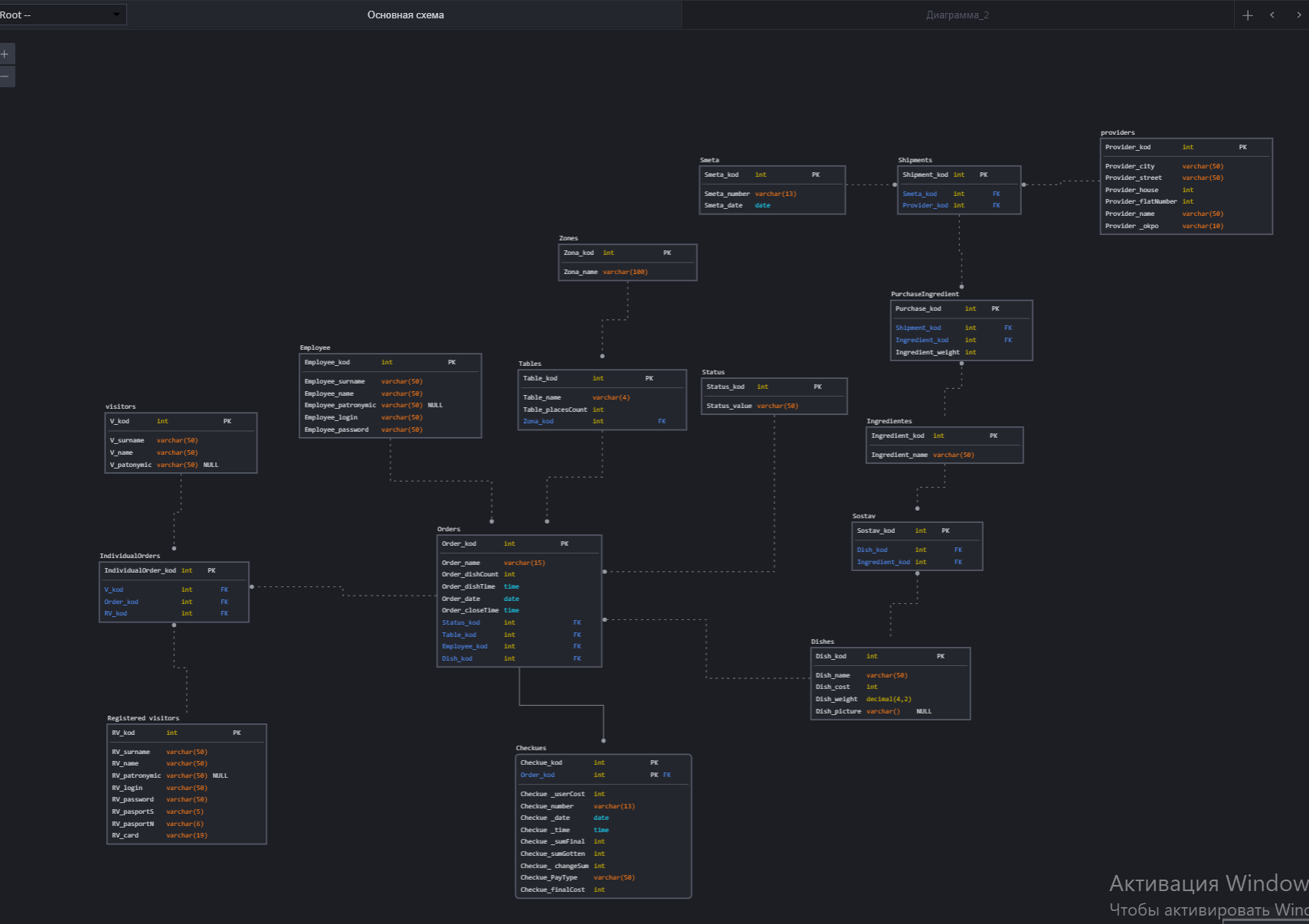
ЭТАПЫ ВЫПОЛНЕНИЯ

**Этапы выполнения работы:**

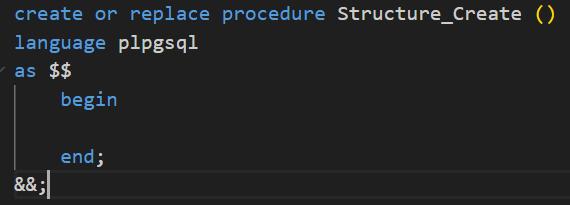
1. Даталогическая модель данных



1. Файл скрипта для разработки структуры;
   1. Создание файла;



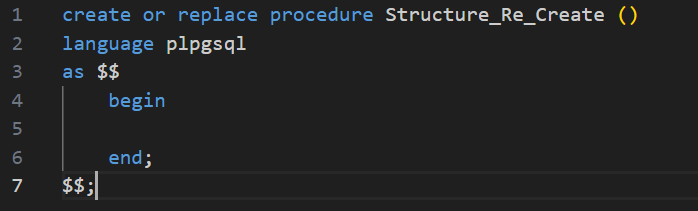
* 1. Заполнение файла хранимой процедурой для создания таблиц БД.



1. Файла скрипта для удаления и перезаписи структуры;
   1. Создание файла;



* 1. Заполнение файла хранимой процедурой для удаления и пересоздания таблиц БД.



