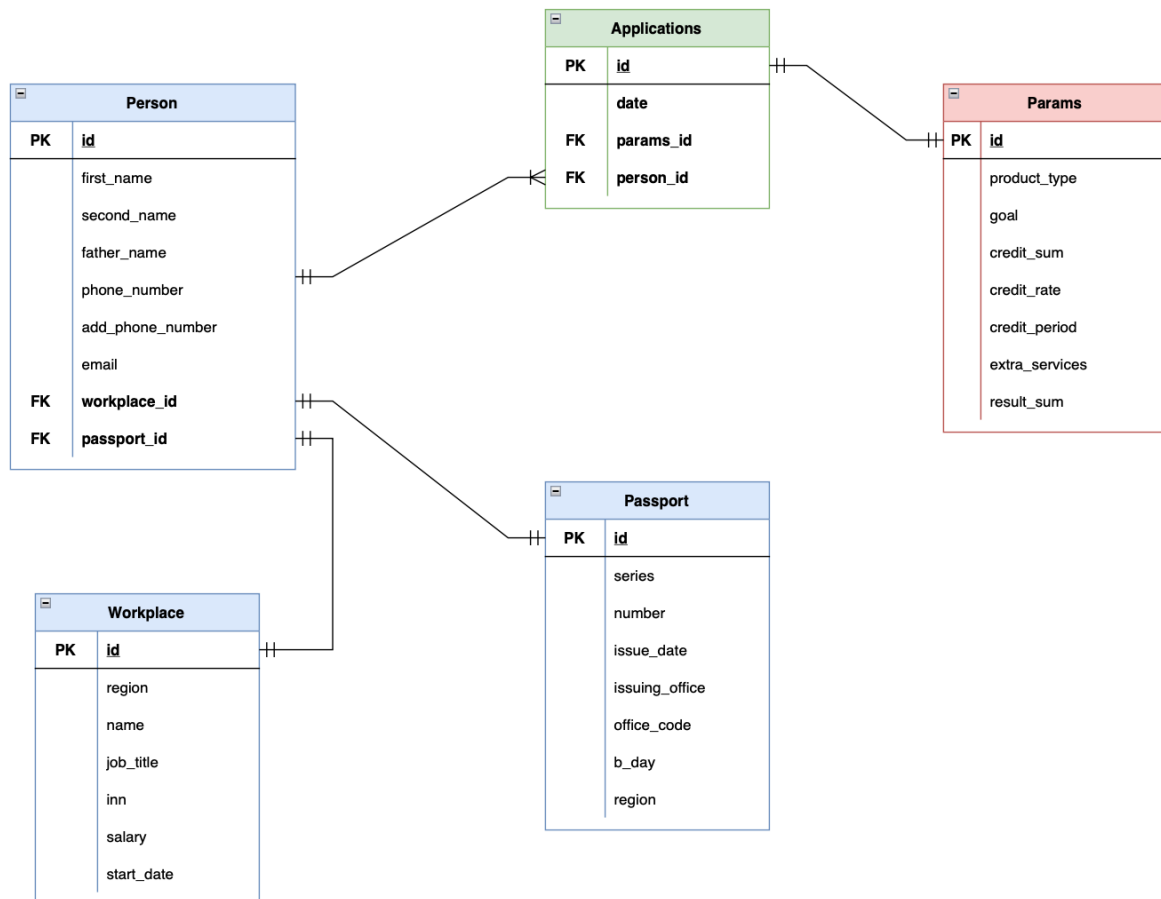


Задание 1.

В основе данного приложения следует использовать реляционную модель данных.

ER-диаграмма базы данных:



Сущности:

Applications	
Имя атрибута	Тип данных и описание
id	INT Первичный ключ
date	DATE Дата поступления заявки
params_id	INT Внешний ключ к

	сущности Params
person_id	INT Внешний ключ к сущности Person

Params	
Имя атрибута	Тип данных и описание
id	INT Первичный ключ
product_type	ENUM Вид продукта
goal	ENUM Цель кредита
credit_sum	INT Сумма кредита
credit_rate	INT Размер ставки
credit_period	INT Срок кредита (на сколько месяцев)
extra_services	SET Дополнительные услуги
result_sum	INT Сумма кредита с учетом доп. услуг

Passport	
Имя атрибута	Тип данных и описание
id	INT Первичный ключ
series	INT Серия паспорта

number	INT Номер паспорта
issue_date	DATE Дата получения
issuing_office	VARCHAR(7) Код подразделения
b_day	DATE Дата рождения
region	VARCHAR(100) Регион регистрации

Workplace	
Имя атрибута	Тип данных и описание
id	INT Первичный ключ
region	VARCHAR(100) Регион работы
name	VARCHAR(100) Название организации
job_title	VARCHAR(100) Название должности
inn	INT ИНН
salary	INT Зарплата
start_date	DATE Дата начала работы

Person	
Имя атрибута	Тип данных и описание

id	INT Первичный ключ
first_name	VARCHAR(100) Имя заемщика
second_name	VARCHAR(100) Фамилия
father_name	VARCHAR(100) Отчество
phone_number	INT Номер телефона
add_phone_number	INT Дополнительный номер телефона
email	VARCHAR(100) Электронная почта
passport_id	INT Внешний ключ к сущности Passport
workplace_id	INT Внешний ключ к сущности Workplacet

Задание 2.

