**Название КТ: КТ № 16. PostgreSQL. Реализация триггеров.**

**Этапы выполнения работы:**

1. Создание файла скрипта;



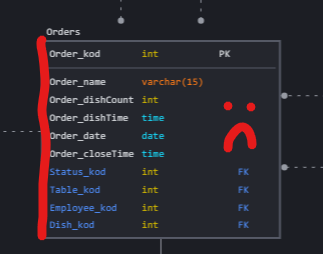
1. Описание предметной области;

Название предметной области: Колледж ITHub

Название предметной области: Ресторан

Заведение занимается предоставлением услуг питания и времяпровождения для клиентов. Ресторан имеет несколько зон: общая, детская и частная. В каждой зоне располагаются столы с местами, каждый стол имеет своё собственное обозначение, для участия в заказах клиентов. Официант прикрепляется за своим заказом, и контролирует процесс подачи блюд на стол. Каждый официант может просматривать свои активные заказы, каждый из посетителей может просматривать историю своих заказов.

1. Определение триггерной таблицы;



1. Определение таблиц и полей для заполнения журнальной таблицы;

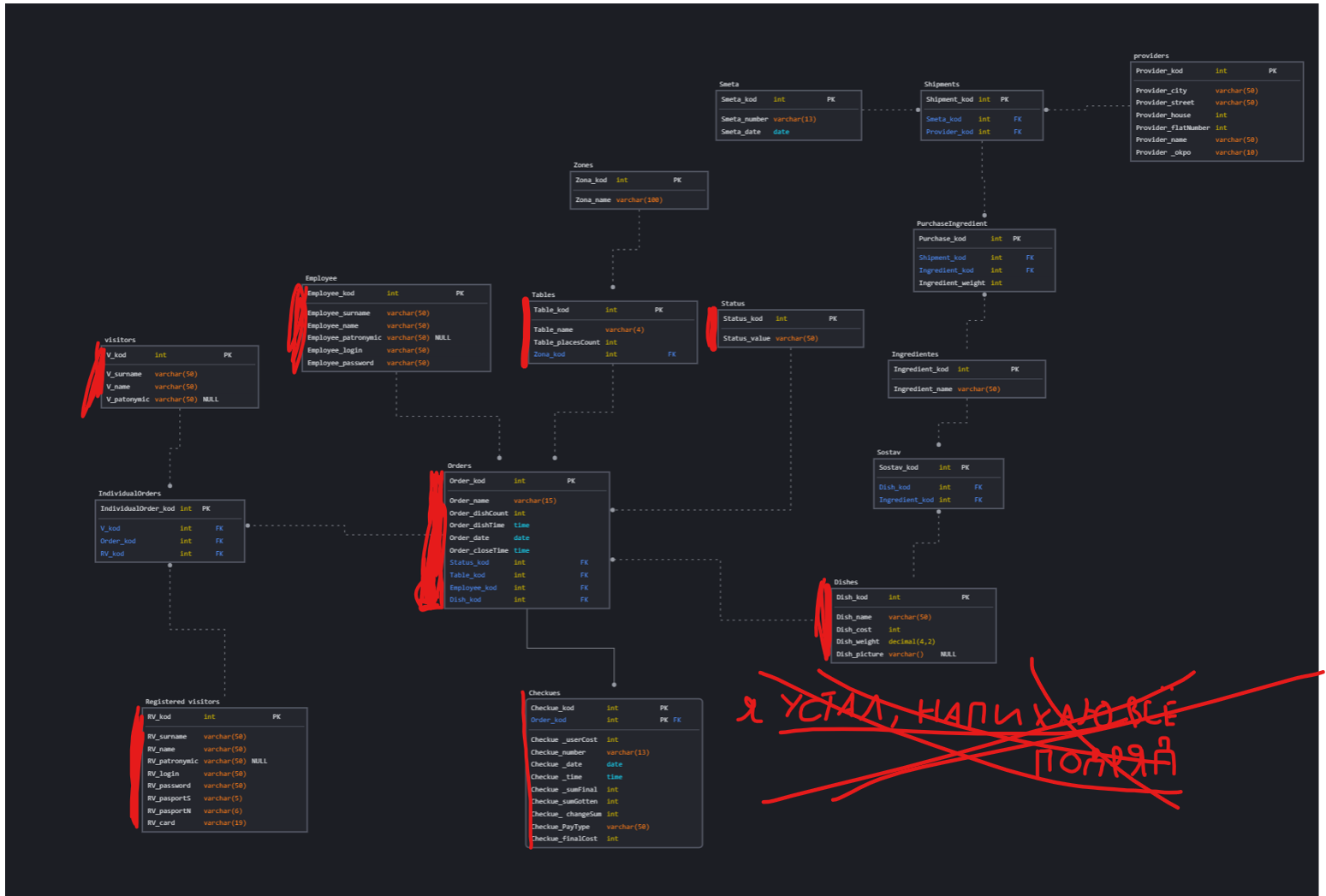


Таблица 1 – Проектирование структуры журнальной таблицы

| Название поля | Тип данных |
| --- | --- |
| Orders\_history | |
| ID\_ Orders\_history | UUID |
| Order\_name | Text |
| Wood\_table\_name | Text |
| Orders\_history\_status | Text |
| sostav | Text |
| Timestamp\_Create | Timestamp |