| Название поля | Тип данных | Обязательное | Тип поля | Скрипт | Результат |
| --- | --- | --- | --- | --- | --- |
| Employee | | | | | |
| Employee\_surname | Varchar(50) | Да |  | create table if not exists Employee  (  Employee\_kod Serial not null constraint PK\_Employee primary key,  Employee\_surname Varchar(50) not null,  Employee\_name Varchar(50) not null,  Employee\_patronymic Varchar(50) null,  Employee\_login Varchar(50) not null,  Employee\_password Varchar(50) not null  ); |  |
| Employee\_name | Varchar(50) | Да |  |
| Employee\_patronymic | Varchar(50) | нет |  |
| Employee\_login | Varchar(50) | Да |  |
| Employee\_password | Varchar(50) | Да |  |
| Employee\_kod | Int | Да | Суррогатный ключ |
| Registered visitors | | | | | |
| RV\_surname | Varchar(50) | Да |  | create table if not exists Registered\_Visitors  (  RV\_surname Varchar(50) not null,  RV\_name Varchar(50) not null,  RV\_patronymic Varchar(50) null,  RV\_login Varchar(50) not null,  RV\_password Varchar(50) not null,  RV\_pasportS Varchar(5) not null,  RV\_pasportN Varchar(6) not null,  RV\_card Varchar(19) not null,  RV\_kod Serial not null constraint PK\_Registered\_Visitors primary key  ); |  |
| RV\_name | Varchar(50) | Да |  |
| RV\_patronymic | Varchar(50) | Нет |  |
| RV\_login | Varchar(50) | Да |  |
| RV\_password | Varchar(50) | Да |  |
| RV\_pasportS | Varchar(5) | Да |  |
| RV\_pasportN | Varchar(6) | Да |  |
| RV\_card | Varchar(19) | Да |  |
| RV\_kod | Int | Да | Суррогатный ключ |
| visitors | | | | | |
| V\_surname | Varchar(50) | Да |  | create table if not exists Visitors  (  V\_surname Varchar(50) not null,  V\_name Varchar(50) not null,  V\_patronymic Varchar(50) null,  V\_kod Serial not null constraint PK\_Visitors primary key  ); |  |
| V\_name | Varchar(50) | Да |  |
| V\_patronymic | Varchar(50) | нет |  |
| V\_kod | Int | Да | Суррогатный ключ |
| Orders | | | | | |
| Order\_name | Varchar(15) | Да |  | create table if not exists Orders  (  Order\_name Varchar(15) not null,  Order\_dishCount Int not null,  Order\_dishTime Time not null,  Order\_date Date not null,  Order\_closeTime Time not null,  Order\_kod Serial not null constraint PK\_Orders primary key,  Table\_kod Int not null references Wood\_tables (Table\_kod),  Employee\_kod Int not null references Employee (Employee\_kod),  Status\_kod Int not null references Status (Status\_kod)  ); |  |
| Order\_dishCount | Int | Да |  |
| Order\_dishTime | Time | Да |  |
| Order\_date | Date | Да |  |
| Order\_closeTime | Time | Да |  |
| Order\_kod | Int | Да | Суррогатный ключ |
| Table\_kod | Int | Да | Внешний ключ |
| Employee\_kod | Int | Да | Внешний ключ |
| Status\_kod | Int | Да | Внешний ключ |
| Checkues | | | | | |
| Checkue \_userCost | Int | Да |  | create table if not exists Checkues  (  heckue\_userCost Int not null,  Checkue\_number Varchar(13) not null,  Checkue\_date Date not null,  Checkue\_time Time not null,  Checkue\_sumFinal Int not null,  Checkue\_sumGotten Int not null,  Checkue\_changeSum Int not null,  Checkue\_PayType Varchar(50) not null,  Checkue\_finalCost Int not null,  Checkue\_kod Serial not null constraint PK\_Checkues primary key,  Order\_kod Int not null references Orders(Order\_kod)  ); |  |
| Checkue\_number | Varchar(13) | Да |  |
| Checkue \_date | Date | Да |  |
| Checkue \_time | Time | Да |  |
| Checkue \_sumFinal | Int | Да |  |
| Checkue\_sumGotten | Int | Да |  |
| Checkue\_ changeSum | Int | Да |  |
| Checkue\_PayType | Varchar(50) | Да |  |
| Checkue\_finalCost | Int | Да |  |
| Checkue\_kod | Int | Да | Суррогатный ключ |
| Order\_kod | Int | Да | Внешний ключ |
| Providers | | | | | |
| Provider\_city | Varchar(50) | Да |  | create table if not exists Providers  (  Provider\_city Varchar(50) not null,  Provider\_street Varchar(50) not null,  Provider\_house Int not null,  Provider\_flatNumber Int not null,  Provider\_name Varchar(50) not null,  Provider\_okpo Varchar(10) not null,  Provider\_kod Serial not null constraint PK\_Providers primary key  ); |  |
| Provider\_street | Varchar(50) | Да |  |
| Provider\_house | Int | Да |  |
| Provider\_flatNumber | Int | Да |  |
| Provider\_name | Varchar(50) | Да |  |
| Provider \_okpo | Varchar(10) | Да |  |
| Provider\_kod | Int | Да | Суррогатный ключ |
| Smeta | | | | | |
| Smeta\_date | Date | Да |  | create table if not exists Smeta  (  Smeta\_date Date not null,  Smeta\_number Varchar(13),  Smeta\_kod Serial not null constraint PK\_Smeta primary key  ); |  |
| Smeta\_number | Varchar(13) | Да |  |
| Smeta\_kod | Int | Да | Суррогатный ключ |
| Wood\_tables | | | | | |
| Table\_placesCount | Int | Да |  | create table if not exists Wood\_tables  (  Table\_placesCount Int not null,  Wood\_table\_name Varchar(4) not null,  Table\_kod Serial not null constraint PK\_Wood\_Tables primary key,  Zona\_kod Int not null references Zones(Zona\_kod)  ); |  |
| Wood\_table\_name | Varchar(4) | Да |  |
| Table\_kod | Int | Да | Суррогатный ключ |
| Zona\_kod | Int | Да | Внешний ключ |
| Dishes | | | | | |
| Dish\_name | Varchar(50) | Да |  | create table if not exists Dishes  (  Dish\_name Varchar(50) not null,  Dish\_cost Int not null,  Dish\_weight Decimal(4,2) not null,  Dish\_picture Varchar(1) not null,  Dish\_kod Serial not null constraint PK\_Dishes primary key  ); |  |
| Dish\_cost | Int | Да |  |
| Dish\_weight | Decimal(4,2) | Да |  |
| Dish\_picture | Varchar() | Да |  |
| Dish\_kod | Int | Да | Суррогатный ключ |
| Status | | | | | |
| Status\_value | Varchar(50) | Да |  | create table if not exists Status  (  Status\_value Varchar(50) not null,  Status\_kod Serial not null constraint PK\_Status primary key  ); |  |
| Status\_kod | Int | Да | Суррогатный ключ |
| Zones | | | | | |
| Zona\_name | Varchar(100) | Да |  | create table if not exists Zones  (  Zona\_name Varchar(100) not null,  Zona\_kod Serial not null constraint PK\_Zones primary key  ); |  |
| Zona\_kod | Int | Да | Суррогатный ключ |
| Ingredientes | | | | | |
| Ingredient\_name | Varchar(50) | Да |  | create table if not exists Ingredientes  (  Ingredient\_name Varchar(50) not null,  Ingredient\_kod Serial not null constraint PK\_Ingredientes primary key  ); |  |
| Ingredient\_kod | Int | Да | Суррогатный ключ |
| IndividualOrders | | | | | |
| IndividualOrder\_kod | Int | Да | Суррогатный ключ | create table if not exists IndividualOrders  (  IndividualOrder\_kod Serial not null constraint PK\_IndividualOrders primary key,  Order\_kod Int not null references Orders(Order\_kod),  V\_kod Int not null references Visitors (V\_kod),  RV\_kod Int not null references Registered\_Visitors(RV\_kod)  ); |  |
| Order\_kod | Int | Да | Внешний ключ |
| V\_kod | Int | Да | Внешний ключ |
| RV\_kod | Int | Да | Внешний ключ |
| PurchaseIngredient | | | | | |
| Purchase\_kod | Int | Да | Суррогатный ключ | create table if not exists PurchaseIngredient  (  Purchase\_kod Serial not null constraint PK\_PurchaseIngredient primary key,  Ingredient\_weight Int not null,  Shipment\_kod Int not null references Shipments (Shipment\_kod),  Ingredient\_kod Int not null references Ingredientes(Ingredient\_kod)  ); |  |
| Ingredient\_weight | Int | Да |  |
| Shipment\_kod | Int | Да | Внешний ключ |
| Ingredient\_kod | Int | Да | Внешний ключ |
| Shipments | | | | | |
| Shipment\_kod | Int | Да | Суррогатный ключ | create table if not exists Shipments  (  Shipment\_kod Serial not null constraint PK\_Shipments primary key,  Smeta\_kod Int not null references Smeta (Smeta\_kod),  Provider\_kod Int not null references Providers(Provider\_kod)  ); |  |
| Smeta\_kod | Int | Да | Внешний ключ |
| Provider\_kod | Int | Да | Внешний ключ |
| Sostav | | | | | |
| Sostav\_kod | Int | Да | Суррогатный ключ | create table if not exists Sostav  (  Sostav\_kod Serial not null constraint PK\_Sostav primary key,  Ingredient\_kod Int not null references Ingredientes(Ingredient\_kod),  Dish\_kod Int not null references Dishes(Dish\_kod)  ); |  |
| Ingredient\_kod | Int | Да | Внешний ключ |
| Dish\_kod | Int | Да | Внешний ключ |

