Федеральное государственное образовательное учреждение

высшего образования

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

Московский приборостроительный техникум

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Профессиональный модуль: ПМ 11 Разработка, администрирование и защита баз данных

Междисциплинарный курс: МДК 11.01 Технология разработки и защиты баз данных

МПТ.09.02.07-П.П50-1-18.09.20

Отчёт

Тема: Разработка базы данных на примере предметной области: «ЦМИТ», реализация подсистем: «Учет выполненных работ», «Учет складских запасов»

|  |  |  |
| --- | --- | --- |
| Проверил: | Выполнил: | Выполнил: |
| Щаников И.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент группы | Студент(ка) группы |
| (\_\_\_\_\_\_\_\_\_\_)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | П50-1-18 | П50-1,2,3,4,5,6,7,11-17,18 |
| «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Дрюпин А.А. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Иванов И.И. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

2020

СОДЕРЖАНИЕ

[1. АНАЛИЗ ПРОБЛЕМНОЙ ЗОНЫ 3](#_Toc52187135)

[1.1. Описание предметной области 3](#_Toc52187136)

[1.2. Описание подсистем 3](#_Toc52187137)

[1.3. Описание сущностей 3](#_Toc52187138)

[2. ПРОЕКТИРОВАНИЕ 4](#_Toc52187139)

[2.1. Модель данных 4](#_Toc52187140)

[2.2. Словарь данных 4](#_Toc52187141)

[3. РЕАЛИЗАЦИЯ 5](#_Toc52187142)

[3.1. Описание объектов 5](#_Toc52187143)

[3.2. Скрипт базы данных 5](#_Toc52187144)

[4. ВЫВОД 7](#_Toc52187145)

[5. СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 8](#_Toc52187146)

1. АНАЛИЗ ПРОБЛЕМНОЙ ЗОНЫ
   1. Описание предметной области

ЦМИТ (Центр Молодежного Инновационного Творчества) - предоставляет доступ к оборудованию жителям города Москвы и организациям, осуществляющим научно-техническую и инновационную деятельность, для выполнение опытно-конструкторских и иных работ. Основной деятельностью лаборатории является разработка, производство и продажа корпусов для СВЧ-электроники.

* 1. Описание подсистем

Учёт выполненных работ – хранит информацию о выполненных работах, затраченных на нее материалах, доходности и временных затратах.

Учёт складских запасов – предназначена для учета запаса расходников, инструмента, оборудования и модулей, предназначенных для работы. Влияет на доходность и количество выполняемых работ и их качество.

* 1. Описание сущностей

Заявка на работы – номер заявки, ФИО исполнителя, номер чека, тип работ, техническое задание, код инвентаря.

Сотрудник – Номер, ФИО.

Статус выполнения – Номер, название статуса.

Тип работы – Номер, название.

Тип хранения – Номер, тип.

Чек – Номер, дата и время, стоимость.

Инвентарь – Номер, наименование позиции, количество, стоимость, тип хранения, номер закупки.

Материалы – Номер, чек, инвентарь, заявка на работы.

