Федеральное государственное автономное образовательное учреждение

высшего образования

"Национальный исследовательский университет

"Высшая школа экономики"

Московский институт электроники и математики им. А.Н. Тихонова НИУ ВШЭ

Курс: Базы данных

**ОТЧЕТ**

**по домашней работе**

**Тема: Служба доставки**

Студенты: Соколовская Карина Евгеньевна,

Царегородцева Елизавета Игоревна

Группа: БИВ196

Москва, 2021 г.

**Служба доставки**

База данных создается для онлайн - службы доставки еды из ресторанов.

В соответствии с предметной областью система строится с учетом следующих особенностей:

1. Каждый ресторан может специализироваться на нескольких кухнях.
2. Каждое блюдо могут предоставлять сразу несколько ресторанов.
3. Каждое блюдо может относиться только к одному виду кухни.
4. Каждое блюдо может относиться только к одному типу блюд.
5. Каждое блюдо может относиться только к одному типу диеты.
6. Скидка на блюдо одинакова для всех ресторанов.
7. Блюда, заказанные в разных ресторанах, попадают в один заказ.
8. Каждый заказ доставляется одним сотрудником. Количество ресторанов, предоставляющих блюда для данного заказа, не имеет значения.
9. Каждый клиент может сделать несколько заказов.
10. Разные заказы каждому клиенту могут доставлять разные сотрудники.
11. Стоимость заказа считается с учетом скидок на определенные

блюда.

1. Курьер не может нести блюд больше чем на 7 кг.
2. Курьеры доставляют заказы, а сборщики собирают и оформляют их
3. Занятость отслеживается только у курьеров
4. Итоговая стоимость зависит от статуса клиента, и стоимость доставки одинакова для всех и не зависит от ресторанов. У новых клиентов цена одинаковая и максимальная

**Сущности:**

**Идентифицирующие** атрибуты выделим полужирным шрифтом, *многозначные* – курсивом, составные подчеркнем.

1. **Блюда**

Атрибуты:

1. **Идентификатор блюда**
2. Название
3. *Тип*
4. *Вид кухни*
5. Вес порции(значение, которое получается в среднем)
6. КБЖУ
7. Стоимость 100 гр.
8. Состав
9. Срок годности
10. *Диета* (вегетарианское, веганское, кето и т.д.)
11. Скидка
12. Рейтинг
13. **Сотрудники нашей службы доставки**

Атрибуты:

1. **Идентификатор сотрудника**
2. ФИО
3. Паспортные данные
4. Дата рождения
5. *Пол*
6. ИНН
7. СНИЛС
8. Адрес
9. *Телефоны*(мобильный, рабочий, домашний)
10. *Должность*
11. Оклад
12. *Занят/не занят на данный момент*
13. **Заказы**

Атрибуты:

1. **Номер заказа**
2. Дата совершения заказа
3. *Статус* (обработка, сборка, передан курьеру, доставлен)
4. Количество персон
5. Адрес доставки
6. Сумма заказа
7. Вес заказа
8. **Клиенты**

Атрибуты:

1. **Идентификатор клиента**
2. ФИО
3. Дата рождения
4. Мобильный телефон
5. Домашний адрес
6. Рейтинг

**5) Рестораны**

Атрибуты:

1. **Идентификатор ресторана**
2. Название
3. Минимальный заказ
4. Стоимость доставки
5. Время доставки
6. Рейтинг

**ER - диаграмма таблиц:**

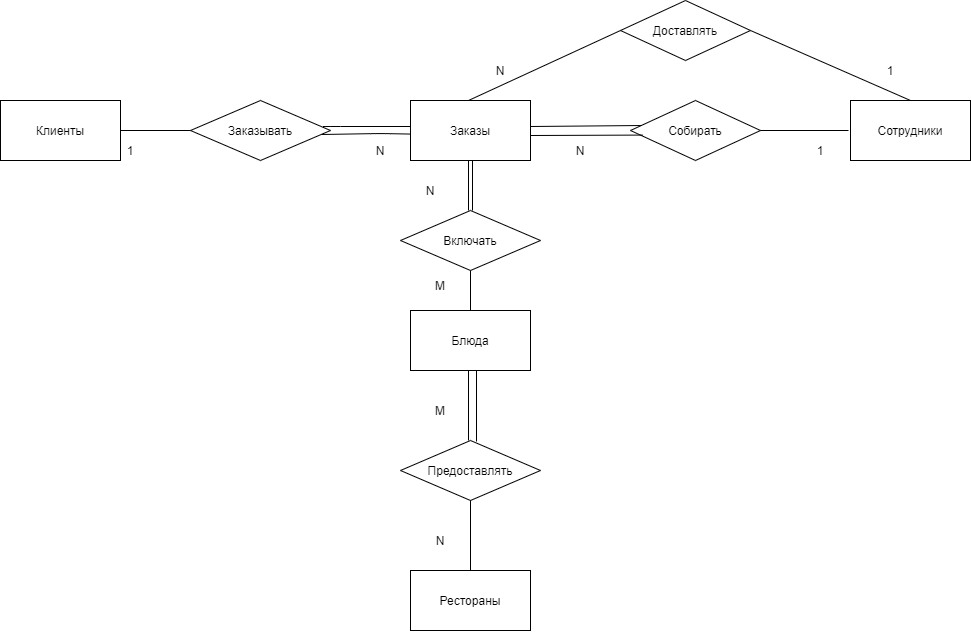


Рис. 1. ER-диаграмма

**Определение требований к операционной обстановке**

Определим групп пользователей, их основные задачи и запрос к БД:

