**Этапы выполнения работы:**

1. Создание файла скрипта;



1. Реализация функций;

Таблица 1 – Реализация функций по тестовым данным

| № Теста | Реализация | |
| --- | --- | --- |
|  | Сценарий тестирования | |
|  | |
| Скрипт реализации | |
| create or replace function get\_employee\_by\_login\_and\_password(p\_login varchar(50), p\_password varchar(50))  returns text  language plpgsql  as $$  declare final\_result text;  begin  if (select count(\*) from public.employee where employee.employee\_login =  p\_login and employee.employee\_password = p\_password) > 0 then  final\_result := employee\_surname||' '||employee\_name||' '||employee\_patronymic from employee where employee.employee\_login =  p\_login and employee.employee\_password = p\_password;    elsif (select count(\*) from public.registered\_visitors where  registered\_visitors.rv\_login = p\_login and registered\_visitors.rv\_password = p\_password) > 0 then  final\_result := concat(rv\_surname,' ',rv\_name,' ',rv\_patronymic) from registered\_visitors where registered\_visitors.employee\_login =  p\_login and registered\_visitors.employee\_password = p\_password;  else  final\_result := 'неверный логин или пароль';  end if;  return final\_result;  end;  $$; | |
| Результат древа объектов | |
|  | |
| Результат тестирования | |
| select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); |  |
| select get\_employee\_by\_login\_and\_password('of\_AndreevAA', 'Pa$$w0rd'); |  |
| select get\_employee\_by\_login\_and\_password('of\_DmitrievOI', 'Pa$$w0rd'); |  |
| select get\_employee\_by\_login\_and\_password('IvanovII', 'Pa$$w0rd'); |  |
| select get\_employee\_by\_login\_and\_password('PetrovAA', 'Pa$$w0rd'); |  |
| select get\_employee\_by\_login\_and\_password('PavlovEG', 'Pa$$w0rd'); |  |
| select get\_employee\_by\_login\_and\_password('User1', 'Password'); |  |
|  | Сценарий тестирования | |
|  | |
| Скрипт реализации | |
| create or replace function get\_dish\_discription(p\_dish\_name text)  returns text  language plpgsql  as $$  declare answer text;  begin  if (select count(\*) from public.dishes where dishes.dish\_name = p\_dish\_name) > 0 then  answer:= concat(dish\_weight, 'гр., цена: ',dish\_cost, 'р., состав: ', string\_agg(ingredient\_name,',')) from public.dishes  inner join public.sostav on  sostav.dish\_kod = dishes.dish\_kod  inner join public.ingredientes on  ingredientes.ingredient\_kod = sostav.ingredient\_kod  where dishes.dish\_name = p\_dish\_name  group by dish\_weight, dish\_cost;  else  answer := 'неверный логин или пароль';  end if;  return answer;  end;  $$; | |
| Результат древа объектов | |
|  | |
| Результат тестирования | |
| select get\_dish\_discription('Суп мечты'); |  |
| select get\_dish\_discription('Филе порося'); |  |
| select get\_dish\_discription('Гарнир офощной'); |  |
| select get\_dish\_discription('Мясная тарелка'); |  |
|  | select get\_dish\_discription('Картофель по своему'); |  |
|  | select get\_dish\_discription('Куриная нарезка'); |  |
|  | Сценарий тестирования | |
| 1. Количество формальных параметров: 1; | |
| Скрипт реализации | |
| create or replace function get\_booking\_discription(p\_booking\_number text)  returns table(client text,date\_and\_time text,  wod\_table\_name text, guests\_count text, dishes text)  language plpgsql  as $$  begin  if (select count(\*) from booking where book\_number =p\_booking\_number)> 0 and (select count(\*) from planning\_dishes where  book\_number = p\_booking\_number) > 0 then  return query  select  concat(rv\_surname,' ',rv\_name,' ',rv\_patronymic)::text,  concat(book\_plan\_date,book\_plan\_time)::text,  wood\_table\_name::text,  book\_guests\_count::text,  string\_agg(concat('"',dish\_name,'"',' X',dishes\_count,' - ',planning\_dishes\_time),', ')::text  from public.booking  inner join public.registered\_visitors  on registered\_visitors.rv\_kod = booking.v\_kod  inner join public.wood\_tables  on wood\_tables.table\_kod = booking.wood\_table\_kod  left join public.planning\_dishes  on planning\_dishes.book\_number = booking.book\_number  left join public.dishes  on dishes.dish\_kod = planning\_dishes.dish\_kod  where booking.book\_number =p\_booking\_number  group by concat(rv\_surname,' ',rv\_name,' ',rv\_patronymic),concat(book\_plan\_date,book\_plan\_time),wood\_table\_name,book\_guests\_count;  elsif (select count(\*) from booking where book\_number =p\_booking\_number)> 0 and (select count(\*) from planning\_dishes  where book\_number = p\_booking\_number) = 0 then  return query select  concat(rv\_surname,' ',rv\_name,' ',rv\_patronymic)::text,  concat(book\_plan\_date,book\_plan\_time)::text,  wood\_table\_name::text,  book\_guests\_count::text,  'нет информации о блюдах в предварительных заказах'  from public.booking  inner join public.registered\_visitors  on registered\_visitors.rv\_kod = booking.v\_kod  inner join public.wood\_tables  on wood\_tables.table\_kod = booking.wood\_table\_kod  where booking.book\_number =p\_booking\_number  group by concat(rv\_surname,' ',rv\_name,' ',rv\_patronymic),concat(book\_plan\_date,book\_plan\_time),wood\_table\_name,book\_guests\_count;  else  return query select  'нет информации',  'нет информации',  'нет информации',  'нет информации',  'нет информации';  end if;  end;  $$; | |
| Результат древа объектов | |
|  | |
| Результат тестирования | |
| select \* from public.get\_booking\_discription('БР/24/0000000001'); |  |
| select \* from public.get\_booking\_discription('БР/24/0000000002'); |  |
| select \* from public.get\_booking\_discription('БР/24/0000000003'); |  |
|  | select \* from public.get\_booking\_discription('БР/24/0000000009'); |  |