1. ПРОЕКТИРОВАНИЕ
   1. Модель данных
      1. Инфологическая модель данных

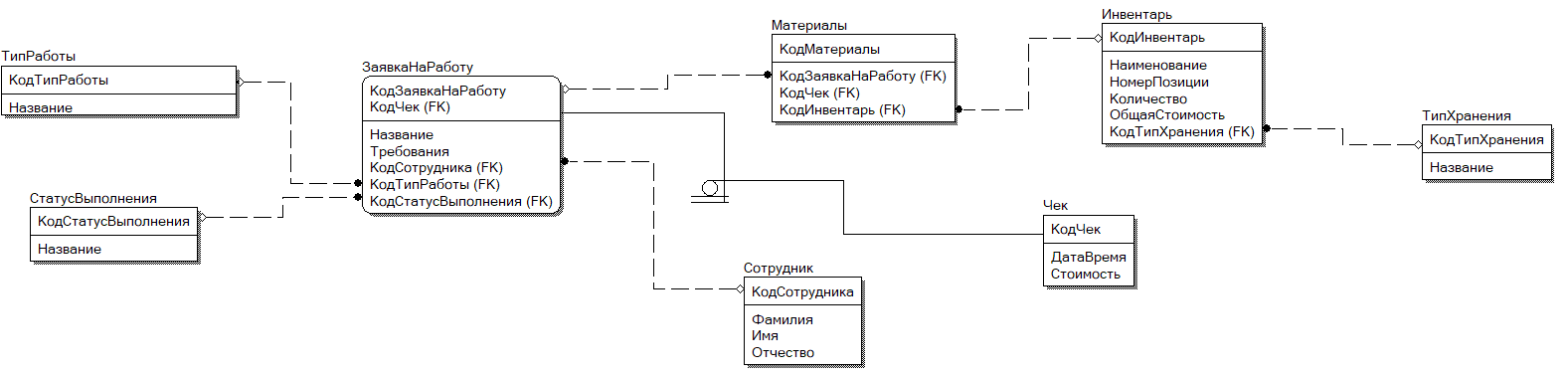


Рисунок - Инфологическая модель данных

* + 1. Даталогическая модель данных

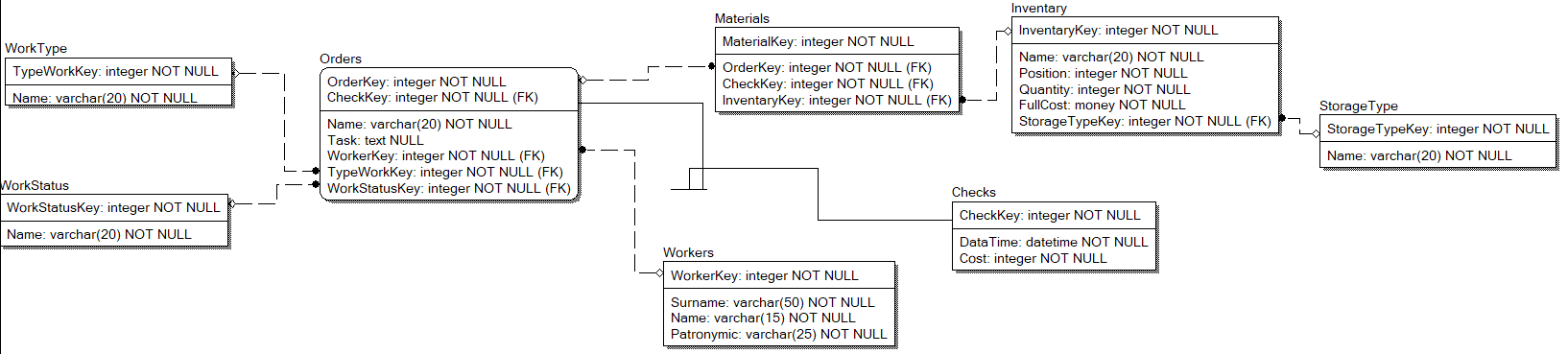


Рисунок – Даталогическая модель данных

* 1. Словарь данных
     1. Входные данные

Таблица 2.1 – Входные данные

| Логическое название | Физическое название | Тип данных | | Ограничения |
| --- | --- | --- | --- | --- |
| ТипРаботы | | | | |
| КодТипРаботы | WorkTypeKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| Название | Name | Varchar(20) | | Not null, unique |
| СтатусВыполнения | | | | |
| КодСтатус выполнения | WorkStatusKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| Название | Name | Varchar(20) | | Not null, unique |
| ЗаявкаНаРаботу | | | | |
| КодЗаявкаНаРаботу | OrderKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| КодЧек(FK) | CheckKey | int | | Not null, foreign key |
| Название | Name | Varchar(20) | | Not null |
| Требования | Task | text | | null |
| КодСотрудника(FK) | WorkerKey | int | | Not null, foreign key |
| КодТипРаботы(FK) | TypeWorkKey | int | | Not null, foreign key |
| КодСтатусВыполнения(FK) | WorkStatusKey | int | | Not null, foreign key |
| Сотрудник | | | | |
| КодСотрудника | WorkerKey | int | Not null, PRIMARY KEY IDENTITY(1,1) | |
| Фамилия | Surname | Varchar(50) | | Not null |
| Имя | Name | Varchar(15) | | Not null |
| Отчество | Patronymic | Varchar(25) | | Null, default(‘-’) |
| Материалы | | | | |
| КодМатериалы | MaterialKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| КодЗаявкаНаРаботу(FK) | OrderKey | int | | Not null, foreign key |
| КодЧек(FK) | CheckKey | int | | Not null, foreign key |
| КодИнвентарь(FK) | InventaryKey | int | | Not null, foreign key |
| Чек | | | | |
| КодЧек | CheckKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| ДатаВремя | DataTime | datetime | | DEFAULT CURRENT\_TIMESTAMP |
| Стоимость | Cost | int | | Not null, check(Cost>0) |
| Инвентарь | | | | |
| КодИнвентарь | InventaryKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| Наименование | Name | Varchar(50) | | Not null, unique |
| НомерПозиции | Position | int | | Not null, check(Position>0) |
| Количество | Quantity | int | | Not null, check(Quantity>0) |
| ОбщаяСтоимость | FullCost | int | | Not null, check(FullCost>0) |
| КодТипХранения(FK) | StorageTypeKey | int | | Not null, foreign key |
| ТипХранения | | | | |
| КодТипХранения | StorageTypeKey | int | | Not null, PRIMARY KEY IDENTITY(1,1) |
| Название | Name | Varchar(20) | | Not null, unique |

* + 1. Выходные данные

Таблица 2.2 – Выходные данные

| Название документа | Поля | Исходные таблицы |
| --- | --- | --- |
| Заявка на работу | Наименование, статус выполнения | «Статус выполнения», «Сотрудник», «Чек», «ЗаявкаНаРаботу» |