1. Клиент:

* Получение информации о блюдах
* Получение информации о ресторанах, из которых возможен заказ блюд
* Получение информации о себе
* Изменение информации о себе
* Получение информации о заказе
* Изменение информации о заказе

1. Директор службы доставки:

* Получение информации о всех сотрудниках
* Получение информации о ресторанах, с которыми заключены контракты
* Назначение сотрудников отдела кадров и своего зама

1. Заместитель директора службы доставки:

* Получение информации о сотрудниках
* Изменение и актуализация данных о блюдах
* Внесение в список новых ресторанов, после заключения контрактов с ними, а также изменение и актуализация данных о них

1. Курьер:

* Получение информации о заказе, изменение его статуса

1. Сборщик:

* Получение информации о заказе, изменение его статуса

1. Сотрудники отдела кадров:

* Приём и увольнение сотрудников на работу (добавление/исключение из работы сотрудников)
* Внесение изменений в информацию о сотрудниках

1. Менеджер:

* Отслеживание работы сотрудников
* Получение информации о заказе
* Изменение статусов заказов

**Преобразование ER-диаграммы в схему базы данных**

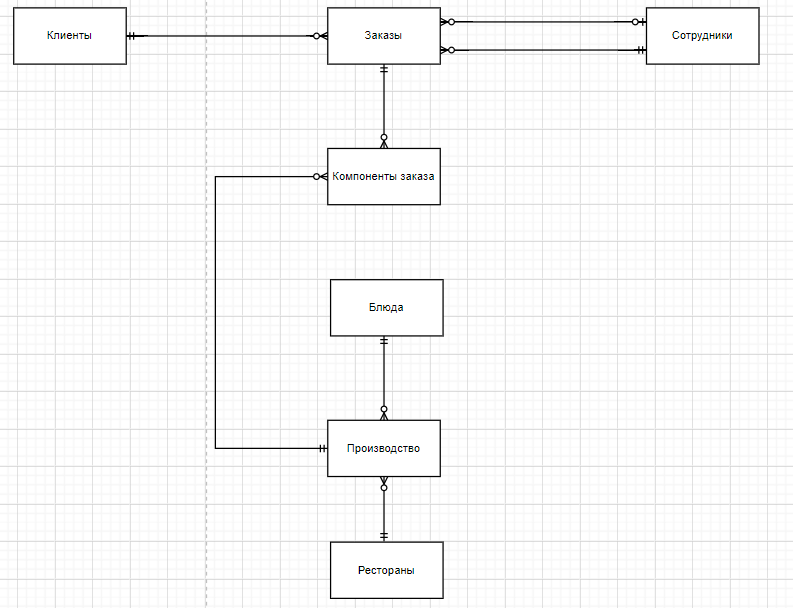
****

Рис. 2. Схема РБД

**Составление реляционных отношений**

Таблица 1. Схема отношения “Клиенты”(Clients)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор клиента | cl\_id | NUMERIC(5) | **первичный ключ** |
| ФИО | cl\_name | VARCHAR(60) | обязательное поле |
| Дата рождения | cl\_born | DATE | обязательное поле |
| Мобильный телефон | cl\_phone | VARCHAR(30) | обязательное поле |
| Домашний адрес | cl\_address | VARCHAR(60) |  |
| Рейтинг | cl\_rating | NUMERIC(4,2) | обязательное поле, [0; 10], значение по умолчанию – 0 |

Таблица 2. Схема отношения “Сотрудники”(Employees)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор сотрудника | em\_id | NUMERIC(3) | **первичный ключ** |
| ФИО | em\_name | VARCHAR(60) | обязательное поле |
| Паспортные данные | em\_passport | VARCHAR(60) | обязательное поле |
| Дата рождения | em\_born | DATE | обязательное поле |
| Пол | em\_gender | CHAR(1) | обязательное поле, “м“ или “ж“ |
| ИНН | em\_inn | VARCHAR(60) | обязательное уникальное поле |
| СНИЛС | em\_snils | VARCHAR(14) | обязательное уникальное поле |
| Адрес | em\_address | VARCHAR(60) | обязательное поле |
| Телефоны | em\_phone | VARCHAR(60) | многозначное поле |
| Должность | em\_post | VARCHAR(60) | обязательное многозначное поле |
| Оклад | em\_salary | NUMERIC(6) | обязательное поле, >12792 |
| Занятость на данный момент  -триггером см. примечание к таблице 16 | em\_status | VARCHAR(60) | многозначное поле, “занят” или “не занят”, значение по умолчанию – NULL |

Таблица 3. Схема отношения “Рестораны”(Restaurants)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор ресторана | r\_id | NUMERIC(3) | **первичный ключ** |
| Название | r\_name | VARCHAR(30) | обязательное поле |
| Минимальный заказ | r\_min | NUMERIC(3) | обязательное поле, >=0 |
| Время доставки | r\_time | VARCHAR(10) | обязательное поле |
| Рейтинг | r\_rating | NUMERIC(4,2) | обязательное поле, [0; 10] |

