

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Нереляционные БД»

Тема: Аналитика пользователей продуктовых интернет магазинов

Студенты гр. 5382

Еськов А.
Ришко С.
Нуреев Н.

Преподаватель

Заславский М.М.

Санкт-Петербург
2018

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студенты Еськов А, Ришко С, Нуреев Р.

Группа 5382

Тема проекта: Аналитика пользователей продуктовых интернет магазинов

Исходные данные:

Разработать приложение для анализа и просмотра статистики интернет-магазинов.

Содержание пояснительной записки:

«Введение», «Качественные требования к решению», «Сценарии использования», «Модель данных», «Разработанное приложение», «Публикация продукта», «Выводы», «Выводы».

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 05.09.2018

Дата сдачи реферата: 26.12.2018

Дата защиты реферата: 26.12.2018

Студент

Еськов А.
Ришко С.
Нуреев Н.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В индивидуальном домашнем задании изучается нереляционная база данных MongoDB на примере разработки приложения для статистики и аналитики интернет-магазинов.

SUMMARY

In an individual homework, MongoDB non-relational database is studied using the example of developing an application for statistics and analytics of online stores.

Оглавление

1.	Введение	6
1.1.	Актуальность решаемой проблемы.	6
1.2.	Постановка задачи	6
1.3.	Предлагаемое решение.....	6
2.	Требования к решению	6
3.	Сценарии использования	6
3.1.	Описание решения задач пользователем хранения, анализа, представления, импорта, экспорта данных.	6
4.	Модель данных	7
4.1.	Пример документа	7
4.2.	Аналог модели данных для SQL СУБД	9
4.3	Расчёт памяти	10
5.	Разработанное приложение	10
5.1.	Краткое описание	10
5.2.	Использованные технологии	11
5.3.	Ссылки на приложение	11
6.	Выводы	11
6.1.	Достигнутые результаты.....	11
6.2.	Недостатки и пути для улучшения полученного решения	12
7.	Приложения.....	12
7.1.	Документация по сборке и разворачиванию приложения	12
7.2.	Инструкция для пользователя	12
7.3.	Снимки экрана приложения	14

1. Введение

1.1. Актуальность решаемой проблемы.

Решаемая проблема – просмотр детальной статистики по периодам, просмотр заказов, пользователей и отзывов интернет-магазинов.

1.2. Постановка задачи

Необходимо разработать web-приложение, позволяющее отслеживать поведение пользователей в интернет-магазине.

Основные функции:

- Просмотр статистики по периодам
- Просмотр активных пользователей
- Просмотр заказов
- Импорт и экспорт данных

1.3. Предлагаемое решение

Предлагается разработать продукт с использованием MongoDB и Node.js в качестве веб-сервера.

2. Требования к решению

Приложение имеет форму входа для администраторов магазинов, удобный и понятный интерфейс, высокую скорость загрузки данных.

3. Сценарии использования

3.1. Описание решения задач пользователем хранения, анализа, представления, импорта, экспорта данных.

Хранение:

В БД будут храниться следующие данные:

- Информация о магазинах
- Товары, сгруппированные по магазинам
- Информация о сессиях по каждому магазину
- Информация о пользователях по каждому магазину

Анализ:

Запрос на добавление нового юзера (первый визит на сайт):

```
db.library.insertOne({_id:1,session_start:"01.09.2018  
15:35",session_end:"01.09.2018 16:12",uid:"u12345",targets:[...]})
```

Вычисление расходов для каждой потребности для всех месяцев:

```
db.library.find({session_start: {$gte: ISODate("2018-04-29T00:00:00.000Z"), $lt: ISODate("2018-05-01T00:00:00.000Z")}}).count()
```

Запрос на добавление новой сессии:

```
db.library.update( { key: "JKsdqHJ213"}, { $push: { "sessions": { _SESSION-INFO_ } } }); INSERT INTO sessions (session_id, uid, start, stop) VALUES(2137, 'u728913DHSas', '01.09.2018 15:35', '01.09.2018 16:12');
```