| Чек |
| ФИО сотрудника |
| Материалы | Код материалы | «Заявка на работу», «Инвентарь», «Материалы» |
| Наименование |
| Код Инвентарь |

1. РЕАЛИЗАЦИЯ
   1. Описание объектов

Таблица 3.1 – Описание объектов базы данных

|  |  |
| --- | --- |
| Название объекта | Описание объекта |
| [dbo].[Checks] | Таблица с чеками |
| [dbo].[Inventary] | Таблица с инвентарем |
| [dbo].[Materials] | Таблица с материалами |
| [dbo].[Orders] | Таблица с заказами |
| [dbo].[StorageType] | Таблица с типами хранения |
| [dbo].[Workers] | Таблица с сотрудниками |
| [dbo].[WorkStatus] | Таблица со статусом выполнения |
| [dbo].[WorkType] | Таблица с типом работы |
| [dbo].[Checks\_Insert] | Процедура добавления записи в таблицу с чеками |
| [dbo].[Inventary\_Insert] | Процедура добавления записи в таблицу с инвентарем |
| [dbo].[Materials\_Insert] | Процедура добавления записи в таблицу с материалами |
| [dbo].[Orders\_Insert] | Процедура добавления записи в таблицу с заказами |
| [dbo].[StorageType\_Insert] | Процедура добавления записи в таблицу с типом хранения |
| [dbo].[Workers\_Insert] | Процедура добавления записи в таблицу с сотрудниками |
| [dbo].[WorkStatus\_Insert] | Процедура добавления записи в таблицу с статусом выполнения |
| [dbo].[WorkType\_Insert] | Процедура добавления записи в таблицу с типом работы |
| [dbo].[Checks\_Delete] | Процедура удаления записи из таблицы с чеками |
| [dbo].[Inventary\_Delete] | Процедура удаления записи из таблицы с инвентарем |
| [dbo].[Materials\_Delete] | Процедура удаления записи из таблицы с материалами |
| [dbo].[Orders\_Delete] | Процедура удаления записи из таблицы с заказами |
| [dbo].[StorageType\_Delete] | Процедура удаления записи из таблицы с типом хранения |
| [dbo].[Workers\_Delete] | Процедура удаления записи из таблицы с сотрудниками |
| [dbo].[WorkStatus\_Delete] | Процедура удаления записи из таблицы с статусом выполнения |
| [dbo].[WorkType\_Delete] | Процедура удаления записи из таблицы с типом работы |
| [dbo].[Inventary\_Update] | Процедура обновления записи в таблице с инвентарем |
| [dbo].[Materials\_Update] | Процедура обновления записи в таблице с материалами |
| [dbo].[Orders\_Update] | Процедура обновления записи в таблице с заказами |
| [dbo].[StorageType\_Update] | Процедура обновления записи в таблице с типом хранения |
| [dbo].[Workers\_Update] | Процедура обновления записи в таблице с сотрудниками |
| [dbo].[WorkStatus\_Update] | Процедура обновления записи в таблице с статусом выполнения |
| [dbo].[WorkType\_Update] | Процедура обновления записи в таблице с типом работы |
| [dbo].[Orders\_View] | Представление «заявка на работу» |
| [dbo].[Materials\_View] | Представление «Материалы» |
| [dbo].[Description\_Func] | Функция получения описания заказа |
| [dbo].[ CheckDeletedTrigger] | Триггер удаления заказа по чеку |

