кредита, сроком погашения, ставкой кредита и итоговой суммой кредита (сумма + доп услуги) – тип длинные цифры.

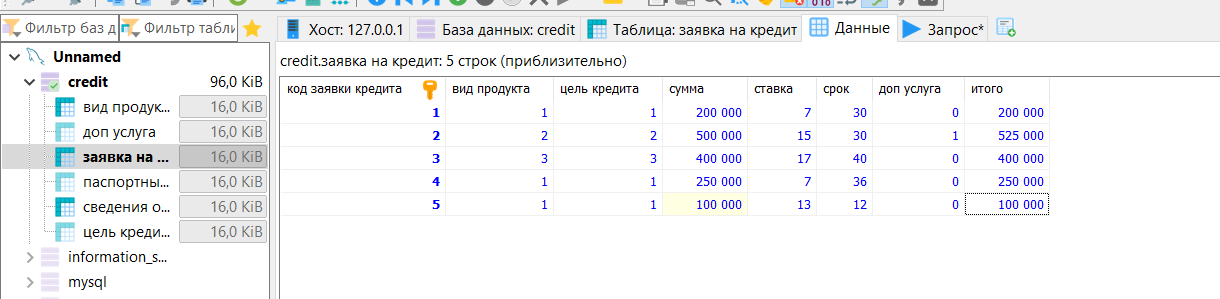
**Задача 2.**

SQL-запрос на самый популярный вид продукта

Для удобства использования БД с заданием 3 БД была сделана в программе HeidiSQL.

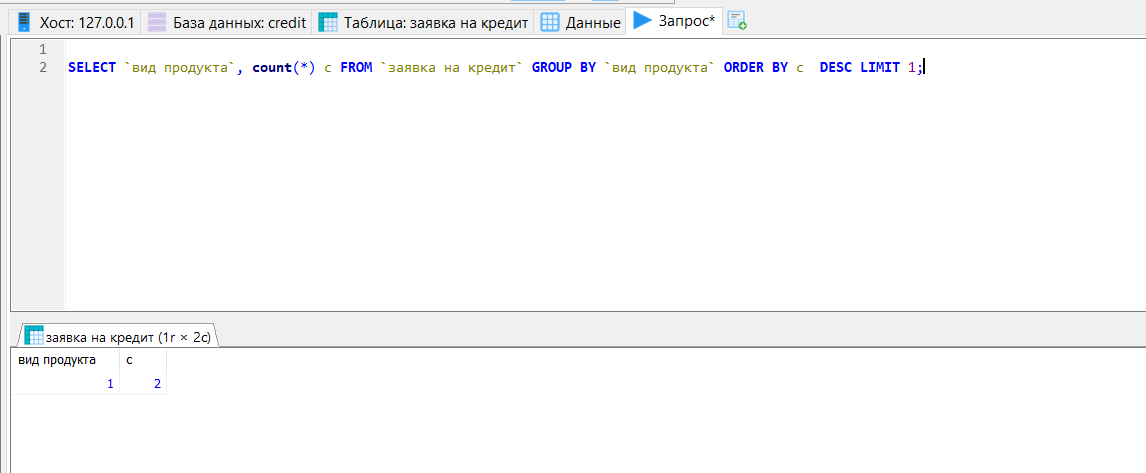


*База данных в HeidiSQL*



*Данные таблицы «Зявка на кредит»*

**SQL – Запрос:**



*SQL – Запрос*

Написан простой SQL – Запрос, который достает таблицы «заявка на кредит» часто встречающееся число (уникальный ключ вида продукта) из столбца «виды продукта».