1. Реализация журнальной таблицы;

Таблица 2 – Создание журнальной таблицы

| Скрипт | Результат |
| --- | --- |
| CREATE TABLE IF NOT EXISTS Orders\_history  (  ID\_Orders\_history UUID DEFAULT uuid\_generate\_v4() CONSTRAINT PK\_Orders\_history PRIMARY KEY,  Order\_name TEXT NOT NULL,  Wood\_table\_name TEXT NOT NULL,  sostav text not null,  Orders\_history\_status TEXT NOT NULL CONSTRAINT CH\_history\_status CHECK (Orders\_history\_status IN ('Новая запись', 'Изменение', 'Удалённая')),  Timestamp\_Create TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP  ); |  |

1. Реализация триггеров;

Таблица 3 – Разработка триггеров

|  |  |  |
| --- | --- | --- |
| Тип триггера | Скрипт | Результат |
| Добавление | create or replace function fc\_orders\_history\_insert()  returns trigger  language plpgsql  as $$  begin  insert into orders\_history(order\_name,wood\_table\_name,sostav,orders\_history\_status)  values (new.order\_name,  (select wood\_table\_name  from public.wood\_tables where table\_kod = (select table\_kod from public.orders where orders.order\_name = new.order\_name)),  (select string\_agg(dish\_name, ', ') from public.orders  inner join public.dishes  on dishes.dish\_kod = orders.dish\_kod  where orders.order\_name = new.order\_name),  'Новая запись'  );  return new;  end;  $$;  create trigger tg\_orders\_history\_insert  after insert on orders  for each row  execute procedure fc\_orders\_history\_insert(); |  |
| Изменение | create or replace function fc\_orders\_history\_update()  returns trigger  language plpgsql  as $$  begin  insert into orders\_history(order\_name,wood\_table\_name,sostav,orders\_history\_status)  values (new.order\_name,  (select wood\_table\_name  from public.wood\_tables where table\_kod = (select table\_kod from public.orders where orders.order\_name = new.order\_name)),  (select string\_agg(dish\_name, ', ') from public.orders  inner join public.dishes  on dishes.dish\_kod = orders.dish\_kod  where orders.order\_name = new.order\_name),  'Изменение'  );  return new;  end;  $$;  create trigger tg\_orders\_history\_update  after update on orders  for each row  execute procedure fc\_orders\_history\_update(); |  |
| Удаление | create or replace function fc\_orders\_history\_delete()  returns trigger  language plpgsql  as $$  begin  insert into orders\_history(order\_name,wood\_table\_name,sostav,orders\_history\_status)  values (old.order\_name,  (select wood\_table\_name  from public.wood\_tables where table\_kod = (select table\_kod from public.orders where orders.order\_name = old.order\_name)),  (select string\_agg(dish\_name, ', ') from public.orders  inner join public.dishes  on dishes.dish\_kod = orders.dish\_kod  where orders.order\_name = old.order\_name),  'Удалённая'  );  return old;  end;  $$;  create trigger tg\_orders\_history\_delete  before delete on orders  for each row  execute procedure fc\_orders\_history\_delete(); |  |

1. Распределение прав доступа к журнальной таблице;

Таблица 4 – Права доступа к журнальной таблице БД

| Роли | | Админ | Шеф | Официант | Зарег юзер | юзер |
| --- | --- | --- | --- | --- | --- | --- |
| Название объекта | Функции |
| orders\_history | Выборка |  |  | Х | Х |  |

1. Выдача прав доступа к журнальной таблице;

Таблица 5 – Реализация разграничения прав доступа

| Название роли | Название объекта | Функция | Скрипт |
| --- | --- | --- | --- |
| Res\_waiter | orders\_history | Select | grant select on public.orders\_history to res\_waiter; |
| Res\_guest | orders\_history | Select | grant select on public.orders\_history to res\_guest; |

1. Тестирование заполнения журнальной таблицы;