4. Модель данных

4.1. Пример документа

```
{
  "key": "JKsdqHJ213",
  "admin": {
    "login": "admin",
    "pass": "admin"
  },
  "name": "Shop",
  "text": "Классный магазин",
  "offers": [{
    "_id": 1,
    "name": "Iphone",
    "cat": "Телефоны",
    "price": 99000,
    "text": "Супер классный телефоны"
  }],
  "pages": [{
    "_id": 13,
    "url": "/page?id=13",
    "name": "Home"
  }],
  "sessions": [{
    "session_id": "7hkjsahdasd23",
    "uid": "uShjo1238",
    "session_start": "27.08.2018 15:35",
    "session_end": "27.08.2018 16:03",
    "targets": [{
      "type": "Изменение корзины",
      "date": "27.08.2018 15:55",
      "offers": [{
        "$ref": "offers",
        "$id": 13,
        "$db": "shops"
      }],
      "type": "Просмотр страницы",
      "date": "27.08.2018 15:48",
      "page": {
        "$ref": "pages",

```

```
        "$id": 11,  
        "$db": "shops"  
    },  
    {  
        "type": "Отзыв",  
        "date": "27.08.2018 15:48",  
        "review": {  
            "nick": "User777",  
            "text": "Классный товар!",  
            "offer": {  
                "$ref": "offers",  
                "$id": 22,  
                "$db": "shops"  
            }  
        }  
    }  
]  
}]  
}
```

4.2.Аналог модели данных для SQL СУБД – графическое представление данных и сравнение с моделью данных для NoSQL БД.

Если сравнить с моделью данных для NoSQL СУБД, можно заметить, что у SQL СУБД модель лучше нормализована.