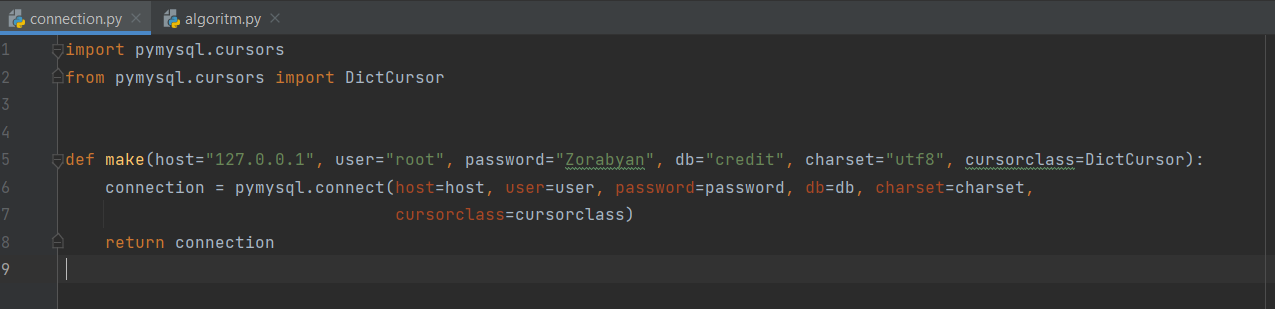
Здесь вид продукта с уникальным ключом 1 встречается чаще. С – сколько раз встретился вид продукта в таблице данных. В данном случае 2 раза.

**Задача 3**

Алгоритм для расчета графика аннуитетных платежей, функция, рассчитывающая график платежей с 5 столбцами: «Месяц», «Ежемесячный платеж», «Основной долг», «Долг по процентам» и «Остаток основного долга».

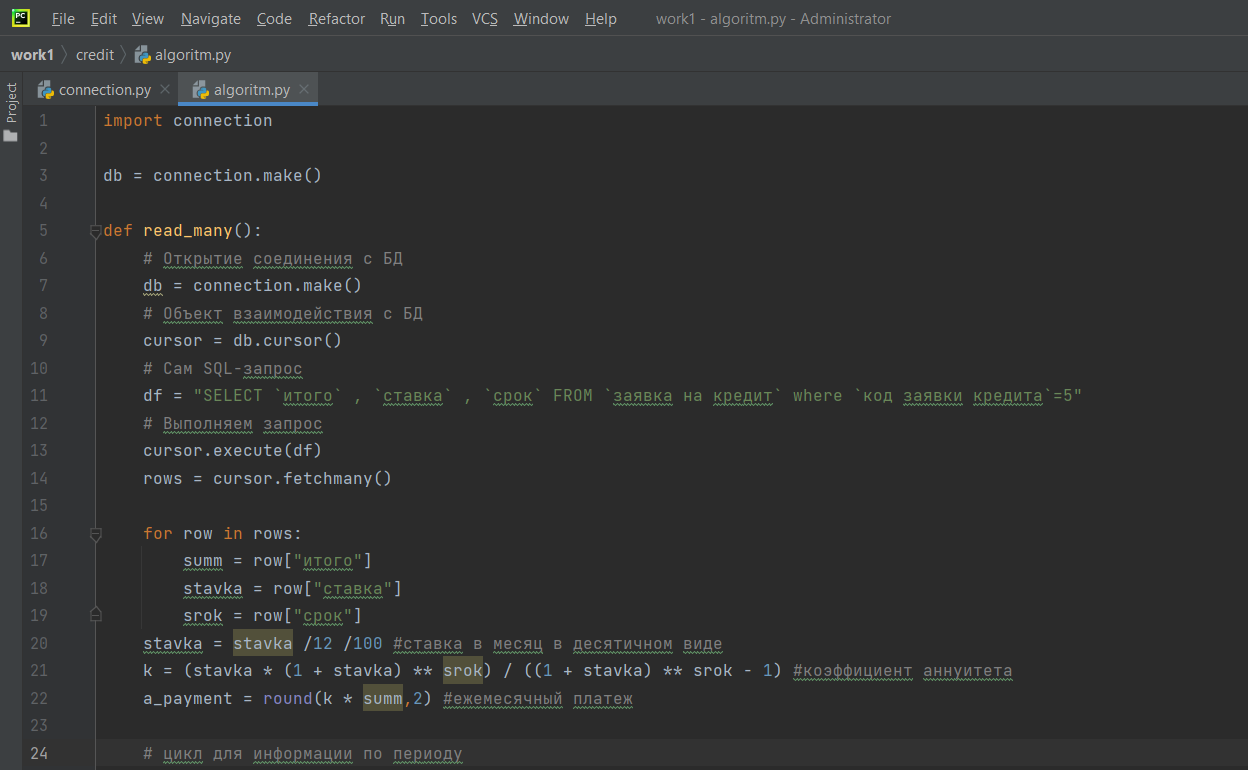
Алгоритм написан на языке Python. Создано 2 файла:

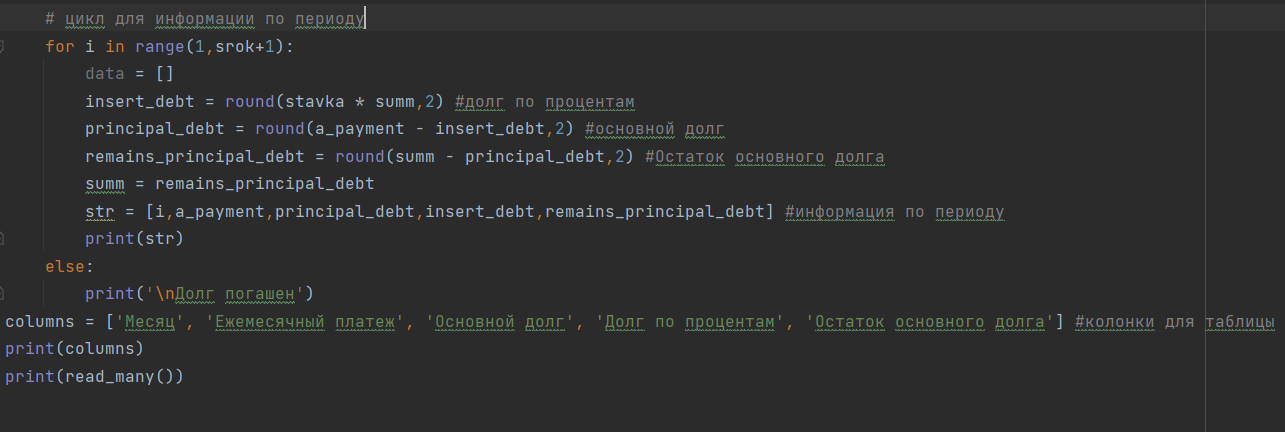
* connection.py – простой модуль для подключения к Базе Данных.