Таблица 4. Схема отношения “Заказы”(Orders)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер заказа | ord\_id | NUMERIC(5) | **первичный ключ** |
| Идентификатор клиента | ord\_client | NUMERIC(5) | внешний ключ к (Clients) |
| Идентификатор сотрудника  (сборщика) | ord\_emp\_s | NUMERIC(3) | внешний ключ к (Employees) |
| Идентификатор сотрудника  (курьера) | ord\_emp\_c | NUMERIC(3) | внешний ключ к (Employees) |
| Дата совершения заказа | ord\_date | DATE | обязательное поле |
| Статус | ord\_status | VARCHAR(15) | обязательное поле |
| Количество персон | ord\_persons | NUMERIC(2) | обязательное поле, >0 |
| Адрес доставки | ord\_address | VARCHAR(60) | обязательное поле |
| Сумма заказа | ord\_price | NUMERIC(5) | обязательное поле, >0 |
| Вес заказа | ord\_weight | NUMERIC(4) | обязательное поле, (0; 7000] |

Таблица 5. Схема отношения “Блюда”(Dishes)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор блюда | d\_id | NUMERIC(5) | **первичный ключ** |
| Название | d\_name | VARCHAR(30) | обязательное поле |
| Тип | d\_type | VARCHAR(15) | обязательное многозначное поле |
| Вид кухни | d\_causine | VARCHAR(20) | обязательное многозначное поле |
| Вес порции | d\_weight | NUMERIC(4) | обязательное поле, >0 |
| КБЖУ | d\_cpfc | VARCHAR(30) | обязательное поле |
| Состав | d\_ingred | VARCHAR(200) | обязательное поле |
| Срок годности | d\_shelflife | VARCHAR(10) | обязательное поле |
| Диета | d\_diet | VARCHAR(20) | обязательное многозначное поле |
| Скидка | d\_discount | NUMERIC(3) | обязательное поле, [0; 100] |
| Рейтинг | d\_rating | NUMERIC(4,2) | обязательное поле, [0; 10] |

Таблица 6. Схема отношения “Производство”(Production)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор | prod\_id | NUMERIC(5) | **первичный ключ** |
| Идентификатор блюда | prod\_dish | NUMERIC(5) | внешний ключ к (Dishes) |
| Идентификатор ресторана | prod\_rest | NUMERIC(3) | внешний ключ к (Restaurants) |
| Стоимость 100 гр. | prod\_price | NUMERIC(4) | обязательное поле, >0 |

Таблица 7. Схема отношения “Компоненты заказа”(Order items)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор блюда | i\_dish | NUMERIC(5) | внешний ключ к (Production) |
| Идентификатор заказа | i\_order | NUMERIC(5) | внешний ключ к (Orders) |
| Вес блюда(полученного по факту) | i\_weight | NUMERIC(4) | обязательное поле, (0; 500] |

**Нормализация полученных отношений**

Таблица 8. Схема отношения “Клиенты”(Clients)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор клиента | cl\_id | NUMERIC(5) | **первичный ключ** |
| Имя | cl\_fname | VARCHAR(20) | обязательное поле |
| Фамилия | cl\_lname | VARCHAR(20) | обязательное поле |
| Отчество | cl\_patronymic | VARCHAR(20) |  |
| Дата рождения | cl\_born | DATE | обязательное поле |
| Мобильный телефон | cl\_phone | VARCHAR(30) | обязательное поле |
| Домашний адрес | cl\_address | VARCHAR(60) |  |
| Рейтинг | cl\_rating | NUMERIC(4,2) | обязательное поле, [0; 10], значение по умолчанию – 0 |

Таблица 9. Схема отношения “Сотрудники”(Employees)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор сотрудника | em\_id | NUMERIC(3) | **первичный ключ** |
| Имя | em\_fname | VARCHAR(20) | обязательное поле |
| Фамилия | em\_lname | VARCHAR(20) |  |
| Отчество | em\_patronymic | VARCHAR(20) | обязательное поле |
| Серия и номер паспорта | em\_passport | CHAR(10) | обязательное уникальное поле |
| Когда выдан паспорт | em\_date | DATE | обязательное поле |
| Кем выдан паспорт | em\_given | VARCHAR(40) | обязательное поле |
| Дата рождения | em\_born | DATE | обязательное поле |
| Пол | em\_gender | CHAR(1) | обязательное поле, “м“ или “ж“ |
| ИНН | em\_inn | CHAR(12) | обязательное уникальное поле |
| СНИЛС | em\_snils | CHAR(14) | обязательное уникальное поле |
| Адрес | em\_address | VARCHAR(60) | обязательное поле |
| Должность | em\_post | VARCHAR(20) | внешний ключ к (Posts) |
| Занят/не занят на данный момент | em\_status | VARCHAR(8) | многозначное поле, “занят” или “не занят”, значение по умолчанию – NULL |

Поле “Занят/не занят на данный момент” будет вычисляться с помощью триггера через отношение “Выполнение заказов”(будет изменяться только у курьеров).

Таблица 10. Схема отношения “Телефоны сотрудников”(Emp\_contacts)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор сотрудника | ec\_id | NUMERIC(3) | внешний ключ к (Employees) |
| Тип телефона | ec\_type | VARCHAR(20) | внешний ключ к (Contacts) |
| Номер телефона | ec\_phone | VARCHAR(60) | обязательное поле |

Таблица 11. Схема отношения “Телефоны”(Contacts)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название типа телефона | cont\_name | VARCHAR(20) | **первичный ключ** |

Таблица 12. Схема отношения “Должности”(Posts)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название должности | p\_post | VARCHAR(20) | **первичный ключ** |
| Оклад | p\_salary | NUMERIC(6) | обязательное поле, >12792 |