* 1. **Скрипт создания индексов.**

| Название поля | Тип поля | Обоснование индекса | Скрипт | | | Результат | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Employee | | | | | | | | | Varchar(50) | Да |
| Employee\_surname |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных | create index if not exists index\_fio\_employee on employee (employee\_name, employee\_surname, employee\_patronymic); | | |  | |
| Employee\_name |  |
| Employee\_patronymic |  |
| Employee\_login | Унакальное | Необходим, так как позволит быстро произвести аутентификацию пользователя | create index if not exists index\_login\_password\_employee on employee (employee\_password, employee\_login, employee\_patronymic); | | |
| Employee\_password |  |
| Employee\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ |  | | |
| Registered visitors | | | | | | | | | Varchar(50) | Да |
| RV\_surname |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных | create index if not exists index\_fio\_rv on registered\_visitors (rv\_name, rv\_surname, rvpatronymic); | | |  | |
| RV\_name |  |
| RV\_patronymic |  |
| RV\_login |  | Необходим, так как позволит быстро произвести аутентификацию пользователя | create index if not exists index\_login\_password\_rv on registered\_visitors (rv\_login, rv\_password); | | |
| RV\_password |  |
| RV\_pasportS |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных | create index if not exists index\_pasportSN\_rv on registered\_visitors (rv\_pasportS, rv\_pasportN); | | |
| RV\_pasportN |  |
| RV\_card | Уникальное | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_card\_rv on registered\_visitors (rv\_card); | | |
| RV\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_rv on registered\_visitors (rv\_kod); | | |
| visitors | | | | | | | | | Varchar(50) | Да |
| V\_surname |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных | create index if not exists index\_fio\_visitor on visitors(v\_surname, v\_name, v\_patronymic); | | |  | |
| V\_name |  |
| V\_patronymic |  |
| V\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_visitor on visitors(v\_kod); | | |
| Orders | | | | | | | | | Int | Да |
| Order\_name | уникальное | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_name\_order on orders (order\_name); | | |  | |
| Order\_dishCount |  |  |  | | |
| Order\_dishTime |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных |  | | |
| Order\_date |  | create index if not exists index\_date\_time\_order on orders (order\_dishtime, order\_date, order\_closetime); | | |
| Order\_closeTime |  |
| Order\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_order on orders (order\_kod); | | |
| Table\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Employee\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Status\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Checkues | | | | | | | | | Varchar(13) | Да |
| Checkue \_userCost |  |  |  | | |  | |
| Checkue\_number | Уникальое | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_number\_checkue on checkues (checkue\_number); | | |
| Checkue \_date |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных | create index if not exists index\_date\_time\_checkue on checkues (checkue\_date, checkue\_time); | | |
| Checkue \_time |  |
| Checkue \_sumFinal |  |  |  | | |
| Checkue\_sumGotten |  |  |  | | |
| Checkue\_ changeSum |  |  |  | | |
| Checkue\_PayType |  |  |  | | |
| Checkue\_finalCost |  |  |  | | |
| Checkue\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_checkue on checkues (checkue\_kod); | | |
| Order\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Providers | | | | | | | | | Varchar(50) | Да |
| Provider\_city |  | Необходим, так как поля уникальны и по данным значениям можно организовать поиск данных | create index if not exists index\_location\_provider on providers (provider\_city, provider\_street, provider\_house, provider\_flatnumber); | | |  | |
| Provider\_street |  |
| Provider\_house |  |
| Provider\_flatNumber |  |
| Provider\_name | Уникальое | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_name\_provider on providers (provider\_name); | | |
| Provider \_okpo | Уникальое | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_okpo\_provider on providers (provider\_okpo); | | |
| Provider\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_provider on providers (provider\_kod); | | |
| Smeta | | | | | | | | | Varchar(13) | Да |
| Smeta\_date |  |  |  | | |  | |
| Smeta\_number | Уникалькое | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_number\_smeta on smeta (smeta\_number); | | |
| Smeta\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_smeta on smeta (smeta\_kod); | | |
| Wood\_tables | | | | | | | | | Varchar(4) | Да |
| Table\_placesCount |  |  |  | | |  | |
| Wood\_table\_name | Уникальое | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_name\_table on wood\_tables(wood\_table\_name); | | |
| Table\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_table on wood\_tables (table\_kod); | | |
| Zona\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Dishes | | | | | | | |
| Dish\_name | Уникальое | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_name\_dish on dishes(dish\_name); | | |  | |
| Dish\_cost |  |  |  | | |
| Dish\_weight |  |  |  | | |
| Dish\_picture |  |  |  | | |
| Dish\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_dish on dishes(dish\_kod); | | |  | |
| Status | | | | | | | | | Int | Да |
| Status\_value | Уникальное | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_value\_status on status(status\_value); | | |  | |
| Status\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_status on status(status\_kod); | | |
| Zones | | | | | | | | | Int | Да |
| Zona\_name | Уникальное | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_kod\_zones on zones(zona\_kod); | | |  | |
| Zona\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_name\_zones on zones(zona\_kod); | | |
| Ingredientes | | | | | | | | | Int | Да |
| Ingredient\_name | Уникально | Необходим, так как поле уникальное и по данному значению можно организовать поиск данных | create index if not exists index\_kod\_ingredient on ingredientes(ingredient\_kod);  create index if not exists index\_name\_ingredient on ingredientes(ingredient\_name); | | |  | |
| Ingredient\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ |
| IndividualOrders | | | | | | | | | Int | Да |
| IndividualOrder\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_IndividualOrders on IndividualOrders(IndividualOrder\_kod); | | |  | |
| Order\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| V\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| RV\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| PurchaseIngredient | | | | | | | | | Int | Да |
| Purchase\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_ PurchaseIngredient on PurchaseIngredient (Purchase\_kod); | | |  | |
| Ingredient\_weight |  |  |  | | |
| Shipment\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Ingredient\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Shipments | | | | | | | | | Int | Да |
| Shipment\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | create index if not exists index\_kod\_Shipments on Shipments(Shipment\_kod); | | |  | |
| Smeta\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Provider\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом |  | | |
| Sostav | | | | | | | | |  |  |
| Sostav\_kod | Суррогатный ключ | Необходим, так как суррогатный ключ | | create index if not exists index\_kod\_sostav on Sostav(Sostav \_kod); |  | |
| Dish\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом | |  |  | |
| Ingredient\_kod | Внешний ключ | Нет необходимости так как, поле внешнего ключа и ключ родительской таблицы уже с индексом | |  |