SQL-запрос, который возвращает самый популярный вид продукта за текущий год:

```
SELECT product_type, COUNT(*) applications_in_2022
FROM Applications
JOIN PARAMS DESCRIPTION USING(id)
WHERE YEAR(date) = 2022
GROUP BY product_type
ORDER BY applications_in_2022 DESC
LIMIT 1;
```

Используемая СУБД: MySQL.

Задание 3.

Код программы для расчета графика аннуитентных платежей на языке JavaScript:

```
function interest_debt(sum, year_rate) {
    return sum*year_rate*30/360;
}

function principal_debt(payment, interest_debt){
    return payment-interest_debt;
}

function output_line(month, payment, sum, principal_debt, interest_debt)
{
    const sep = " | ";

    console.log(
        month + sep +
        payment.toFixed(2) + sep +
        principal_debt.toFixed(2) + sep +
        interest_debt.toFixed(2) + sep +
        sum.toFixed(2)
    );
}

function calc_payments(sum, year_rate, num){

    console.log("Сумма кредита: " + sum);
    console.log("Ставка: " + year_rate + "%");
    console.log("Срок: " + num);

    const month_rate = year_rate/(12*100);

    const payment = sum*month_rate*Math.pow(1+month_rate,
num)/(Math.pow(1+month_rate, num)-1);

    const table = "Месяц | Ежемесячный платеж | Основной долг |
Долг по процентам | Остаток основного долга";

    console.log(table);

    for (var month = 1; month <= num; month++) {

        const interest_debt = (sum*year_rate*30)/(360*100);
        const principal_debt = payment-interest_debt;
```

```

        sum = sum - principal_debt;

        output_line(month, payment, sum, principal_debt,
interest_debt);
    }
}

```

Описание функций:

`interest_debt(sum, year_rate)` - функция, которая рассчитывает сумму процентов по ссуде за месяц;

`principal_debt(payment, interest_debt)` - функция, которая рассчитывает сумму в погашение тела займа;

`output_line(month, payment, sum, principal_debt, interest_debt)` - функция, которая выводит строку со следующими данными: «Месяц», «Ежемесячный платеж», «Основной долг», «Долг по процентам» и «Остаток основного долга».

`calc_payments(sum, year_rate, num)` - функция, которая считает рассчитывает график аннуитентных платежей.

Логика функции `calc_payments`:

Для упрощения модели полагалось, что в каждом месяце 30 дней.

Расчетная формула выглядит так

$$P = S \times \frac{r \times (1 + r)^n}{(1 + r)^n - 1}$$

Где:

P — аннуитетный платеж по кредиту,

S — запрашиваемая сумма займа,

r — процентная ставка (в месяц),

n — срок кредита в месяцах.

Тестирование работы функции проводилось при `sum = 100000`, `year_rate = 13`, `num = 12`.

Результат тестирования:

✕ 📄 🔍 ⚙️ |
 🗂️ Элементы |
 🖥️ Консоль |
 📋 Источники |
 🌐 Сеть |
 🕒 Временные шкалы |
 💾 Хранилище |
 📊 Графика |
 🗣️ Слои |
 🔍 Аудит |
 🔍 🌐

< > |
 📄 Сохранить журнал |
 👤 Симулировать жест пользователя |
 Vue |
 📝 Оценки |
 ❌ Ошибки |
 ⚠️ Предупреждения |
 📅 Журналы |
 ⌛ 🔄

Консоль очищена в 13:58:36

ⓘ Сумма кредита: 100000 ⓘ Ставка: 13% ⓘ Срок: 12 ⓘ Месцац Ежемесячный платеж Основной долг Долг по процентам Остаток основного долга ⓘ 1 8931.73 7848.39 1083.33 92151.61 ⓘ 2 8931.73 7933.42 998.31 84218.19 ⓘ 3 8931.73 8019.36 912.36 76198.82 ⓘ 4 8931.73 8106.24 825.49 68092.58 ⓘ 5 8931.73 8194.06 737.67 59898.53 ⓘ 6 8931.73 8282.83 648.90 51615.70 ⓘ 7 8931.73 8372.56 559.17 43243.14 ⓘ 8 8931.73 8463.26 468.47 34779.88 ⓘ 9 8931.73 8554.95 376.78 26224.94 ⓘ 10 8931.73 8647.62 284.10 17577.31 ⓘ 11 8931.73 8741.31 190.42 8836.00 ⓘ 12 8931.73 8836.00 95.72 -0.00	<div>f calc_payments calc_pay.js:23</div> <div>f calc_payments calc_pay.js:24</div> <div>f calc_payments calc_pay.js:25</div> <div>f calc_payments calc_pay.js:33</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div> <div>f output_line calc_pay.js:12</div>
--	---

> |
 >