Таблица 13. Схема отношения “Рестораны”(Restaurants)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор ресторана | r\_id | NUMERIC(3) | **первичный ключ** |
| Название | r\_name | VARCHAR(30) | обязательное поле |
| Минимальный заказ | r\_min | NUMERIC(3) | обязательное поле, >=0 |
| Время доставки | r\_time | INTERVAL | обязательное поле |
| Рейтинг | r\_rating | NUMERIC(4, 2) | обязательное поле, [0; 10] |

Таблица 14. Схема отношения “Кухни”(Causines)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название вида кухни | сaus\_name | VARCHAR(20) | **первичный ключ** |

Таблица 15. Схема отношения “Статусы заказов”(Statuses)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название статуса | stat\_name | VARCHAR(15) | **первичный ключ** |

Таблица 16. Схема отношения “Выполнение заказов”(Stat\_orders)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор заказа | so\_id | NUMERIC(5) | внешний ключ к (Orders) |
| Статус | so\_status | VARCHAR(15) | внешний ключ к (Statuses) |
| Время обновления статуса | so\_stat\_tm | TIME | обязательное поле, значение по умолчанию - текущее время |
| Идентификатор сотрудника (сборщик) | so\_emp\_s | NUMERIC(3) | внешний ключ к (Employees) |
| Идентификатор сотрудника (курьер) | so\_emp\_c | NUMERIC(3) | внешний ключ к (Employees) |
| Стоимость доставки | so\_delivery | NUMERIC(5) | обязательное поле, >=0 |

Статус заказа будет изменяться с помощью триггера. При установке значения “Передан курьеру” система найдет свободного курьера и изменит его статус на “занят”.

Сумму заказа и вес заказа из отношения “Заказы” уберем, так как эти поля являются вычислимыми.

Таблица 17. Схема отношения “Заказы”(Orders)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер заказа | ord\_id | NUMERIC(5) | **первичный ключ** |
| Дата совершения заказа | ord\_date | DATE | обязательное поле |
| Количество персон | ord\_persons | NUMERIC(2) | обязательное поле, >0 |
| Адрес доставки | ord\_address | VARCHAR(60) | обязательное поле |
| Идентификатор клиента | ord\_client | NUMERIC(5) | внешний ключ к (Clients) |

Таблица 18. Схема отношения “Блюда”(Dishes)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор блюда | d\_id | NUMERIC(5) | **первичный ключ** |
| Название | d\_name | VARCHAR(30) | обязательное поле |
| Вес порции | d\_weight | NUMERIC(4) | обязательное поле, >0 |
| Тип блюда | d\_type | VARCHAR(20) | внешний ключ к (Dish\_types) |
| Диета | d\_diet | VARCHAR(20) | внешний ключ к (Diets) |
| Вид кухни | d\_сaus | VARCHAR(20) | внешний ключ к (Causines) |
| Калории на 100 гр. | d\_cal | NUMERIC(4) | обязательное поле, >=0 |
| Белки на 100 гр | d\_prot | NUMERIC(4) | обязательное поле, >=0 |
| Жиры на 100 гр | d\_fats | NUMERIC(4) | обязательное поле, >=0 |
| Углеводы на 100 гр | d\_carb | NUMERIC(4) | обязательное поле, >=0 |
| Состав | d\_ingred | VARCHAR(200) | обязательное поле |
| Срок годности | d\_shelflife | VARCHAR(10) | обязательное поле |
| Скидка | d\_discount | NUMERIC(3) | обязательное поле, [0; 100] |
| Рейтинг | d\_rating | NUMERIC(4,2) | обязательное поле, [0; 10] |

Таблица 19. Схема отношения “Типы блюд”(Dish\_types)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название типа блюда | t\_name | VARCHAR(20) | **первичный ключ** |

Таблица 20. Схема отношения “Диеты”(Diets)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название типа диеты | td\_id | VARCHAR(20) | **первичный ключ** |

Таблица 21. Схема отношения “Производство”(Production)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор | prod\_id | NUMERIC(5) | **первичный ключ** |
| Идентификатор блюда | prod\_dish | NUMERIC(5) | внешний ключ к (Dishes) |
| Идентификатор ресторана | prod\_rest | NUMERIC(3) | внешний ключ к (Restaurants) |
| Стоимость 100 гр. | prod\_price | NUMERIC(4) | обязательное поле, >0 |

Таблица 22. Схема отношения “Компоненты заказа”(Order\_items)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор блюда | i\_dish | NUMERIC(5) | внешний ключ к (Production) |
| Идентификатор заказа | i\_order | NUMERIC(5) | внешний ключ к (Orders) |
| Вес блюда(полученного по факту) | i\_weight | NUMERIC(4) | обязательное поле, (0; 500] |
| Количество порций | i\_num | NUMERIC(2) | обязательное поле, [1; 10] |