1. Распределение прав доступа к таблицам БД;

Таблица 3 – Права доступа к таблицам БД

| Роли | | официант | Зарегистрированный посетитель | Дополнительный гость | Шеф | Администратор |
| --- | --- | --- | --- | --- | --- | --- |
| Название объекта | Функции |
| Employee | Выборка | X |  |  | X | Х |
| Добавление |  |  |  |  | Х |
| Изменение | X |  |  | X | Х |
| Удаление |  |  |  |  | Х |
| Registered visitors | Выборка |  | Х |  |  | Х |
| Добавление | X |  |  |  | Х |
| Изменение | X | X |  |  | Х |
| Удаление |  |  |  |  | Х |
| visitors | Выборка |  |  | Х |  | X |
| Добавление | X |  |  |  | X |
| Изменение | X |  | X |  | X |
| Удаление |  |  |  |  | X |
| Checkues | Выборка | X | X | X | X | Х |
| Добавление | X |  |  |  |  |
| Изменение | X |  |  |  |  |
| Удаление | X |  |  |  |  |
| Providers | Выборка |  |  |  | X | Х |
| Добавление |  |  |  | X | Х |
| Изменение |  |  |  | X | Х |
| Удаление |  |  |  | X | Х |
| Smeta | Выборка |  |  |  | X | Х |
| Добавление |  |  |  | X | X |
| Изменение |  |  |  | X | X |
| Удаление |  |  |  | X | Х |
| Dishes | Выборка | X | X | X | X |  |
| Добавление |  |  |  | X |  |
| Изменение |  |  |  | X |  |
| Удаление |  |  |  | X |  |
| Zones | Выборка | X | X | X |  | X |
| Добавление |  |  |  |  | X |
| Изменение |  |  |  |  | X |
| Удаление |  |  |  |  | X |
| Ingredientes | Выборка |  |  |  | Х |  |
| Добавление |  |  |  | Х |  |
| Изменение |  |  |  | Х |  |
| Удаление |  |  |  | Х |  |
| IndividualOrders | Выборка | X | Х | Х |  | Х |
| Добавление | X |  |  |  |  |
| Изменение | X |  |  |  |  |
| Удаление | X |  |  |  |  |
| PurchaseIngredient | Выборка |  |  |  | X | Х |
| Добавление |  |  |  | X |  |
| Изменение |  |  |  | X |  |
| Удаление |  |  |  | X |  |
| Shipments | Выборка |  |  |  | X | Х |
| Добавление |  |  |  | X |  |
| Изменение |  |  |  | X |  |
| Удаление |  |  |  | X |  |
| Sostav | Выборка | X | X | X | Х | Х |
| Добавление |  |  |  | Х |  |
| Изменение |  |  |  | Х |  |
| Удаление |  |  |  | X |  |
| orders | Выборка | X | Х | Х | X | Х |
| Добавление | X |  |  |  |  |
| Изменение | X |  |  |  |  |
| Удаление | X |  |  |  |  |

1. Выдача прав доступа к таблицам БД;