* 1. Скрипт базы данных
     1. Таблицы

Таблица «Чеки»

CREATE TABLE [dbo].[Checks]

(

[CheckKey] [int] not null IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[DataTime] [datetime] NOT NULL DEFAULT CURRENT\_TIMESTAMP,

--Временная переменная, обязательная для заполнения

[Cost] [int] NOT NULL,

--Целочисленная переменная, обязательная для заполнения с проверкой больше 0

constraint [PK\_Checks] primary key clustered ([CheckKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [CH\_Cost] check ([Cost] >= 0)

--Проверка на положительный знак числа

)

go

Таблица «Инвентарь»

CREATE TABLE [dbo].[Inventary]

(

[InventaryKey] int IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[Name] [char] (50) NOT NULL ,

--Cтроковая переменная с размерностью в 50 символов, обязательная для заполнения

[Position] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[Quantity] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[FullCost] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[StorageTypeKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

constraint [PK\_Inventary] primary key clustered ([InventaryKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [CH\_Position] check ([Position] >= 0),

--Проверка на положительный знак числа

constraint [CH\_Quantity] check ([Quantity] >= 0),

--Проверка на положительный знак числа

constraint [CH\_FullCost] check ([FullCost] >= 0),

--Проверка на положительный знак числа

constraint [UQ\_InventaryName] unique ([Name]),

--Проверка на уникальность

constraint [FK\_Storage\_Type\_Key] foreign key ([StorageTypeKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[StorageType]([StorageTypeKey]),

--Ссылка на значение первичного ключа родительской таблицы

)

go

Таблица «Материалы»

CREATE TABLE [dbo].[Materials]

(

[MaterialKey] [int] IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[OrderKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[CheckKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[InventaryKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

constraint [PK\_Inventary] primary key clustered ([InventaryKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [FK\_Inventary\_Key] foreign key ([InventaryKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[Inventary]([InventaryKey]),

--Ссылка на значение первичного ключа родительской таблицы

constraint [FK\_Order\_Key] foreign key ([OrderKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[Orders]([OrderKey]),

--Ссылка на значение первичного ключа родительской таблицы

constraint [FK\_Check\_Key] foreign key ([CheckKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[Checks]([CheckKey]),

--Ссылка на значение первичного ключа родительской таблицы

)

go

Таблица «Заказы»

CREATE TABLE [dbo].[Orders]

(

[OrderKey] [int] IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[Name] [varchar] (20) NOT NULL ,

--Cтроковая переменная с размерностью в 50 символов, обязательная для заполнения

[Task] [text] NULL ,

--Текстовая переменная

[WorkerKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[TypeWorkKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[WorkStatusKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

[CheckKey] [int] NOT NULL ,

--Целочисленное обязательное для заполнения поле

constraint [PK\_Orders] primary key clustered ([OrderKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [FK\_CheckKey] foreign key ([CheckKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[Checks]([CheckKey]),

--Ссылка на значение первичного ключа родительской таблицы

constraint [FK\_WorkerKey] foreign key ([WorkerKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[Workers]([WorkerKey]),

--Ссылка на значение первичного ключа родительской таблицы

constraint [FK\_TypeWorkKey] foreign key ([TypeWorkKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[WorkType]([TypeWorkKey]),