**Схема базы данных после нормализации**

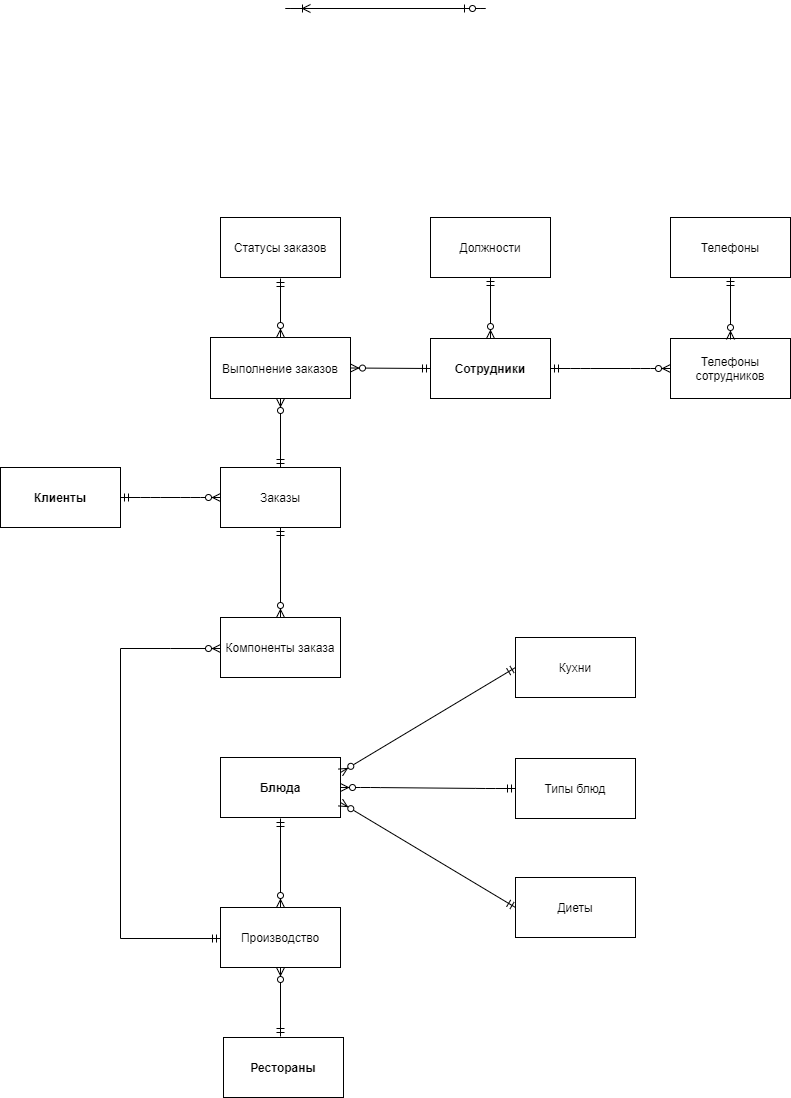
****

Рис. 3.Схема РБД

**Определение дополнительных ограничений целостности**

1. После очередного заказа к рейтингу клиента добавляется 0.05, при этом рейтинг не может быть больше 10.
2. В таблице “Компоненты заказа” комбинация “номер заказа”-”идентификатор блюда” должна быть уникальной.

**Описание групп пользователей и прав доступа**

Опишем для каждой группы пользователей права доступа к каждой таблице с учётом обозначений:

* i (insert) – добавление данных
* s (select) – чтение данных
* u (update) – модификация данных
* d(delete) – удаление данных

Таблица 23. Права доступа к таблицам

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Таблицы | Группы пользователей (роли) | | | | | | |
| Клиент | Директор | Зам. директора | Курьер | Сборщик | Отдел кадров | Менеджер |
| Клиенты |  |  |  |  |  |  | sd |
| Заказы |  |  |  | s | s |  | isud |
| Выполнение заказов |  |  |  | su | su |  | isud |
| Статусы заказов | s |  |  | s | s |  | isud |
| Сотрудники |  | s | s |  |  | isud |  |
| Должности |  | s |  |  |  | isud |  |
| Телефоны сотрудников |  | s |  |  |  | isud |  |
| Телефоны |  | s | s |  |  | isud |  |
| Компоненты заказов | iud |  |  |  | s |  |  |
| Блюда | s |  | isud |  |  |  |  |
| Кухни | s |  | isud |  |  |  |  |
| Типы блюд | s |  | isud |  |  |  |  |
| Диеты | s |  | isud |  |  |  |  |
| Производство |  |  |  |  |  |  | isud |
| Рестораны | s | s | isud |  |  |  |  |

**Реализация проекта базы данных всё новое**

**Создание таблиц**

1. Отношение contacts

CREATE TABLE contacts (

cont\_name VARCHAR(20) CONSTRAINT contacts\_pk PRIMARY KEY

);

1. Отношение posts

CREATE TABLE posts (

p\_post VARCHAR(20) CONSTRAINT posts\_pk PRIMARY KEY,

p\_salary NUMERIC(6) NOT NULL CONSTRAINT p\_salary\_check CHECK(p\_salary > 12792)

);

1. Отношение causines

CREATE TABLE causines (

caus\_name VARCHAR(20) CONSTRAINT causines\_pk PRIMARY KEY

);

1. Отношение statuses

CREATE TABLE statuses (

stat\_name VARCHAR(20) CONSTRAINT statuses\_pk PRIMARY KEY

);

1. Отношение dish\_types

CREATE TABLE dish\_types (

t\_name VARCHAR(20) CONSTRAINT dish\_types\_pk PRIMARY KEY

);

1. Отношение diets

CREATE TABLE diets (

td\_name VARCHAR(20) CONSTRAINT diets\_pk PRIMARY KEY

);

1. Отношение clients

CREATE TABLE clients (

cl\_id NUMERIC(5) CONSTRAINT clients\_pk PRIMARY KEY,

cl\_fname VARCHAR(20) NOT NULL,

cl\_lname VARCHAR(20) NOT NULL,

cl\_patronymic VARCHAR(20),

cl\_born DATE NOT NULL,

cl\_phone VARCHAR(30) NOT NULL,

cl\_address VARCHAR(60),

cl\_rating NUMERIC(4,2) DEFAULT 0 CONSTRAINT rating\_check CHECK(cl\_rating >= 0 AND cl\_rating <= 10)

);