Таблица 4 – Реализация разграничения прав доступа

| Название роли | Название объекта | Функция | Скрипт |
| --- | --- | --- | --- |
| res\_waiter | Registered visitors | Insert, Update | grant insert, update on Registered\_visitors to res\_waiter;  grant usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_waiter; |
| visitors | grant insert, update on visitors to res\_waiter;  grant usage, select on sequence visitors\_v\_kod\_seq to res\_waiter; |
| Employee | Select, Update | grant select, update on Employee to res\_waiter; |
| zones | Select | grant select on zones to res\_waiter; |
| dishes | grant select on dishes to res\_waiter; |
| sostav | grant select on sostav to res\_waiter; |
| orders | Select, Insert, Update, Delete | grant select, insert, update, delete on orders to res\_waiter;  grant usage, select on sequence orders\_order\_kod\_seq to res\_waiter; |
| IndividualOrders | grant select, insert, update, delete on IndividualOrders to res\_waiter;  grant usage, select on sequence individualorders\_individualorder\_kod\_seq to res\_waiter; |
| Checkues | grant select, insert, update, delete on Checkues to res\_waiter;  grant usage, select on sequence checkues\_checkue\_kod\_seq to res\_waiter; |
| res\_guest | Registered\_visitors | Select, Update | grant select, update on Registered\_visitors to res\_guest; |
| orders | Select | grant select on orders to res\_guest; |
| IndividualOrders | grant select on IndividualOrders to res\_guest; |
| Checkues | grant select on Checkues to res\_guest; |
| dishes | grant select on dishes to res\_guest; |
| rl\_ dguest | visitors | Select, Update | res\_dguest |
| orders | Select | grant select on orders to res\_dguest; |
| IndividualOrders | grant select on IndividualOrders to res\_dguest; |
| Checkues | grant select on Checkues to res\_dguest; |
| dishes | grant select on dishes to res\_dguest; |
| res\_chef | Employee | Select, Update | grant select, update on employee to res\_chef; |
| PurchaseIngredient | Select, Insert, Update, Delete | grant select, insert, update, delete on PurchaseIngredient to res\_chef;  grant usage, select on sequence purchaseingredient\_purchase\_kod\_seq to res\_chef; |
| Shipments | grant select, insert, update, delete on Shipments to res\_chef;  grant usage, select on sequence shipments\_shipment\_kod\_seq to res\_chef; |
| sostav | grant select, insert, update, delete on sostav to res\_chef;  grant usage, select on sequence sostav\_sostav\_kod\_seq to res\_chef; |
| Ingredientes | grant select, insert, update, delete on Ingredientes to res\_chef;  grant usage, select on sequence ingredientes\_ingredient\_kod\_seq to res\_chef; |
| Providers | grant select, insert, update, delete on Providers to res\_chef;  grant usage, select on sequence providers\_provider\_kod\_seq to res\_chef; |
| Smeta | grant select, insert, update, delete on Smeta to res\_chef;  grant usage, select on sequence smeta\_smeta\_kod\_seq to res\_chef; |
| Dishes | grant select, insert, update, delete on Dishes to res\_chef;  grant usage, select on sequence dishes\_dish\_kod\_seq to res\_chef; |
| orders | select | grant select on orders to res\_chef; |
| checkues | grant select on checkues to res\_chef; |
| res\_administrator | Employee | Select, Insert, Update, Delete | grant select, insert, update, delete on Employee to res\_chef;  grant usage, select on sequence employee\_employee\_kod\_seq to res\_chef; |
| Registered\_visitors | grant select, insert, update, delete on Registered\_visitors to res\_chef;  grant usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_chef; |
| visitors | grant select, insert, update, delete on visitors to res\_chef;  grant usage, select on sequence visitors\_v\_kod\_seq to res\_chef; |
| Providers | grant select, insert, update, delete on Providers to res\_chef;  grant usage, select on sequence providers\_provider\_kod\_seq to res\_chef; |
| Smeta | grant select, insert, update, delete on Smeta to res\_chef;  grant usage, select on sequence smeta\_smeta\_kod\_seq to res\_chef; |
| zones | grant select, insert, update, delete on zones to res\_chef;  grant usage, select on sequence zones\_zona\_kod\_seq to res\_chef; |
| IndividualOrders | select | grant select on IndividualOrders to res\_administrator; |
| PurchaseIngredient | grant select on PurchaseIngredient to res\_administrator; |
| Shipments | grant select on Shipments to res\_administrator; |
| orders | grant select on orders to res\_administrator; |

1. Скрипт процедуры на создание таблиц и индексов;