--Ссылка на значение первичного ключа родительской таблицы

constraint [FK\_WorkStatusKey] foreign key ([WorkStatusKey])

--Ограничение на поле с типом "Внешний ключ" с указанием какое поле в таблице является внешним ключом

references [dbo].[WorkStatus]([WorkStatusKey])

--Ссылка на значение первичного ключа родительской таблицы

)

go

Таблица «Типы хранения»

CREATE TABLE [dbo].[StorageType]

(

[StorageTypeKey] [int] IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[Name] [varchar] (20) NOT NULL,

--Cтроковая переменная с размерностью в 20 символов, обязательная для заполнения

constraint [PK\_StorageType] primary key clustered ([StorageTypeKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [UQ\_StorageType] unique ([Name]),

--Проверка на уникальность

)

go

Таблица «Сотрудники»

CREATE TABLE [dbo].[Workers]

(

[WorkerKey] [int] IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[Surname] [varchar] (50) NOT NULL ,

--Cтроковая переменная с размерностью в 50 символов, обязательная для заполнения

[Name] [varchar] (50) NOT NULL ,

--Cтроковая переменная с размерностью в 50 символов, обязательная для заполнения

[Patronymic] [varchar] (50) NULL default('-'),

--Строковая переменная, необязательная для заполнения с свойством значения по умолчанию

constraint [PK\_Workers] primary key clustered ([WorkerKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

)

go

Таблица «Статус выполнения»

CREATE TABLE [dbo].[WorkStatus]

(

[WorkStatusKey] [int] IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[Name] [varchar] (20) NOT NULL ,

--Cтроковая переменная с размерностью в 20 символов, обязательная для заполнения

constraint [PK\_WorkStatus] primary key clustered ([WorkStatusKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [UQ\_WorkStatus] unique ([Name]),

--Проверка на уникальность

)

go

Таблица «Тип работы»

CREATE TABLE [dbo].[WorkType]

(

[TypeWorkKey] [int] IDENTITY (1,1) ,

--Целочисленное обязательное для заполнения поле с автоинкрементом на единицу с начальным значением 1

[Name] [varchar] (20) NOT NULL ,

--Cтроковая переменная с размерностью в 20 символов, обязательная для заполнения

constraint [PK\_WorkType] primary key clustered ([TypeWorkKey] ASC) on [PRIMARY],

-- Ограничение на поле с типом первичный ключ, значения первичного ключа идут по возрастанию и находятся в кластаре в группе ключей PRIMARY

constraint [UQ\_WorkType] unique ([Name]),

--Проверка на уникальность

)

go

* + 1. Хранимые процедуры

Процедуры к таблице «Чеки»

create procedure [dbo].[Checks\_Insert]

@DataTime [datetime], @Cost [int]

as

insert into [dbo].[Checks] (DataTime, Cost)

values (@DataTime, @Cost)

go -- Процедура добавления записи

create procedure [dbo].[Checks\_Delete]

@CheckKey [int]

as

delete from [dbo].[Checks]

where CheckKey = @CheckKey

go -- Процедура удаления записи по номеру записи

Процедуры к таблице «Инвентарь»

create procedure [dbo].[Inventary\_Insert]

@Name [varchar] (50), @Position [int], @Quantity [int], @FullCost [int], @StorageTypeKey [int]

as

insert into [dbo].[Inventary] (Name, Position, Quantity, FullCost, StorageTypeKey)

values (@Name, @Position, @Quantity, @FullCost, @StorageTypeKey)

go -- Процедура добавления записи

create procedure [dbo].[Inventary\_Delete]

@InventaryKey [int]

as

delete from [dbo].[Inventary]

where InventaryKey = @InventaryKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[Inventary\_Update]

@InventaryKey [int], @Name [varchar] (50), @Position [int], @Quantity [int], @FullCost [int], @StorageTypeKey [int]

as

update [dbo].[Inventary] set

Name = @Name,

Position = @Position,

Quantity = @Quantity,

FullCost = @FullCost,

StorageTypeKey = @StorageTypeKey

where InventaryKey = @InventaryKey

go -- Процедура обновления записи по номеру записи

Процедуры к таблице «Материалы»

create procedure [dbo].[Materials\_Insert]