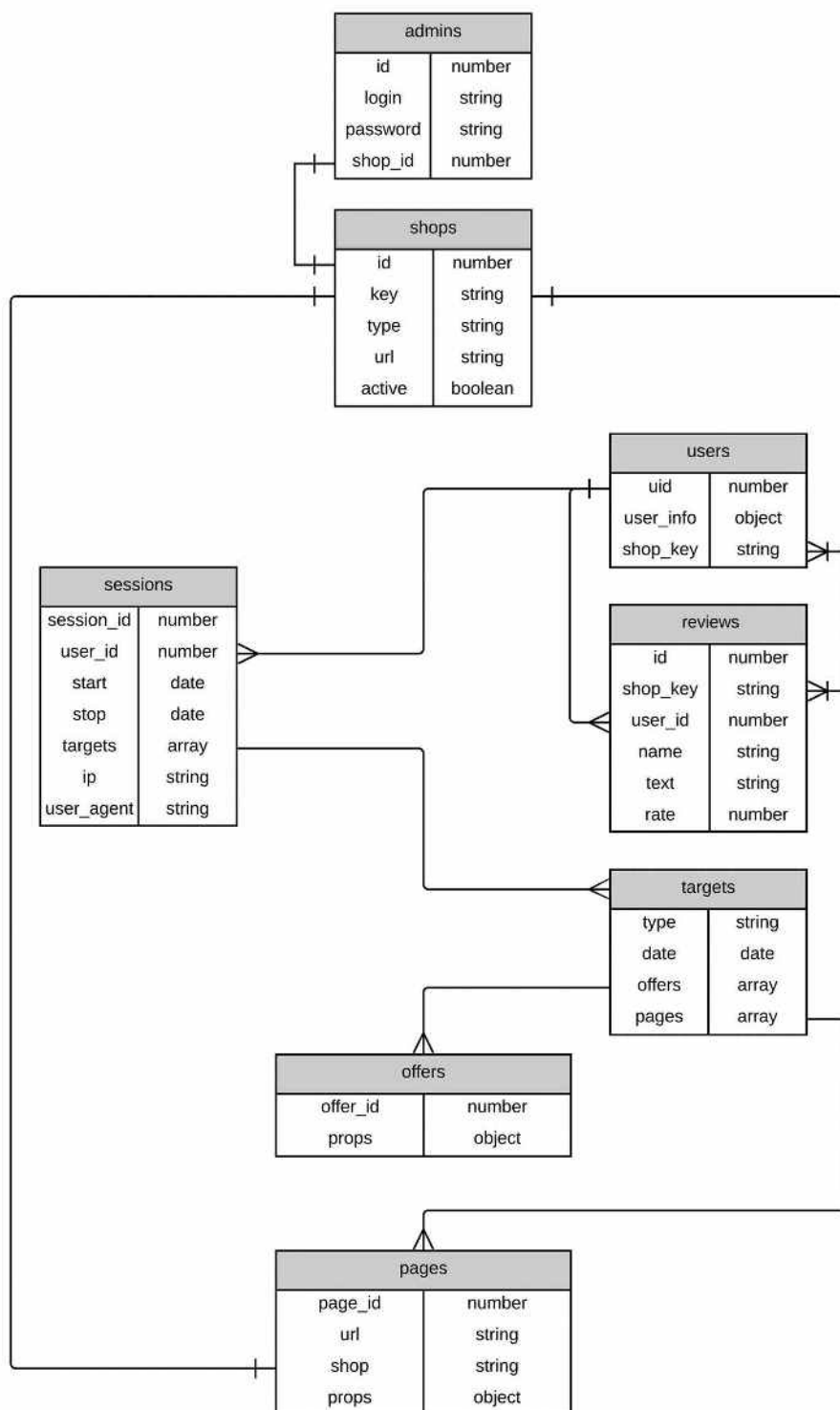


Рис 2. Модель данных для SQL СУБД

Анализ:

Запрос на добавление нового юзера (первый визит на сайт):

```
INSERT INTO users (uid, shop_key) VALUES('u728913DHSas',  
'Keh91y9ASDH');  
INSERT INTO sessions (session_id, uid, start, stop) VALUES(2137,  
'u728913DHSas', '01.09.2018 15:35', '01.09.2018 16:12');
```

Запрос на количества клиентов в конкретный день:

```
SELECT COUNT(*) FROM sessions WHERE user_id IN (  
    SELECT uid FROM users WHERE shop_key = 'JE01o3kls21';  
) AND start BETWEEN 2018-04-29 AND 2018-05-01
```

Запрос на добавление новой сессии:

```
db.library.update( { key: "JKsdqHJ213"}, { $push: { "sessions":  
{_SESSION-INFO_} } } ); INSERT INTO sessions (session_id, uid,  
start, stop) VALUES(2137, 'u728913DHSas', '01.09.2018 15:35',  
'01.09.2018 16:12');
```

4.3 Расчёт памяти

Хранение данных будет занимать:

1. В MongoDB будет занимать $M * P * N + M * P * K$
2. В реляционных базах данных будет занимать $M * N + M * P * K$

В MongoDB - 474000 байт

В реляционных базах данных - 481000 байт

Пусть кол-во магазинов N , кол-во всех юзеров - M , P - среднее кол-во визитов. Тогда запрос на поиск всех визитов юзера в конкретный магазин будет занимать $O(MNP)$, в реляционных БД нужно будет делать два join'a, в следствие чего мы проиграем в производительности.

5. Разработанное приложение

5.1.Краткое описание

Результатом разработки приложения на клиентской стороне стало SPA(Single Page Application), состоящее из 5 страниц:

1. Страница авторизации
Страница для авторизации администратора интерне-магазина.

2. Главная страница с основной информацией;

На данной странице отображается основная информация о магазине

3. Страница агрегированной статистики;

На этой странице отображаются данные о посещении сайта, среднем чеке, количестве покупок, конверсии за за выбранный период

4. Страница пользователей;

На странице можно увидеть, когда и какой пользователь совершил заказ, а также детали заказа.

5. Страница заказов;

На странице можно увидеть, кем, когда и какой товар был куплен

6. Страница отзывов включает в себя отзывы о текущем магазине, есть возможность включать и выключать их отображение.

5.2.Использованные технологии

Использованные технологии:

- MongoDB — документоориентированная СУБД;
- Express.js — каркас веб-приложений, работающий поверх Node.js;
- jQuery – Java-script фреймворк для взаимодействия с HTML документом.
- Node.js — JavaScript платформа для серверной разработки.

5.3.Ссылки на приложение

Исходный код приложения и инструкция по установке находится по ссылке:

https://github.com/moevm/nosql2018-grocery_store_analysis

6. Выводы

6.1.Достигнутые результаты

Достигнутые результаты:

- Приложение разработано меньше, чем за 80 часов;
- Приложение достаточно динамичное и быстрое;
- Приложение решает минимальные нужды пользователя.

6.2. Недостатки и пути для улучшения полученного решения

В рамках курса “Нереляционные БД” приложение можно считать хорошо разработанным, но в рамках будущего продукта существует ряд недостатков:

- Технический долг

Причина: отсутствие опыта в выбранных технологиях.

Решение: потребуется около 16 часов для рефакторинга, оптимизации запросов и улучшения юзабилити приложения.

- Недостаточный функционал для продукта

Текущее состояние приложение можно оценивать, как прототип продукта, т.к. функционала для желаемого отслеживания денежного потока недостаточно.