| Скрипт объекта |
| --- |
| create or replace procedure Structure\_Create ()  language plpgsql  as $$      begin    create table if not exists Employee  (      Employee\_kod Serial not null constraint PK\_Employee primary key,      Employee\_surname Varchar(50) not null,      Employee\_name Varchar(50) not null,      Employee\_patronymic Varchar(50) null,      Employee\_login Varchar(50) not null,      Employee\_password Varchar(50) not null  );  create table if not exists Registered\_Visitors  (      RV\_surname Varchar(50) not null,      RV\_name Varchar(50) not null,      RV\_patronymic Varchar(50) null,      RV\_login Varchar(50) not null,      RV\_password Varchar(50) not null,      RV\_pasportS Varchar(5) not null,      RV\_pasportN Varchar(6) not null,      RV\_card Varchar(19) not null,      RV\_kod Serial not null constraint PK\_Registered\_Visitors primary key  );  create table if not exists Visitors  (      V\_surname Varchar(50) not null,      V\_name Varchar(50) not null,      V\_patronymic Varchar(50) null,      V\_kod Serial not null constraint PK\_Visitors primary key  );  create table if not exists Providers  (      Provider\_city Varchar(50) not null,      Provider\_street Varchar(50) not null,      Provider\_house Int not null,      Provider\_flatNumber Int not null,      Provider\_name Varchar(50) not null,      Provider\_okpo Varchar(10) not null,      Provider\_kod Serial not null constraint PK\_Providers primary key  );  create table if not exists Smeta  (      Smeta\_date Date not null,      Smeta\_number Varchar(13),      Smeta\_kod Serial not null constraint PK\_Smeta primary key  );  create table if not exists Dishes  (      Dish\_name Varchar(50) not null,      Dish\_cost Int not null,      Dish\_weight Decimal(4,2) not null,      Dish\_picture Varchar(1) not null,      Dish\_kod Serial not null constraint PK\_Dishes primary key  );  create table if not exists Status  (      Status\_value Varchar(50) not null,      Status\_kod Serial not null constraint PK\_Status primary key  );  create table if not exists Zones  (      Zona\_name Varchar(100) not null,      Zona\_kod Serial not null constraint PK\_Zones primary key  );  create table if not exists Ingredientes  (      Ingredient\_name Varchar(50) not null,      Ingredient\_kod Serial not null constraint PK\_Ingredientes primary key  );  create table if not exists Wood\_tables  (      Table\_placesCount Int not null,      Wood\_table\_name Varchar(4) not null,      Table\_kod Serial not null constraint PK\_Wood\_Tables primary key,      Zona\_kod Int not null references Zones(Zona\_kod)  );  create table if not exists Orders  (      Order\_name Varchar(15) not null,      Order\_dishCount Int not null,      Order\_dishTime Time not null,      Order\_date Date not null,      Order\_closeTime Time not null,      Order\_kod Serial not null constraint PK\_Orders primary key,      Table\_kod Int not null references Wood\_tables (Table\_kod),      Employee\_kod Int not null references Employee (Employee\_kod),      Status\_kod Int not null references Status (Status\_kod)  );  create table if not exists Checkues  (      heckue\_userCost Int not null,      Checkue\_number Varchar(13) not null,      Checkue\_date Date not null,      Checkue\_time Time not null,      Checkue\_sumFinal Int not null,      Checkue\_sumGotten Int not null,      Checkue\_changeSum Int not null,      Checkue\_PayType Varchar(50) not null,      Checkue\_finalCost Int not null,      Checkue\_kod Serial not null constraint PK\_Checkues primary key,      Order\_kod Int not null references Orders(Order\_kod)  );  create table if not exists IndividualOrders  (      IndividualOrder\_kod Serial not null constraint PK\_IndividualOrders primary key,      Order\_kod Int not null references Orders(Order\_kod),      V\_kod Int not null references Visitors (V\_kod),      RV\_kod Int not null references Registered\_Visitors(RV\_kod)  );  create table if not exists Shipments  (  Shipment\_kod Serial not null constraint PK\_Shipments primary key,  Smeta\_kod Int not null references Smeta (Smeta\_kod),  Provider\_kod Int not null references Providers(Provider\_kod)  );  create table if not exists PurchaseIngredient  (  Purchase\_kod Serial not null constraint PK\_PurchaseIngredient primary key,  Ingredient\_weight Int not null,  Shipment\_kod Int not null references Shipments (Shipment\_kod),  Ingredient\_kod Int not null references Ingredientes(Ingredient\_kod)  );  create table if not exists Sostav  (  Sostav\_kod Serial not null constraint PK\_Sostav primary key,  Ingredient\_kod Int not null references Ingredientes(Ingredient\_kod),  Dish\_kod Int not null references Dishes(Dish\_kod)  );  create index if not exists index\_kod\_sostav on Sostav(Sostav\_kod);  create index if not exists index\_kod\_Shipments on Shipments(Shipment\_kod);  create index if not exists index\_kod\_PurchaseIngredient on PurchaseIngredient (Purchase\_kod);  create index if not exists index\_kod\_IndividualOrders on IndividualOrders(IndividualOrder\_kod);  create index if not exists index\_kod\_ingredient on ingredientes(ingredient\_kod);  create index if not exists index\_name\_ingredient on ingredientes(ingredient\_name);  create index if not exists index\_kod\_zones on zones(zona\_kod);  create index if not exists index\_name\_zones on zones(zona\_kod);  create index if not exists index\_kod\_status on status(status\_kod);  create index if not exists index\_name\_dish on dishes(dish\_name);  create index if not exists index\_kod\_dish on dishes(dish\_kod);  create index if not exists index\_name\_table on wood\_tables(wood\_table\_name);  create index if not exists index\_kod\_table on wood\_tables (table\_kod);  create index if not exists index\_number\_smeta on smeta (smeta\_number);  create index if not exists index\_kod\_smeta on smeta (smeta\_kod);  create index if not exists index\_location\_provider on providers (provider\_city, provider\_street, provider\_house, provider\_flatnumber);  create index if not exists index\_name\_provider on providers (provider\_name);  create index if not exists index\_okpo\_provider on providers (provider\_okpo);  create index if not exists index\_kod\_provider on providers (provider\_kod);  create index if not exists index\_number\_checkue on checkues (checkue\_number);  create index if not exists index\_date\_time\_checkue on checkues (checkue\_date, checkue\_time);  create index if not exists index\_kod\_checkue on checkues (checkue\_kod);  create index if not exists index\_name\_order on orders (order\_name);  create index if not exists index\_date\_time\_order on orders (order\_dishtime, order\_date, order\_closetime);  create index if not exists index\_kod\_order on orders (order\_kod);  create index if not exists index\_fio\_visitor on visitors(v\_surname, v\_name, v\_patronymic);  create index if not exists index\_kod\_visitor on visitors(v\_kod);  create index if not exists index\_fio\_rv on registered\_visitors (rv\_name, rv\_surname, rv\_patronymic);  create index if not exists index\_login\_password\_rv on registered\_visitors (rv\_login, rv\_password);  create index if not exists index\_pasportSN\_rv on registered\_visitors (rv\_pasportS, rv\_pasportN);  create index if not exists index\_card\_rv on registered\_visitors (rv\_card);  create index if not exists index\_kod\_rv on registered\_visitors (rv\_kod);  create index if not exists index\_fio\_employee on employee (employee\_name, employee\_surname, employee\_patronymic);  create index if not exists index\_login\_password\_employee on employee (employee\_password, employee\_login, employee\_patronymic);  create index if not exists index\_value\_status on status(status\_value);  grant select on zones to res\_waiter;  grant select on dishes to res\_waiter;  grant select on sostav to res\_waiter;  grant select, update on Employee to res\_waiter;  grant select, insert, update, delete on orders to res\_waiter;  grant usage, select on sequence orders\_order\_kod\_seq to res\_waiter;  grant select, insert, update, delete on IndividualOrders to res\_waiter;  grant usage, select on sequence individualorders\_individualorder\_kod\_seq to res\_waiter;  grant select, insert, update, delete on Checkues to res\_waiter;  grant usage, select on sequence checkues\_checkue\_kod\_seq to res\_waiter;  grant insert, update on visitors to res\_waiter;  grant usage, select on sequence visitors\_v\_kod\_seq to res\_waiter;  grant insert, update on Registered\_visitors to res\_waiter;  grant usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_waiter;    grant select on orders to res\_guest;  grant select on IndividualOrders to res\_guest;  grant select on Checkues to res\_guest;  grant select on dishes to res\_guest;  grant select, update on Registered\_visitors to res\_guest;  grant select on orders to res\_dguest;  grant select on IndividualOrders to res\_dguest;  grant select on Checkues to res\_dguest;  grant select on dishes to res\_dguest;  grant select, update on visitors to res\_dguest;  grant select on orders to res\_chef;  grant select on checkues to res\_chef;  grant select, update on visitors to res\_dguest;  grant select, insert, update, delete on PurchaseIngredient to res\_chef;  grant usage, select on sequence purchaseingredient\_purchase\_kod\_seq to res\_chef;  grant select, insert, update, delete on Shipments to res\_chef;  grant usage, select on sequence shipments\_shipment\_kod\_seq to res\_chef;  grant select, insert, update, delete on sostav to res\_chef;  grant usage, select on sequence sostav\_sostav\_kod\_seq to res\_chef;  grant select, insert, update, delete on Ingredientes to res\_chef;  grant usage, select on sequence ingredientes\_ingredient\_kod\_seq to res\_chef;  grant select, insert, update, delete on Providers to res\_chef;  grant usage, select on sequence providers\_provider\_kod\_seq to res\_chef;  grant select, insert, update, delete on Smeta to res\_chef;  grant usage, select on sequence smeta\_smeta\_kod\_seq to res\_chef;  grant select, insert, update, delete on Dishes to res\_chef;  grant usage, select on sequence dishes\_dish\_kod\_seq to res\_chef;  grant select on IndividualOrders to res\_administrator;  grant select on PurchaseIngredient to res\_administrator;  grant select on Shipments to res\_administrator;  grant select on orders to res\_administrator;  grant select, insert, update, delete on Registered\_visitors to res\_administrator;  grant usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_administrator;  grant select, insert, update, delete on visitors to res\_administrator;  grant usage, select on sequence visitors\_v\_kod\_seq to res\_administrator;  grant select, insert, update, delete on Employee to res\_administrator;  grant usage, select on sequence employee\_employee\_kod\_seq to res\_administrator;  grant select, insert, update, delete on Providers to res\_administrator;  grant usage, select on sequence providers\_provider\_kod\_seq to res\_administrator;  grant select, insert, update, delete on Smeta to res\_administrator;  grant usage, select on sequence smeta\_smeta\_kod\_seq to res\_administrator;  grant select, insert, update, delete on zones to res\_administrator;  grant usage, select on sequence zones\_zona\_kod\_seq to res\_administrator;      end;  &&; |