1. Отношение employees

CREATE TABLE employees (

em\_id NUMERIC(3) CONSTRAINT employees\_pk PRIMARY KEY,

em\_fname VARCHAR(20) NOT NULL,

em\_lname VARCHAR(20) NOT NULL,

em\_patronymic VARCHAR(20),

em\_passport CHAR(10) NOT NULL UNIQUE,

em\_date DATE NOT NULL,

em\_given VARCHAR(40) NOT NULL,

em\_born DATE NOT NULL,

em\_gender CHAR(1) NOT NULL CONSTRAINT em\_gender\_check CHECK(em\_gender IN('м', 'ж')),

em\_inn CHAR(12) NOT NULL UNIQUE,

em\_snils CHAR(14) NOT NULL UNIQUE,

em\_address VARCHAR(60) NOT NULL,

em\_post VARCHAR(20) NOT NULL CONSTRAINT posts\_fk REFERENCES posts,

em\_status VARCHAR(8) DEFAULT NULL CONSTRAINT em\_status\_check CHECK(em\_status IN('занят', 'не занят'))

);

1. Отношение emp\_contacts

CREATE TABLE emp\_contacts (

ec\_id NUMERIC(3) NOT NULL CONSTRAINT employees\_fk REFERENCES employees,

ec\_type VARCHAR(20) NOT NULL CONSTRAINT contacts\_fk REFERENCES contacts,

ec\_phone VARCHAR(60) NOT NULL

);

1. Отношение restaurants

CREATE TABLE restaurants (

r\_id NUMERIC(3) CONSTRAINT restaurants\_pk PRIMARY KEY,

r\_name VARCHAR(30) NOT NULL,

r\_min NUMERIC(3) NOT NULL CONSTRAINT r\_min\_check CHECK(r\_min >= 0),

r\_time VARCHAR(10) NOT NULL,

r\_rating NUMERIC(4, 2) CONSTRAINT r\_rating\_check CHECK(r\_rating >= 0 AND r\_rating <= 10)

);

1. Отношение orders

CREATE TABLE orders (

ord\_id NUMERIC(5) CONSTRAINT orders\_pk PRIMARY KEY,

ord\_date DATE NOT NULL,

ord\_persons NUMERIC(2) NOT NULL CONSTRAINT ord\_persons\_check CHECK(ord\_persons > 0),

ord\_address VARCHAR(60) NOT NULL,

ord\_client NUMERIC(5) NOT NULL CONSTRAINT clients\_fk REFERENCES clients

);

1. Отношение stat\_orders

CREATE TABLE stat\_orders (

so\_id NUMERIC(5) NOT NULL CONSTRAINT stat\_orders\_fk REFERENCES orders,

so\_status VARCHAR(15) NOT NULL CONSTRAINT statuses\_fk REFERENCES statuses,

so\_stat\_tm TIME NOT NULL DEFAULT current\_time,

so\_emp\_s NUMERIC(3) NOT NULL(не нужно) CONSTRAINT employees\_s\_fk REFERENCES employees,

so\_emp\_c NUMERIC(3) NOT NULL(не нужно) CONSTRAINT employees\_c\_fk REFERENCES employees,

so\_delivery NUMERIC(5) NOT NULL CONSTRAINT so\_delivery CHECK(so\_delivery >= 0)

);

Эти поля изначально должны быть NULL. Клиент сначала кидает продукты в условную “корзину”. Этому моменту соответствует статус “обработка”. Во время обработки ни сборщик, ни курьер человеку ещё не назначены.

1. Отношение dishes

CREATE TABLE dishes (

d\_id NUMERIC(5) CONSTRAINT dishes\_pk PRIMARY KEY,

d\_name VARCHAR(30) NOT NULL,

d\_weight NUMERIC(4) NOT NULL CONSTRAINT d\_weight\_check CHECK(d\_weight > 0),

d\_type VARCHAR(20) NOT NULL CONSTRAINT dish\_types\_fk REFERENCES dish\_types,

d\_diet VARCHAR(20) NOT NULL CONSTRAINT diets\_fk REFERENCES diets,

d\_caus VARCHAR(20) NOT NULL CONSTRAINT causines\_fk REFERENCES causines,

d\_cal NUMERIC(4) NOT NULL CONSTRAINT d\_cal\_check CHECK(d\_cal > 0),

d\_prot NUMERIC(4) NOT NULL CONSTRAINT d\_prot\_check CHECK(d\_prot >= 0),

d\_fats NUMERIC(4) NOT NULL CONSTRAINT d\_fats\_check CHECK(d\_fats >= 0),

d\_carb NUMERIC(4) NOT NULL CONSTRAINT d\_carb\_check CHECK(d\_carb >= 0),

d\_ingred VARCHAR(200) NOT NULL,

d\_shelflife VARCHAR(10) NOT NULL,

d\_discount NUMERIC(3) NOT NULL CONSTRAINT d\_discount\_check CHECK(d\_discount >= 0 AND d\_discount <= 100),

d\_rating NUMERIC(4, 2) NOT NULL CONSTRAINT d\_rating\_check CHECK(d\_rating >= 0 AND d\_rating <= 10)

);

1. Отношение production

CREATE TABLE production (

prod\_id NUMERIC(5) CONSTRAINT production\_pk PRIMARY KEY,

prod\_dish NUMERIC(5) NOT NULL CONSTRAINT dishes\_fk REFERENCES dishes,

prod\_rest NUMERIC(3) NOT NULL CONSTRAINT restaurants\_fk REFERENCES restaurants,

prod\_price NUMERIC(4) NOT NULL CONSTRAINT prod\_price\_check CHECK(prod\_price > 0)