Причина: недостаточное время для разработки и отсутствие опыта у разработчика.

Решение: продолжать разработку продукта.

7. Приложения

7.1. Документация по сборке и развертыванию приложения

Инструкция по сборке и запуску:

1. Скачать проект из репозитория;
2. Перейти в корневую папку проекта и в терминале ввести: `npm install`;
3. Запустить сервер MongoDB: `mongod`;
4. Запустить Node-сервер: `node server.js`;
5. Перейти в браузере по адресу: `http://localhost:8080`.

7.2. Инструкция для пользователя

Вход

При входе на сайт отображается форма для входа. Пользователь должен ввести свои данные (логин, пароль), чтобы получить доступ к ресурсу. На странице входа можно импортировать данные по своему магазину

Главная страница

После входа пользователь попадает на главную страницу, на которой отображается основная информация о пользователях интернет-магазина. Переключение между вкладками осуществляется с помощью кнопок бокового меню.

Отчеты

Во вкладке "Отчеты" отображается расширенная статистика(посещаемость, конверсия, выручка) с графиками за выбранный период времени.

Пользователи

Во вкладке "Пользователи" отображается информация о каждом клиенте и его покупках. При клике на ID клиента отображается более подробная информация о нем.

Заказы

Во вкладке "Заказы" отображается таблица с информацией о поступивших заказах.

Отзывы

Во вкладке "Отзывы" отображаются отзывы об интернет-магазине, оставленные клиентами.