@OrderKey [int], @CheckKey [int], @InventaryKey [int]

as

insert into [dbo].[Materials] (OrderKey, CheckKey, InventaryKey)

values (@OrderKey, @CheckKey, @InventaryKey)

go -- Процедура добавления записи

create procedure [dbo].[Materials\_Delete]

@MaterialKey [int]

as

delete from [dbo].[Materials]

where MaterialKey = @MaterialKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[Materials\_Update]

@MaterialKey [int], @OrderKey [int], @CheckKey [int], @InventaryKey [int]

as

update [dbo].[Materials] set

OrderKey = @OrderKey,

CheckKey = @CheckKey,

InventaryKey = @InventaryKey

where MaterialKey = @MaterialKey

go -- Процедура обновления записи по номеру записи

Процедуры к таблице «Заказы»

create procedure [dbo].[Orders\_Insert]

@Name [varchar] (20), @Task [text], @WorkerKey [int], @TypeWorkKey [int], @WorkStatusKey [int], @CheckKey [int]

as

insert into [dbo].[Orders] (Name, Task, WorkerKey, TypeWorkKey, WorkStatusKey, CheckKey)

values (@Name, @Task, @WorkerKey, @TypeWorkKey, @WorkStatusKey, @CheckKey)

go -- Процедура добавления записи

create procedure [dbo].[Orders\_Delete]

@OrderKey [int]

as

delete from [dbo].[Orders]

where OrderKey = @OrderKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[Orders\_Update]

@OrderKey [int], @Name [varchar] (20), @Task [text], @WorkerKey [int], @TypeWorkKey [int], @WorkStatusKey [int], @CheckKey [int]

as

update [dbo].[Orders] set

Name = @Name,

Task = @Task,

WorkerKey = @WorkerKey,

TypeWorkKey = @TypeWorkKey,

WorkStatusKey = @WorkStatusKey,

CheckKey = @CheckKey

where OrderKey = @OrderKey

go -- Процедура обновления записи по номеру записи

Процедуры к таблице «Типы хранения»

create procedure [dbo].[StorageType\_Insert]

@Name [varchar] (20)

as

insert into [dbo].[StorageType] (Name)

values (@Name)

go -- Процедура добавления записи

create procedure [dbo].[StorageType\_Delete]

@StorageTypeKey [int]

as

delete from [dbo].[StorageType]

where StorageTypeKey = @StorageTypeKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[StorageType\_Update]

@StorageTypeKey [int], @Name [varchar] (20)

as

update [dbo].[StorageType] set

Name = @Name

where StorageTypeKey = @StorageTypeKey

go -- Процедура обновления записи по номеру записи

Процедуры к таблице «Сотрудники»

create procedure [dbo].[Workers\_Insert]

@Surname [varchar] (50), @Name [varchar] (50), @Patronymic [varchar] (50)

as

insert into [dbo].[Workers] (Surname, Name, Patronymic)

values (@Surname, @Name, @Patronymic)

go -- Процедура добавления записи

create procedure [dbo].[Workers\_Delete]

@WorkerKey [int]

as

delete from [dbo].[Workers]

where WorkerKey = @WorkerKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[Workers\_Update]

@WorkerKey [int], @Surname [varchar] (50), @Name [varchar] (50), @Patronymic [varchar] (50)

as

update [dbo].[Workers] set

Surname = @Surname,

Name = @Name,

Patronymic = @Patronymic

where WorkerKey = @WorkerKey

go -- Процедура обновления записи по номеру записи

Процедуры к таблице «Статус выполнения»

create procedure [dbo].[WorkStatus\_Insert]

@Name [varchar] (20)

as

insert into [dbo].[WorkStatus] (Name)

values (@Name)

go -- Процедура добавления записи

create procedure [dbo].[WorkStatus\_Delete]

@WorkStatusKey [int]

as

delete from [dbo].[WorkStatus]

where WorkStatusKey = @WorkStatusKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[WorkStatus\_Update]

