Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет им. Г.В. Плеханова»

Московский приборостроительный техникум

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Общепрофессиональная дисциплина: ОП 08 Основы проектирования баз данных

МПТ.09.02.07-П.08.ОПБД.П50-5-20.01.22

Отчёт к практической работе № 11

«Разработка триггеров»

Тема: Разработка базы данных на примере предметной области «Организация наземных

грузоперевозок». Реализация подсистем: учёт автомобильного парка, организация доставки грузов.

|  |  |
| --- | --- |
| Проверил: | Выполнил: |
| Щаников И.М.\_\_\_\_\_\_\_\_\_\_ | Студент группы |
| (\_\_\_\_\_\_\_\_)\_\_\_\_\_\_\_\_\_\_\_\_\_ | П50-5-20 |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. | \_\_\_\_\_\_\_\_\_\_\_\_Атаниязов А.А. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. |

2022

СОДЕРЖАНИЕ

[1.](#_30j0zll) ЦЕЛИ ПРАКТИЧЕСКОЙ РАБОТЫ 3

[2.](#_1fob9te) ЭТАПЫ ВЫПОЛНЕНИЯ 4

[3.](#_3znysh7) ВЫВОД 5

[4.](#_2et92p0) СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 6

1. ЦЕЛИ ПРАКТИЧЕСКОЙ РАБОТЫ

На основании разработанной структуры базы данных, реализовать статистические функции

ЭТАПЫ ВЫПОЛНЕНИЯ

1. Из описания предметной области, определить фрагмент(ы) описания вывода истории для того или иного пользователя;

**Предметная область:** Организация осуществляет грузовые наземные перевозки, в соответчики с маршрутным листом. Организация заказчик оставляет заявку, организации перевозчику, которую обрабатывают и формируют маршрутный лист, в котором указаны все точки доставки, для

водителя. При этом после каждой доставки в конкретную точку, делается пометка о завершении пункта маршрутного листа. Водители могут просматривать историю своих рейсов. Заказчики могут видеть историю своих

заявок.

1. Реализовать, не связанную с общей структурой, таблицу, для хранения истории. Таблица может быть выполнена не по правилам нормализации;

|  |  |
| --- | --- |
| Скрипт DDL | Результат |
| create table [dbo].[Employee\_Application\_History]  (  [ID\_Employee\_Application] [uniqueidentifier] not null,  [Employee\_Info] [varchar] (max) not null,  [Application\_Info] [varchar] (max) not null,  constraint [PK\_Agent\_Customer] primary key clustered  ([ID\_Employee\_Application] ASC) on [PRIMARY]  )  go |  |
| Скрипт DDL | Результат |
| create table [dbo].[Customer\_Application\_History]  (  [ID\_Customer\_Application] [uniqueidentifier] not null,  [Customer\_Info] [varchar] (max) not null,  [Application\_Info] [varchar] (max) not null,  constraint [PK\_Customer\_Application] primary key clustered  ([ID\_Customer\_Application] ASC) on [PRIMARY]  )  go |  |

1. Реализовать триггеры для 3-ёх ситуаций: после добавления, изменения и удаления данных;

|  |  |
| --- | --- |
| Скрипт DDL | Результат |
| Триггер на добавление данных |  |
| create or alter trigger dbo.Employee\_Application\_Insert  on dbo.EmployeeCust after insert  as  insert into dbo.Employee\_Application\_History ([ID\_Employee\_Application], [Employee\_Info], [Application\_Info])  values (NewID(),  (select CONCAT('ФИО: ',Lastname\_Employee,' ',Name\_Employee,' ',Surname\_Employee) from dbo.Employee  where ID\_Employee = (select Employee\_ID from inserted)),  (select CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) from dbo.Application  where ID\_Application = (select Application\_ID from inserted)))  print('Insert Record Complete!')  go |
| Триггер на изменение данных |
| create or alter trigger dbo.Employee\_Application\_Update  on dbo.[EmployeeCust] after update  as  insert into dbo.Employee\_Application\_History ([ID\_Employee\_Application], [Employee\_Info], [Application\_Info])  values (NewID(),  (select CONCAT('ФИО: ',Lastname\_Employee,' ',Name\_Employee,' ',Surname\_Employee) from dbo.Employee  where ID\_Employee = (select Employee\_ID from inserted)),  (select CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) from dbo.Application  where ID\_Application = (select Application\_ID from inserted)))  print('Update Record Complete!')  go |
| Триггер на удаление данных |
| create or alter trigger dbo.Employee\_Application\_Delete  on dbo.[EmployeeCust] after delete  as  insert into dbo.Employee\_Application\_History ([ID\_Employee\_Application], [Employee\_Info], [Application\_Info])  values (NewID(),  (select CONCAT('ФИО: ',Lastname\_Employee,' ',Name\_Employee,' ',Surname\_Employee) from dbo.Employee  where ID\_Employee = (select Employee\_ID from deleted)),  (select CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) from dbo.Application  where ID\_Application = (select Application\_ID from inserted)))  print('Delete Record Complete!')  go |