7.3. Снимки экрана приложения

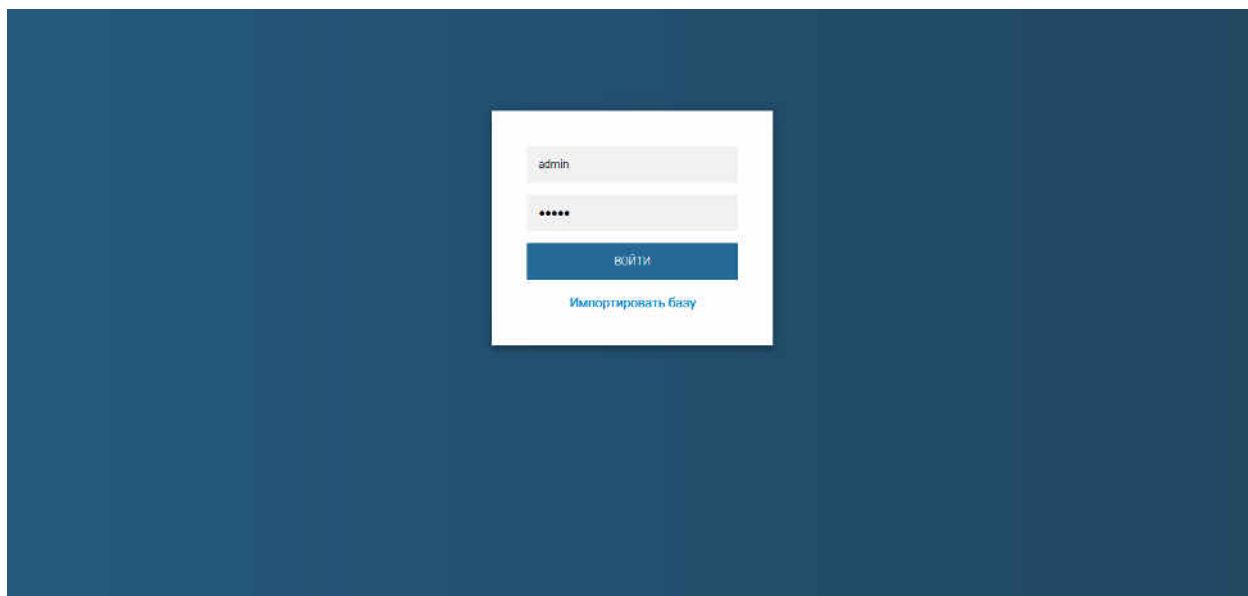


Рис 6. Форма входа

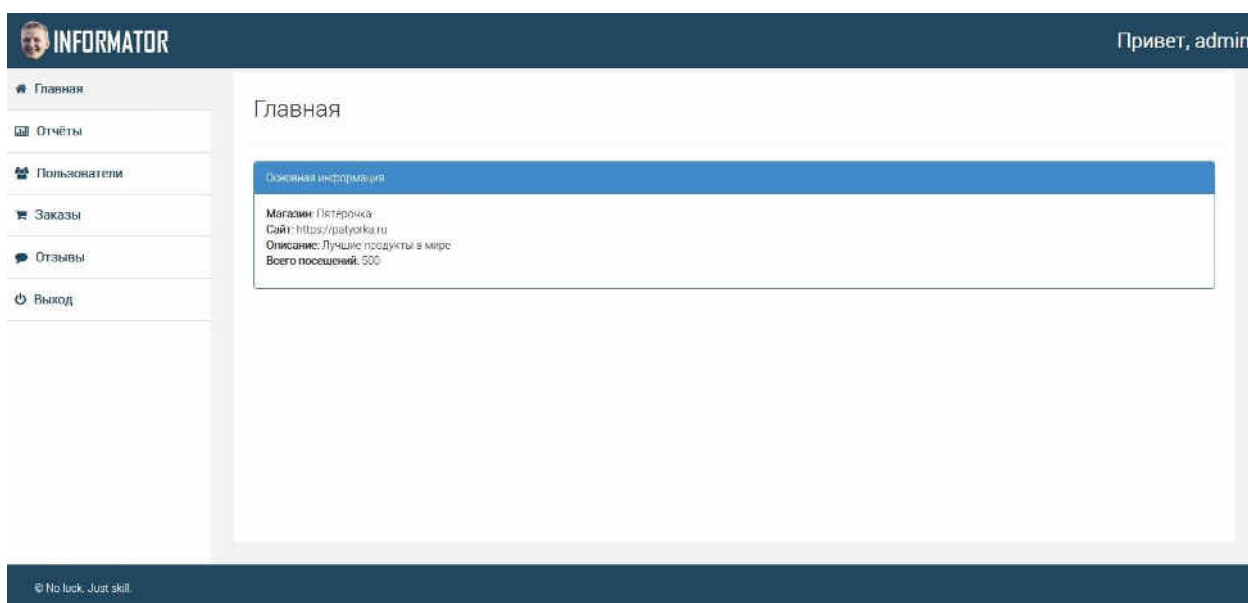


Рис 7. Главная страница

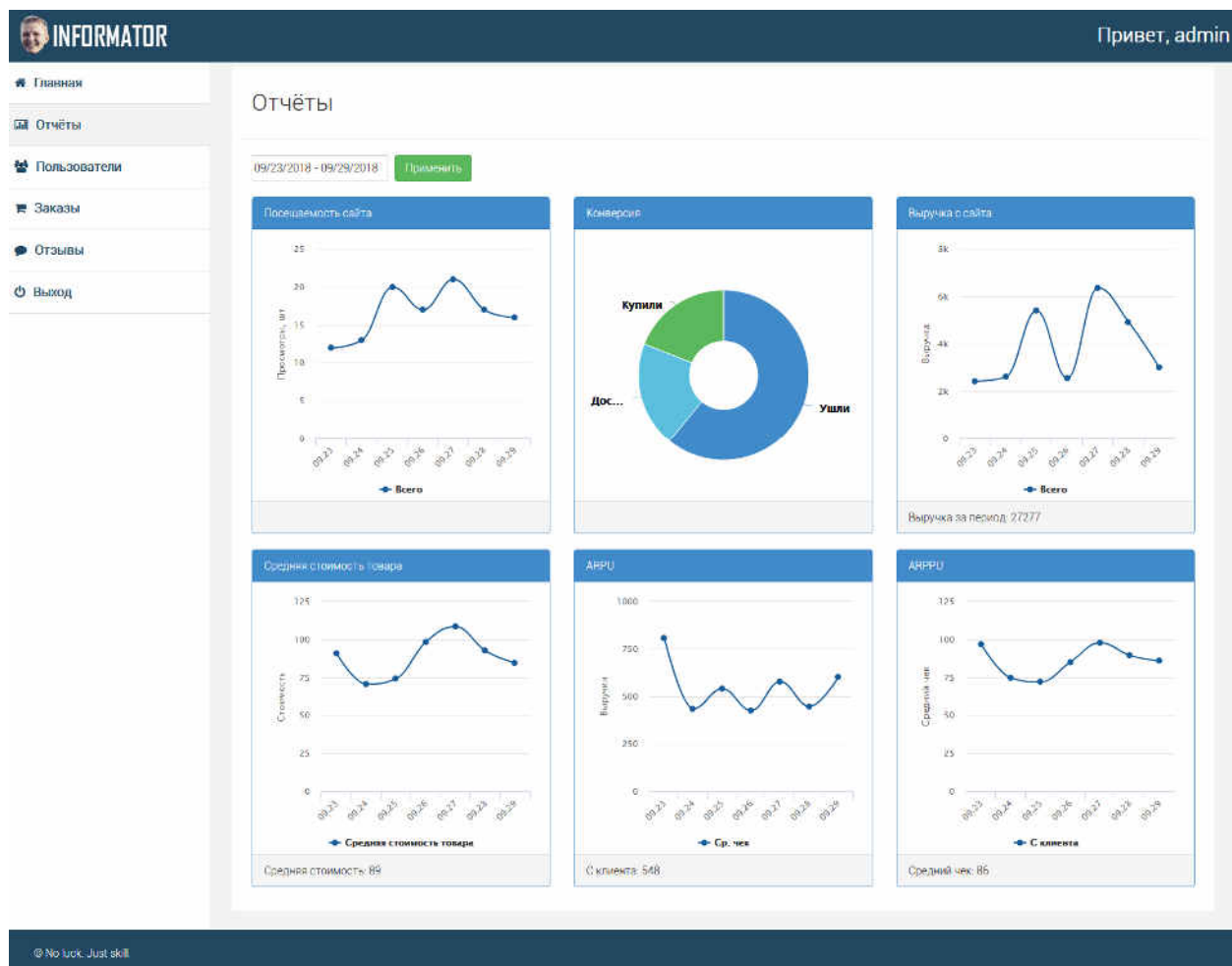


Рис 8. Отчеты

INFORMATOR Привет, admin

Главная | Отчёты | **Пользователи** | Заказы | Отзывы | Выход

Пользователи

#	uid	Визитов	Покупок	Куплено товаров	Контакты
1	uU1NFJzA	10	0	0	
2	u078e6iPp	18	8	15	ehA6dy2v@yandex.ru, ne5Zlog@gmail.com, iM6CRio4@yandex.ru, Yv8ESm2@yandex.ru, R6DesQW2@mail.ru, sGT3Qx3w@yandex.ru, K8mqBMlu@yandex.ru, RyVozabb@yandex.ru
3	u2qWojHf0	13	1	5	jkri0aBh@yandex.ru
4	uc9cZFdbx	4	1	5	MJYo3YVv@gmail.com
5	uETSyh6Gq	3	1	5	caMwtTIE@mail.ru
6	uCfozHT0	11	3	10	IKKnFgV@yandex.ru, 8RR3ujU6@yandex.ru, tfoz43J2@gmail.com
7	uqW07NisV	11	2	10	VqUT52vo@mail.ru, SO06m021@yandex.ru
8	ujv6DjGnq	5	0	0	
9	uCEqatbjb	13	2	10	C9LYSPmU@mail.ru, koDEHMP@mail.ru
10	u8rp1Pzd0	5	0	0	

« « > >> 10

© No luck. Just skill