@WorkStatusKey [int], @Name [varchar] (20)

as

update [dbo].[WorkStatus] set

Name = @Name

where WorkStatusKey = @WorkStatusKey

go -- Процедура обновления записи по номеру записи

Процедуры к таблице «Тип работы»

create procedure [dbo].[WorkType\_Insert]

@Name [varchar] (20)

as

insert into [dbo].[WorkType] (Name)

values (@Name)

go -- Процедура добавления записи

create procedure [dbo].[WorkType\_Delete]

@TypeWorkKey [int]

as

delete from [dbo].[WorkType]

where TypeWorkKey = @TypeWorkKey

go -- Процедура удаления записи по номеру записи

create procedure [dbo].[WorkType\_Update]

@TypeWorkKey [int], @Name [varchar] (20)

as

update [dbo].[WorkType] set

Name = @Name

where TypeWorkKey = @TypeWorkKey

go -- Процедура обновления записи по номеру записи

* + 1. Представления

Представление заявок на работу

create or alter view [dbo].[Order\_View] ("ФИО сотрудника", "Наименование", "Статус выполнения", "Чек")

as

select

[dbo].[Workers].[Surname] + ' ' + substring([dbo].[Workers].[Name], 1, 1) + '.' + substring([dbo].[Workers].[Patronymic], 1, 1) + '.',

--Получение ФИО сотрудника

[dbo].[Orders].[Name], --Получение наименования

[dbo].[WorkStatus].[Name], --Получение статуса

[dbo].[Checks].[CheckKey] --Получение номера чека

from [dbo].[Orders]

inner join [dbo].[Workers] on [NameWorkers] = [WorkerKey]

inner join [dbo].[Orders] on [NameOrders] = [OrderKey]

inner join [dbo].[WorkStatus] on [NameStatus] = [WorkStatusKey]

inner join [dbo].[Checks] on [KeyCheck] = [CheckKey]

go

Представление материалов

create or alter view [dbo].[Materials\_View] ("Код материалов", "Наименование", "Код инвентаря")

as

select

[dbo].[Materials].[MaterialKey], --Получение кода материалов

[dbo].[Orders].[Name], --Получение имени заказа

[dbo].[Inventary].[InventaryKey], --Получение кода инвентаря

from [dbo].[Materials]

inner join [dbo].[Materials] on [KeyMaterials] = [MaterialKey]

inner join [dbo].[Orders] on [KeyOrders] = [OrderKey]

inner join [dbo].[Inventary] on [KeyInventary] = [InventaryKey]

go

* + 1. Функции

Функция получения ТЗ

create or alter function [dbo].[Description\_Func](@OrderKey [int]) --Создание функции

returns [text] --Возвращаемый тип

with execute as caller

as

begin

declare @TextOrder [text] = (select [dbo].[Orders].[Task] from [dbo].[Orders]

where [dbo].[Orders].[OrderKey] = @OrderKey)

return(@TextOrder) --Возвращение значения

end

go

select [dbo].[Description\_Func](1) --Вызов функции

* + 1. Триггеры

Триггер удаления заказа при удалении чека (возврат)

create or alter trigger [dbo].[CheckDeletedTrigger] on [dbo].[Checks]

after delete --Нужные операции для выполнения триггера

as

delete from [dbo].[Orders] where CheckKey = SELECT CheckKey

FROM DELETED --удаляем заказ по ключу удаленного чека

go

1. ВЫВОД

В процессе выполнения данной практической работы получены знания о создании таблиц, процедур, представлений, функций и триггеров в БД SQL. Работа базы данных была приведена к удобному использованию. Плюсы Management Studio: работа с элементами базы данных осуществляется с помощью программного кода, а также при взаимодействиях с элементами в иерархии при помощи интерфейса; минусы: неконкретное отображение ошибок базы данных, постоянная нужда в соединении с сервером. При выполнении работы была использована MS SQL Server Management Studio, MS Word, ERwin Data Modeler r7.

1. СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ
2. <https://icvao.ru/> - описание предметной области
3. Щанников И.М. Лекции 2020 – Лекции по предмету
4. <https://docs.microsoft.com/ru-ru/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15> – описание типов данных в SQL
5. <http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab08.html> - уроки по работе с CA ErWin DataModeller