1. Распределение прав доступа к пользовательским функциям;

Таблица 2 – Права доступа к пользовательским функциям БД

| Роли | | официант | Зарег пользователь | пользоветлеь | шеф | админ |
| --- | --- | --- | --- | --- | --- | --- |
| Название объекта | Функции |
| get\_employee\_by\_login\_and\_password | Вызов |  |  |  | X | X |
| get\_dish\_discription | Вызов |  | X | X |  |  |
| get\_booking\_discription | Вызов | X |  |  |  |  |

1. Выдача прав доступа к пользовательским функциям;

Таблица 3 – Реализация разграничения прав доступа

| Название роли | Название объекта | Функция | Скрипт |
| --- | --- | --- | --- |
| Res\_waiter | get\_booking\_discription | Execute | grant execute on function get\_booking\_discription to res\_waiter; |
| Res\_chef | get\_employee\_by\_login\_and\_password | Execute | grant execute on function get\_employee\_by\_login\_and\_password to res\_chef; |
| Res\_dguest | get\_dish\_discription | Execute | grant execute on function get\_dish\_discription to res\_dguest; |
| Res\_guest | get\_dish\_discription | Execute | grant execute on function get\_dish\_discription to res\_guest; |
| res\_administrator | get\_employee\_by\_login\_and\_password | Execute | grant execute on function get\_employee\_by\_login\_and\_password to res\_administrator; |

1. Тестирование доступа к функциям;

Таблица 4 – Доступ ролей к функциям

| Роль | Локальная БД | | Удалённая БД | |
| --- | --- | --- | --- | --- |
| Есть доступ | Нет доступа | Есть доступ | Нет доступа |
| Res\_waiter | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from public.get\_booking\_discription('БР/24/0000000001'); | select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); | select \* from public.get\_booking\_discription('БР/24/0000000001'); | select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| Res\_chef | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from public.get\_booking\_discription('БР/24/0000000001'); | select \* from public.get\_booking\_discription('БР/24/0000000001'); | select \* from public.get\_booking\_discription('БР/24/0000000001'); | select \* from public.get\_booking\_discription('БР/24/0000000001'); |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| Res\_dguest | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from public.get\_booking\_discription('БР/24/0000000001'); | select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); | select \* from public.get\_booking\_discription('БР/24/0000000001'); | select Get\_Discip\_Group\_Count('3БК1.21'); |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| Res\_guest | Запрос: | Запрос: | Запрос: | Запрос: |
| select \* from public.get\_booking\_discription('БР/24/0000000001'); | select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); | select \* from public.get\_booking\_discription('БР/24/0000000001'); | select Get\_Discip\_Group\_Count('3БК1.21'); |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |
| res\_administrator | Запрос: | Запрос: | Запрос: | Запрос: |
| select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); | select get\_dish\_discription('Филе порося'); | select get\_employee\_by\_login\_and\_password('of\_SemenovKN', 'Pa$$w0rd'); | select get\_dish\_discription('Филе порося'); |
| Результат: | Результат: | Результат: | Результат: |
|  |  |  |  |

1. Создание резервной копии;



1. Версия базы данных:
   1. Отчёт о создании объектов;

Таблица 5 – Перечень созданных объектов

|  | Информация по объектам |
| --- | --- |
| Запрос | select  routine\_name as "Название функции"  from information\_schema.routines  where  routine\_type = 'FUNCTION' and routine\_schema = 'public'  union all  select  count(specific\_name)::text  from information\_schema.routines  where  routine\_type = 'FUNCTION' and routine\_schema = 'public'; |
| Результат локальной БД |  |
| Результат удалённой БД |  |

* 1. Версия БД.

Таблица 6 – Версия файла БД

|  |  |
| --- | --- |
| Параметры | PostgreSQL |
| Номер версии | 3.8.0.1 |
| Что сделано | * Созданы 3 пользовательские функции; * Произведено распределение доступа ролей к функциям; * Создан Backup файл. |