1. Скрипт процедуры на перезапись структуры БД;

Таблица 6 – Скрипт объекта

| drop or replace procedure Structure\_Re\_drop ()  language plpgsql  as $$      begin  revoke  usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_waiter;  revoke usage, select on sequence orders\_order\_kod\_seq to res\_waiter;  revoke usage, select on sequence individualorders\_individualorder\_kod\_seq to res\_waiter;  revoke usage, select on sequence checkues\_checkue\_kod\_seq to res\_waiter;  revoke usage, select on sequence visitors\_v\_kod\_seq to res\_waiter;  revoke usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_waiter;  revoke usage, select on sequence dishes\_dish\_kod\_seq to res\_chef;  revoke usage, select on sequence purchaseingredient\_purchase\_kod\_seq to res\_chef;  revoke usage, select on sequence shipments\_shipment\_kod\_seq to res\_chef;  revoke usage, select on sequence visitors\_v\_kod\_seq to res\_administrator;  revoke usage, select on sequence registered\_visitors\_rv\_kod\_seq to res\_administrator;  revoke usage, select on sequence employee\_employee\_kod\_seq to res\_administrator;  revoke usage, select on sequence sostav\_sostav\_kod\_seq to res\_chef;  revoke usage, select on sequence ingredientes\_ingredient\_kod\_seq to res\_chef;  revoke usage, select on sequence providers\_provider\_kod\_seq to res\_chef;  revoke usage, select on sequence smeta\_smeta\_kod\_seq to res\_chef;  revoke usage, select on sequence dishes\_dish\_kod\_seq to res\_chef;  revoke usage, select on sequence providers\_provider\_kod\_seq to res\_administrator;  revoke usage, select on sequence zones\_zona\_kod\_seq to res\_administrator;  revoke usage, select on sequence smeta\_smeta\_kod\_seq to res\_administrator;  revoke select on zones to res\_waiter;  revoke select on dishes to res\_waiter;  revoke select on sostav to res\_waiter;  revoke select, update on Employee to res\_waiter;  revoke select, insert, update, delete on orders to res\_waiter;  revoke select, insert, update, delete on IndividualOrders to res\_waiter;  revoke select, insert, update, delete on Checkues to res\_waiter;  revoke insert, update on visitors to res\_waiter;  revoke insert, update on Registered\_visitors to res\_waiter;  revoke select on orders to res\_guest;  revoke select on IndividualOrders to res\_guest;  revoke select on Checkues to res\_guest;  revoke select on dishes to res\_guest;  revoke select, update on Registered\_visitors to res\_guest;  revoke select on orders to res\_dguest;  revoke select on IndividualOrders to res\_dguest;  revoke select on Checkues to res\_dguest;  revoke select on dishes to res\_dguest;  revoke select, update on visitors to res\_dguest;  revoke select on orders to res\_chef;  revoke select on checkues to res\_chef;  revoke select, update on visitors to res\_dguest;  revoke select, insert, update, delete on PurchaseIngredient to res\_chef;  revoke select, insert, update, delete on Shipments to res\_chef;  revoke select, insert, update, delete on sostav to res\_chef;  revoke select, insert, update, delete on Ingredientes to res\_chef;  revoke select, insert, update, delete on Providers to res\_chef;  revoke select, insert, update, delete on Smeta to res\_chef;  revoke select, insert, update, delete on Dishes to res\_chef;  revoke select on IndividualOrders to res\_administrator;  revoke select on PurchaseIngredient to res\_administrator;  revoke select on Shipments to res\_administrator;  revoke select on orders to res\_administrator;  revoke select, insert, update, delete on Registered\_visitors to res\_administrator;  revoke select, insert, update, delete on visitors to res\_administrator;  revoke select, insert, update, delete on Employee to res\_administrator;  revoke select, insert, update, delete on Providers to res\_administrator;  revoke select, insert, update, delete on Smeta to res\_administrator;  revoke select, insert, update, delete on zones to res\_administrator;  drop index if  exists index\_kod\_sostav;  drop index if  exists index\_kod\_Shipments;  drop index if  exists index\_kod\_PurchaseIngredient;  drop index if  exists index\_kod\_IndividualOrders;  drop index if  exists index\_kod\_ingredient;  drop index if  exists index\_name\_ingredient;  drop index if  exists index\_kod\_zones;  drop index if  exists index\_name\_zones;  drop index if  exists index\_kod\_status;  drop index if  exists index\_name\_dish;  drop index if  exists index\_kod\_dish;  drop index if  exists index\_name\_table;  drop index if  exists index\_kod\_table;  drop index if  exists index\_number\_smeta;  drop index if  exists index\_kod\_smeta;  drop index if  exists index\_location\_provider;  drop index if  exists index\_name\_provider;  drop index if  exists index\_okpo\_provider;  drop index if  exists index\_kod\_provider;  drop index if  exists index\_number\_checkue;  drop index if  exists index\_date\_time\_checkue;  drop index if  exists index\_kod\_checkue;  drop index if  exists index\_name\_order;  drop index if  exists index\_date\_time\_order;  drop index if  exists index\_kod\_order;  drop index if  exists index\_fio\_visitor;  drop index if  exists index\_kod\_visitor;  drop index if  exists index\_fio\_rv;  drop index if  exists index\_login\_password\_rv;  drop index if  exists index\_pasportSN\_rv;  drop index if  exists index\_card\_rv;  drop index if  exists index\_kod\_rv;  drop index if  exists index\_fio\_employee;  drop index if  exists index\_login\_password\_employee;  drop index if  exists index\_value\_status;  drop table if  exists Employee;  drop table if  exists Registered\_Visitors;  drop table if  exists Visitors;  drop table if  exists Providers;  drop table if  exists Smeta;  drop table if  exists Dishes;  drop table if  exists Status;  drop table if  exists Zones;  drop table if  exists Ingredientes;  drop table if  exists Wood\_tables;  drop table if  exists Orders;  drop table if  exists Checkues;  drop table if  exists IndividualOrders;  drop table if  exists Shipments;  drop table if exists PurchaseIngredient;  drop table if exists Sostav;  Structure\_Create();      end;  $$; |
| --- |
|  |