*Подключение к БД connection.py*

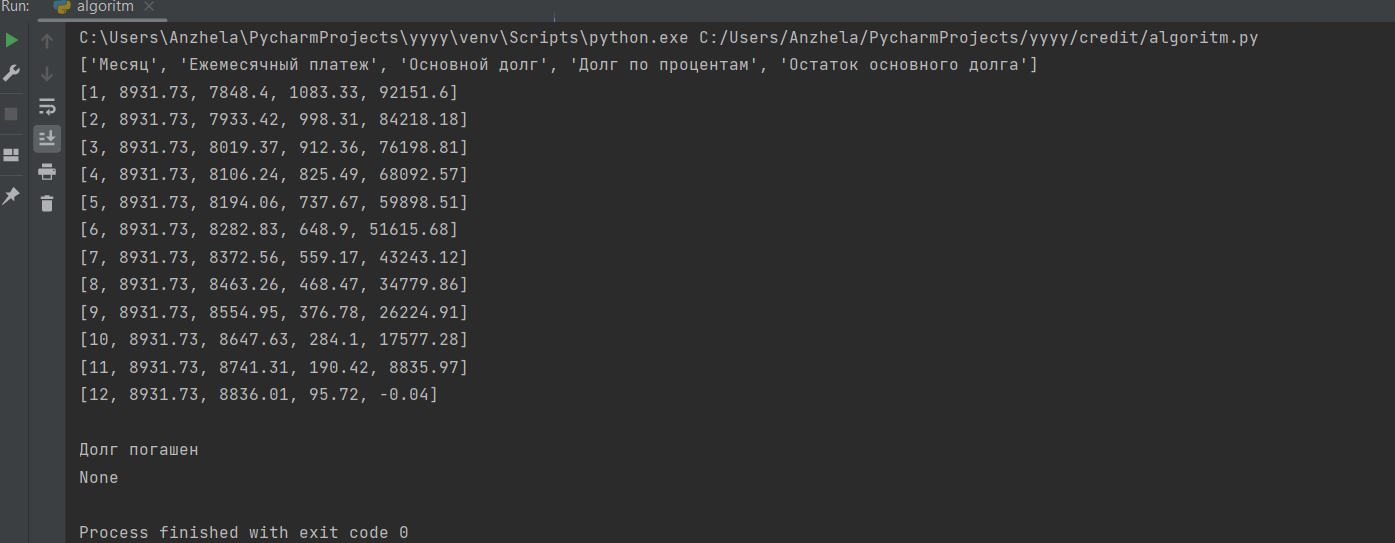
* algoritm.py – файл с алгоритмом платежей.





*Алгоритм*

Скриншот работы программы:



*Результат программы*

**В connection.py:**

Создана функция make, которая будет создавать соединение с БД и возвращать модуль, при помощи которого можно управлять БД.

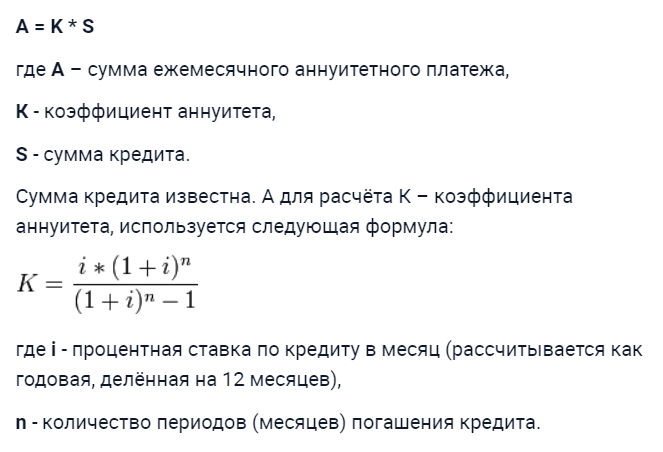
**В algoritm.py**

Импортируем модуль connection и открываем соединение с БД.

Далее функция. Здесь пишется sql запрос и достаются значения из столбцов итоговая сумма, срок и процентная ставка. В данном случае использовала 5-ую заявку на кредит из скрина БД Таблицы «Заявка на кредит» HeidiSQL.

Далее цикл. Приравниваются переменные к значениям. Рассчитывается ставка в месяц в десятичном виде (годовая процентная ставка делится на 12 месяцев и на 100 (для десятичного вида)).

Затем рассчитывается коэффициент аннуитета k по формуле:



Далее рассчитывается ежемесячный платеж: коэффициент аннуитета умножается на сумму кредита.

Следующий цикл написан для вывода информации по каждому периоду. Цикл от 1 до кол-ва периодов. Рассчитывается Долг по процентам (ставка в месяц умножить на сумму кредита), основной долг (ежемесячный платеж минус долг по процентам), остаток основного долга (сумма минус основной долг). Далее сумме приравнивается остаток основного долга и цикл продолжается до последнего периода.

Далее выводятся колонки и функция.

**\*Прикрепляю файл алгоритма программы с расширением .py.**