|  |  |
| --- | --- |
| Скрипт DDL | Результат |
| Триггер на добавление данных |  |
| create or alter trigger dbo.Customer\_Application\_Insert  on dbo.CustomerCust after insert  as  insert into dbo.Customer\_Application\_History ([ID\_Customer\_Application], [Customer\_Info], [Application\_Info])  values (NewID(),  (select CONCAT('ФИО: ',Lastname\_Customer,' ',Name\_Customer,' ',Surname\_Customer) from dbo.Customer  where ID\_Customer = (select Customer\_ID from inserted)),  (select CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) from dbo.Application  where ID\_Application = (select Application\_ID from inserted)))  print('Insert Record Complete!')  go |
| Триггер на изменение данных |
| create or alter trigger dbo.Customer\_Application\_Update  on dbo.[CustomerCust] after update  as  insert into dbo.Customer\_Application\_History ([ID\_Customer\_Application], [Customer\_Info], [Application\_Info])  values (NewID(),  (select CONCAT('ФИО: ',Lastname\_Customer,' ',Name\_Customer,' ',Surname\_Customer) from dbo.Customer  where ID\_Customer = (select Customer\_ID from inserted)),  (select CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) from dbo.Application  where ID\_Application = (select Application\_ID from inserted)))  print('Update Record Complete!')  go |
| Триггер на удаление данных |
| create or alter trigger dbo.Customer\_Application\_Delete  on dbo.[CustomerCust] after delete  as  insert into dbo.Customer\_Application\_History ([ID\_Customer\_Application], [Customer\_Info], [Application\_Info])  values (NewID(),  (select CONCAT('ФИО: ',Lastname\_Customer,' ',Name\_Customer,' ',Surname\_Customer) from dbo.Customer  where ID\_Customer = (select Customer\_ID from inserted)),  (select CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) from dbo.Application  where ID\_Application = (select Application\_ID from inserted)))  print('Delete Record Complete!')  go |

1. Заполнить триггерную таблицу (при необходимости, ряд дочерних таблиц), тестовыми данными не менее 5 записей и вывести результат;

|  |
| --- |
| Скрипт DML |
| insert into dbo.EmployeeCust ([Employee\_ID], [Application\_ID])  values (1,1)  go  insert into dbo.EmployeeCust ([Employee\_ID], [Application\_ID])  values (2,2)  go  insert into dbo.EmployeeCust ([Employee\_ID], [Application\_ID])  values (3,3)  go  insert into dbo.EmployeeCust ([Employee\_ID], [Application\_ID])  values (4,4)  go  insert into dbo.EmployeeCust ([Employee\_ID], [Application\_ID])  values (5,5)  go |
| Результат |
|  |

|  |
| --- |
| Скрипт DML |
| insert into dbo.CustomerCust ([Customer\_ID], [Application\_ID])  values (1,1)  go  insert into dbo.CustomerCust ([Customer\_ID], [Application\_ID])  values (2,2)  go  insert into dbo.CustomerCust ([Customer\_ID], [Application\_ID])  values (3,3)  go  insert into dbo.CustomerCust ([Customer\_ID], [Application\_ID])  values (4,4)  go  insert into dbo.CustomerCust ([Customer\_ID], [Application\_ID])  values (5,5)  go |
| Результат |
|  |

1. Вывести результат из таблицы с историей;

|  |
| --- |
| Скрипт DML |
| select \* from [dbo].[Employee\_Application\_History] |
|  |

|  |
| --- |
| Скрипт DML |
| select \* from [dbo].[Customer\_Application\_History] |
|  |

1. Реализовать любым из методов: запросом, представлением или функцией вывод данных для каждого из пользователя.

|  |  |
| --- | --- |
| Роль | Создание объекта и вызов |
| Сотрудник | create or alter function [dbo].Employee\_App\_History (@Employee\_Info [varchar] (100))  returns table  as  return (select CONCAT('ФИО: '+Lastname\_Employee,' ',Name\_Employee,' ',Surname\_Employee) as "ФИО сотрудника",  CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) as "Номер заявки"  from dbo.[Application]  inner join dbo.Route\_Sheet on ID\_Application = Route\_Sheet\_ID  inner join dbo.Carrier on Carrier\_ID = ID\_Carrier  inner join dbo.Employee on Carrier.ID\_Carrier = ID\_Employee  group by CONCAT('ФИО: '+Lastname\_Employee,' ',Name\_Employee,' ',Surname\_Employee),  CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]))  Go  select \* from Employee\_App\_History ('2') |
| Результат |
|  |

|  |  |
| --- | --- |
| Роль | Создание объекта и вызов |
| Заказчик | create or alter function [dbo].Customer\_App\_History (@Customer\_Info [varchar] (100))  returns table  as  return (select CONCAT('ФИО: '+Lastname\_Customer,' ',Name\_Customer,' ',Surname\_Customer) as "ФИО сотрудника",  CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]) as "Номер заявки"  from dbo.[Customer]  inner join dbo.Application on Application.Customer\_ID = ID\_Customer  group by CONCAT('ФИО: '+Lastname\_Customer,' ',Name\_Customer,' ',Surname\_Customer),  CONCAT('Номер заявки: ', Number\_Application, ' Статус: ', [Status\_Application], ' Дата: ', [Date\_Application], ' Время: ', [Time\_Application]))  Go |
| Результат |
|  |