1. Тестирование доступа к таблицам БД;

Таблица 7 – Доступ ролей к таблицам

| Роль | Локальная БД | | Удалённая БД | |
| --- | --- | --- | --- | --- |
| Есть доступ | Нет доступа | Есть доступ | Нет доступа |
| res\_waiter | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from zones; | select \* providers; | select \* fromorders; | select \* from providers; |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| res\_guest | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from orders; | select \* from ingredients; | select \* from orders; | select \* from providers; |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| res\_dguest | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from orders; | select \* from providers; | select \* from visitors; | select \* from providers; |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| res\_chef | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from smeta; | select \* from individualorders; | select \* from order; | select \* from zones; |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| res\_administrator | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from smeta; | select \*from dishes; | select \* from smeta; | select \*from dishes; |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |

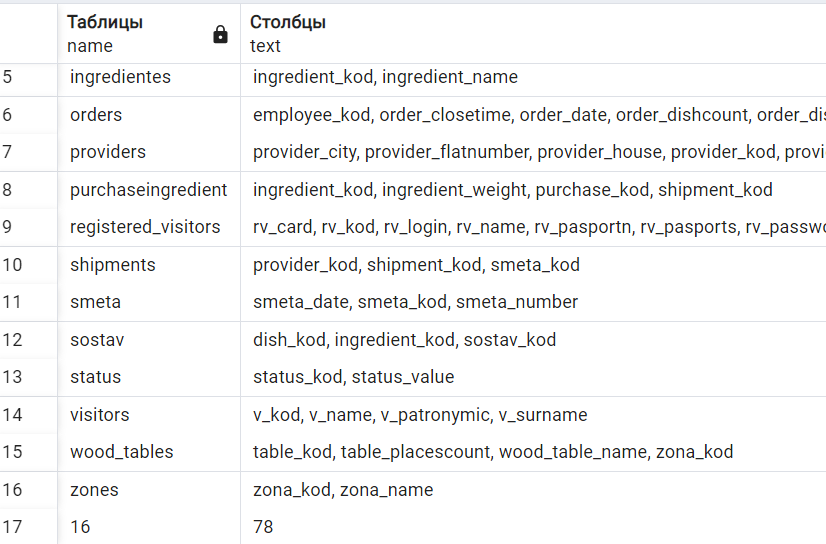
1. Создание резервной копии БД;



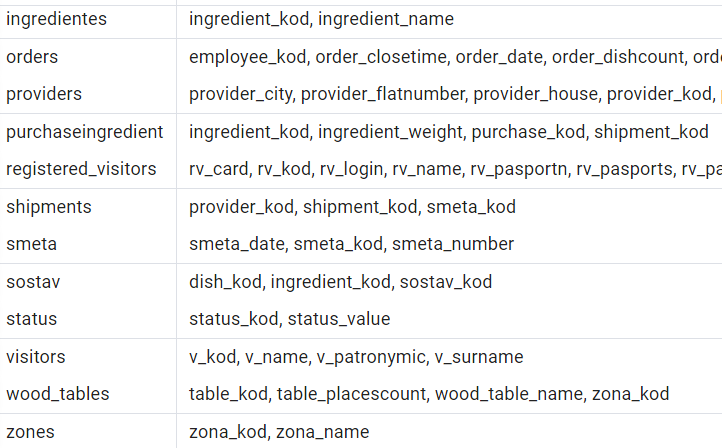
1. Версия базы данных

|  | Информация по объектам |
| --- | --- |
| Запрос | select  information\_schema.tables.table\_name as "Таблицы",  string\_agg(distinct information\_schema.columns.column\_name, ', ') as "Столбцы",  string\_agg(distinct pg\_indexes.indexname, ', ') as "Индексы"  from information\_schema.tables  inner join information\_schema.columns  on information\_schema.columns.table\_name = information\_schema.tables.table\_name  inner join pg\_indexes  on information\_schema.tables.table\_name = pg\_indexes.tablename  where  information\_schema.tables.table\_schema = 'public' and  indexname not like 'pk\_%'  group by  information\_schema.tables.table\_name  union all  select  (select  count(\*)::text  from information\_schema.tables  where  table\_schema = 'public'),  (select  count(information\_schema.columns.column\_name)::text  from information\_schema.tables  inner join information\_schema.columns  on information\_schema.columns.table\_name = information\_schema.tables.table\_name  where  information\_schema.tables.table\_schema = 'public'),  (select  count(pg\_indexes.indexname)::text  from information\_schema.tables  inner join pg\_indexes  on information\_schema.tables.table\_name = pg\_indexes.tablename  where  information\_schema.tables.table\_schema = 'public' and  indexname not like 'pk\_%'); |

Локальная бд



Удаленная бд



* 1. Версия БД.

Таблица 9 – Версия файла БД

|  |  |
| --- | --- |
| Параметры | PostgreSQL |
| Номер версии | 1.0.0.1 |
| Что сделано | * Созданы 16 таблиц; * Созданы 58 столбцов и межтабличные связи; * Созданы 35 индекса; * Произведено распределение доступа ролей к таблицам; * Созданы процедуры, для быстрого создания и перезаписи всей структуры таблиц, индексов и прав доступа; * Создан Backup файл. |