);

1. Отношение order\_items

CREATE TABLE order\_items (

i\_dish NUMERIC(5) NOT NULL CONSTRAINT production\_fk REFERENCES production,

i\_order NUMERIC(5) NOT NULL CONSTRAINT orders\_fk REFERENCES orders,

i\_weight NUMERIC(4) NOT NULL CONSTRAINT i\_weight\_check CHECK(i\_weight > 0 AND i\_weight <= 500),

i\_num NUMERIC(2) NOT NULL CONSTRAINT i\_num\_check CHECK(i\_num >=1 AND i\_num <= 10),

UNIQUE (i\_dish, i\_order)

);

**Начальный состав данных для справочных таблиц**

INSERT INTO causines VALUES ('Итальянская'),

('Французская'), ('Японская'), ('Русская'), ('Американская'), ('Армянская'), ('Грузинская'), ('Испанская'),('Индийская'), ('Креольская')

INSERT INTO statuses VALUES ('Обработка'), ('Сборка'), ('Передан курьеру'),

('Доставлен')

INSERT INTO dish\_types VALUES ('Супы'),('Рыба и морепродукты'),('Блюда из мяса'),('Десерты'),

('Напитки'),('Овощи'),('Гарниры'),('Салаты'), ('Холодные закуски'),('Паста'),('Пицца'), ('Бургеры')

INSERT INTO diets VALUES ('Отсутствует'), ('Веган'), ('Кето'), ('Сыроедение'), ('Палео'), ('Вегетарианство')

INSERT INTO contacts VALUES ('Домашний'), ('Мобильный'), ('Рабочий')

INSERT INTO Posts VALUES ('Директор', 200000 ), ('Замдиректора', 140000),('Курьер', 40000), ('Сборщик', 30000 ), ('Кадровик', 65000 ), ('Менеджер', 85000)

**Создание представлений (готовых запросов)**

1. Компоненты конкретного заказа конкретного (использующего представление) пользователя my\_order\_parts - s

CREATE OR REPLACE VIEW my\_order\_parts

AS SELECT prod\_dish, d\_name, i\_weight, i\_num

FROM order\_items JOIN production ON i\_dish=prod\_id JOIN dishes ON prod\_dish=d\_id

WHERE i\_order IN (SELECT ord\_id

FROM orders

WHERE ord\_client = 1 AND ord\_id IN (SELECT so\_id

FROM stat\_orders

WHERE so\_status = 'Обработка') )

Это представление показывает пользователю, что лежит у него в корзине. Если человек хочет изменить что-то в своей корзине, то он делает это напрямую через таблицу компоненты заказов. Это реализуется в интерфейсе. Нажатие кнопки "Изменить заказ" отправило бы его к компонентам его заказа (пользователь всегда видит только свои заказы и работает только с ними)

1. Номер телефона клиента, его имя и/или адрес work\_ifo - s

CREATE OR REPLACE VIEW work\_ifo

AS SELECT so\_emp\_c, ord\_address, cl\_id, cl\_fname, cl\_lname, cl\_patronymic, cl\_phone, cl\_address

FROM stat\_orders so, clients cl, orders ord

WHERE so\_emp\_c = 3 AND so\_id = 1 AND so.so\_id = ord.ord\_id AND cl.cl\_id = ord.ord\_client;

1. Итоговый заказ конкретного пользователя + столбец с итоговой ценой для каждого блюда final\_order - s

CREATE OR REPLACE VIEW final\_order

AS SELECT d\_name, (1-d.d\_discount\*0.01)\*oi.i\_num\*pr.prod\_price\*oi.i\_weight\*0.01\*(1-cl\_rating\*0.02) цена блюда

FROM clients cl JOIN orders o ON cl\_id=ord\_client JOIN order\_items oi ON ord\_id=i\_order JOIN production pr ON i\_dish=prod\_id

JOIN dishes d ON prod\_dish=d\_id

WHERE i\_order IN (SELECT ord\_id

FROM orders

WHERE ord\_client = 1 AND ord\_id IN (SELECT so\_id

FROM stat\_orders

WHERE so\_status = 'Сборка' OR so\_status = 'Передан курьеру') )

1. Информация о личных контактных данных my\_data - isud

CREATE OR REPLACE VIEW my\_data

AS SELECT \*

FROM clients cl

WHERE cl\_id = 1;

1. Список товаров и ресторанов, из которых можно заказать еду (без подробностей) all\_menu - s

CREATE OR REPLACE VIEW all\_menu

AS SELECT d\_name, d\_weight, d\_type, d\_diet, d\_caus, d\_rating оценка\_блюда, r\_name, r\_rating оценка\_ресторана

FROM restaurants JOIN production ON r\_id=prod\_rest JOIN dishes ON prod\_dish=d\_id

ORDER BY 8 DESC

Таблица 24. Права доступа к представлениям

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Представления | Группы пользователей (роли) | | | | | | |
| Клиент | Директор | Зам. директора | Курьер | Сборщик | Отдел кадров | Менеджер |
| my\_order\_  parts | s |  |  |  | s |  |  |
| work\_ifo |  |  |  | s |  |  | s |
| final\_order | s |  |  | s | s |  |  |
| my\_data | isud |  |  |  |  |  |  |
| all\_menu | s |  |  |  |  |  |  |

**Назначение прав доступа**

В соответствии с таблицами, представленными выше назначим права доступа:

1. Администратору БД (administrator)

GRANT ALL PRIVILEGES ON Employees

TO administrator WITH GRANT OPTION

Аналогично администратору предоставляются права ко всем остальным отношениям

1. Клиенту (client)

GRANT SELECT ON dishes TO client

Подобным образом предоставляются права к указанным таблицам для клиентов

GRANT SELECT, INSERT, DELETE, UPDATE ON my\_order\_parts TO client

Права доступа к компонентам своего заказа для клиента. Аналогичные он получает для представления my\_data.