Рис 9. Пользователи

#	Дата	Пользователь	Товары	Итоговый чек
1	2018-9-1 03:37:36	u1o2gW4Hk	Чайник, Холодильник, Миксер, Тостер, Плита газовая	111500 р.
2	2018-9-1 05:16:48	ugLecMJM	Чайник, Хлебопечка, Микроволновая печь, Электроник, Тостер	89500 р.
3	2018-9-1 06:22:14	uMAujurg	Холодильник, Хлебопечка, Микроволновая печь, Тостер, Плита газовая	180000 р.
4	2018-9-1 13:44:59	u1atX06iG	Чайник, Миксер, Морозильник, Микроволновая печь, Тостер	96500 р.
5	2018-9-1 13:48:20	u1adX06iG	Чайник, Холодильник, Морозильник, Тостер, Плита газовая	122500 р.
6	2018-9-1 19:33:13	uXZ0arBtB	Чайник, Миксер, Морозильник, Хлебопечка, Плита газовая	70800 р.
7	2018-9-1 19:30:32	uK9zFHA7J	Осковыжмалка, Миксер, Морозильник, Хлебопечка, Электроник	40000 р.
8	2018-9-2 01:29:56	ug96SXRE2	Чайник, Холодильник, Миксер, Хлебопечка, Электроник	80500 р.
9	2018-9-2 01:31:40	ug96SXRE2	Чайник, Осковыжмалка, Морозильник,	76600 р.

Рис 10. Заказы