Таблица 5 – Тестовые данные для заполнения триггерной таблицы

| Событие | Входные данные |
| --- | --- |
| Добавить | ЗКЗ-000000001-24, Иванов Иван Иванович, Семёнов Кирилл Николаевич, «Суп мечты», x3, 2250 р |
| Добавить | ЗКЗ-000000002-24, Петров Алексей Алексеевич, Андреев Андрей Андреевич, «Гарнир овощной», x1, 650 р |
| Изменить | ЗКЗ-000000001-24, Иванов Иван Иванович, Семёнов Кирилл Николаевич, «Суп мечты», x3, 2250 р → ЗКЗ-000000001-24, Иванов Иван Иванович, Семёнов Кирилл Николаевич, Мясная тарелка», x1, 1670 р |
| Удалить | ЗКЗ-000000002-24, Петров Алексей Алексеевич, Андреев Андрей Андреевич, «Гарнир овощной», x1, 650 р |

Таблица 6 – Манипулирование данными

|  |  |
| --- | --- |
| Запрос | Выборка |
| insert into public.individualorders(order\_kod,v\_kod,person\_cost)  values(18, 'зп3', 2250);  call orders\_insert('ЗКЗ-000000001-24',3,'12:20:35', '2023.09.03', '12:10:41',8,1,2,2); | Результат триггерной таблицы |
|  |
| Результат журнальной таблицы |
|  |
| call orders\_insert('ЗКЗ-000000001-24',3,'12:20:35', '2023.09.03', '12:10:41',8,1,2,2); | Результат триггерной таблицы |
|  |
| Результат журнальной таблицы |
|  |
| call orders\_update('ЗКЗ-000000002-24',1,'12:20:35', '2023.09.03', '12:10:41',8,3,2,4,20); | Результат триггерной таблицы |
|  |
| Результат журнальной таблицы |
|  |
| call orders\_delete(20); |  |

Таблица 7 – Создание функций для вывода данных их журнальной таблицы пользователями

| Скрипт | Результат |
| --- | --- |
| create or replace function get\_orders\_list\_for\_user(p\_user\_login text)  returns table(order\_name text,wood\_table\_name text,sostav text,orders\_history\_status text,timestamp\_create timestamp)  language plpgsql  as $$  begin  return query  select orders\_history.order\_name,orders\_history.wood\_table\_name,orders\_history.sostav,  orders\_history.orders\_history\_status,orders\_history.timestamp\_create  from public.orders\_history  inner join orders  on orders.order\_name = orders\_history.order\_name  inner join public.individualorders  on individualorders.order\_kod = orders.order\_kod  inner join public.registered\_visitors  on registered\_visitors.rv\_login = p\_user\_login;  end;  $$; | Обозреватель объектов |
|  |
| Выдача прав | grant execute on function get\_orders\_list\_for\_user res\_guest; |
| Тестовый запрос | select \* from get\_orders\_list\_for\_user('IvanovII'); |
| Вывод значений |  |
| create or replace function get\_orders\_list\_for\_employee(p\_employee\_login text)  returns table(order\_name text,wood\_table\_name text,sostav text,orders\_history\_status text,timestamp\_create timestamp)  language plpgsql  as $$  begin  return query  select orders\_history.order\_name,orders\_history.wood\_table\_name,orders\_history.sostav,  orders\_history.orders\_history\_status,orders\_history.timestamp\_create  from public.orders\_history  inner join orders  on orders.order\_name = orders\_history.order\_name  inner join public.employee  on employee.employee\_login = p\_employee\_login;  end;  $$; | Обозреватель объектов |
|  |
| Выдача прав | grant execute on function Function\_ get\_orders\_list\_for\_employee res\_waiter; |
| Тестовый запрос | select \* from get\_orders\_list\_for\_employee('of\_SemenovKN'); |
| Вывод значений |  |

1. Создание резервной копии;



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

Таблица 8 – Перечень созданных объектов

|  | Информация по объектам |
| --- | --- |
| Запрос | select  table\_name as "Таблица",  string\_agg(trigger\_name||' '||event\_manipulation, ', ') as "Триггеры"  from information\_schema.triggers  inner join information\_schema.tables on  table\_name = event\_object\_table  group by  table\_name  union all  select  'Количество'::name,  count(trigger\_name)::text  from information\_schema.triggers  inner join information\_schema.tables on  table\_name = event\_object\_table  group by  table\_name; |
| Результат локальной БД |  |
| Результат удалённой БД |  |

* 1. Версия БД.

Таблица 9 – Версия файла БД

|  |  |
| --- | --- |
| Параметры | PostgreSQL |
| Номер версии | 3.10.0.1 |
| Что сделано | * Создана 1 таблица; * Созданы 3 триггера; * Создана 1 функция; * Произведено распределение доступа таблице; * Создан Backup файл. |