1. Директору (director)

GRANT SELECT ON Employees TO director

Аналогичные права директор получает к остальным указанным для него таблицам

1. Зам. директору (substitute)

GRANT SELECT, INSERT, DELETE, UPDATE ON restaurants TO substitute

Подобным образом предоставляются права к указанным отношениям для зама

1. Курьеру (courier)

GRANT SELECT, UPDATE ON Stat\_orders TO courier

Подобным образом предоставляются права к указанным отношениям для курьера

1. Сборщику (collector)

GRANT SELECT, UPDATE ON Stat\_orders TO collector

Подобным образом предоставляются права к указанным отношениям для сборщика

1. Отделу кадров (ok\_user)

GRANT SELECT, INSERT, DELETE, UPDATE ON Employees TO ok\_user

Подобным образом предоставляются права к указанным отношениям для отдела кадров

1. Менеджеру (manager)

GRANT SELECT, DELETE ON clients TO manager

Подобным образом предоставляются права к указанным отношениям для менеджера

**Создание триггеров**

1. Триггер, добавляющий новую строку в таблицу “Выполнение заказов” после появления новой строки в таблице “Заказы”.

CREATE OR REPLACE FUNCTION fn\_add\_status() RETURNS TRIGGER AS $$

BEGIN

INSERT INTO stat\_orders VALUES( NEW.ord\_id, 'Обработка', current\_time, NULL, NULL, 500);

RETURN NEW;

END;

$$ LANGUAGE plpgsql

RETURNS NULL ON NULL INPUT;

CREATE TRIGGER tr\_add\_status

AFTER INSERT ON orders

FOR EACH ROW

EXECUTE FUNCTION fn\_add\_status();

1. Триггер, находящий сборщика и свободного курьера. В зависимости от статуса заказа устанавливающий сотруднику статус “занят” или “не занят”. Изменение происходит при изменении строки в таблице “Выполнение заказов”.

CREATE OR REPLACE FUNCTION fn\_add\_emp() RETURNS TRIGGER AS $$

DECLARE

collector NUMERIC(3);

courier NUMERIC(3);

BEGIN

IF (NEW.so\_status = 'Сборка')

THEN

collector := (SELECT em\_id from employees WHERE em\_post = 'Сборщик' LIMIT 1 OFFSET 0);

NEW.so\_emp\_s = collector;

END IF;

IF (NEW.so\_status = 'Передан курьеру')

THEN

courier = (SELECT em\_id from employees WHERE em\_post = 'Курьер' AND em\_status = 'не занят' LIMIT 1 OFFSET 0);

NEW.so\_emp\_c = courier;

UPDATE employees

SET em\_status = 'занят'

WHERE em\_id = courier;

END IF;

IF (NEW.so\_status = 'Доставлен')

THEN

courier = (SELECT so\_emp\_c from stat\_orders so WHERE so.so\_id = NEW.so\_id);

UPDATE employees

SET em\_status = 'не занят'

WHERE em\_id = courier;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql

RETURNS NULL ON NULL INPUT;

CREATE TRIGGER tr\_add\_emp

BEFORE UPDATE ON stat\_orders

FOR EACH ROW

EXECUTE FUNCTION fn\_add\_emp();

При реализации интерфейса было бы предусмотрено, что как только пользователь нажимает кнопку " купить", то это автоматически меняет статус его заказа на "Сборка". Пользователь взаимодействует только с теми заказами, которые находятся в статусе "Обработка". Для остальных он может видеть только итоговый заказ.

Для сборщика ситуация была бы похожей. Как только он собирает заказ, он отмечает это в приложении переводя статус в "Передано курьеру". После этого триггер начинает искать свободного человека

1. Триггер, добавляющий к рейтингу клиента 0,05 после совершения заказа (после установки статуса заказа “Доставлен”).

CREATE OR REPLACE FUNCTION f\_update\_rating() RETURNS TRIGGER AS $$

BEGIN

IF (NEW.so\_status = 'Доставлен')

THEN

UPDATE clients cl

SET cl\_rating = cl\_rating + 0.05

WHERE cl\_rating < 10 and cl.cl\_id =

(select ord\_client from orders o

WHERE NEW.so\_id = o.ord\_id);

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql

RETURNS NULL ON NULL INPUT;

CREATE TRIGGER tr\_update\_rating

AFTER UPDATE ON stat\_orders

FOR EACH ROW

EXECUTE FUNCTION f\_update\_rating();

**Создание индексов**

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей. Приведем примеры создания индексов:

create index ind\_em\_id on employees(em\_id)

create index ind\_r\_id on restaurants(r\_id)

create index ind\_ord\_id on orders(ord\_id)

create index ind\_d\_id on dishes(d\_id)

**Разработка стратегии резервного копирования**

Интенсивность обновления разработанной базы данных низкая, поэтому для обеспечения сохранности вполне достаточно проводить полное резервное копирование БД раз в день (перед окончанием рабочего дня). Для разработанной БД нет необходимости держать сервер включенным круглосуточно, поэтому можно создать соответствующее задание операционной системы, которое будет автоматически запускаться перед выключением сервера.