

ГУАП
КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ст. преп.

Путилова Н.В.

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

vk.com/club152685050
vk.com/id446425943

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1

СОЗДАНИЕ БД

по дисциплине: Проектирование баз данных

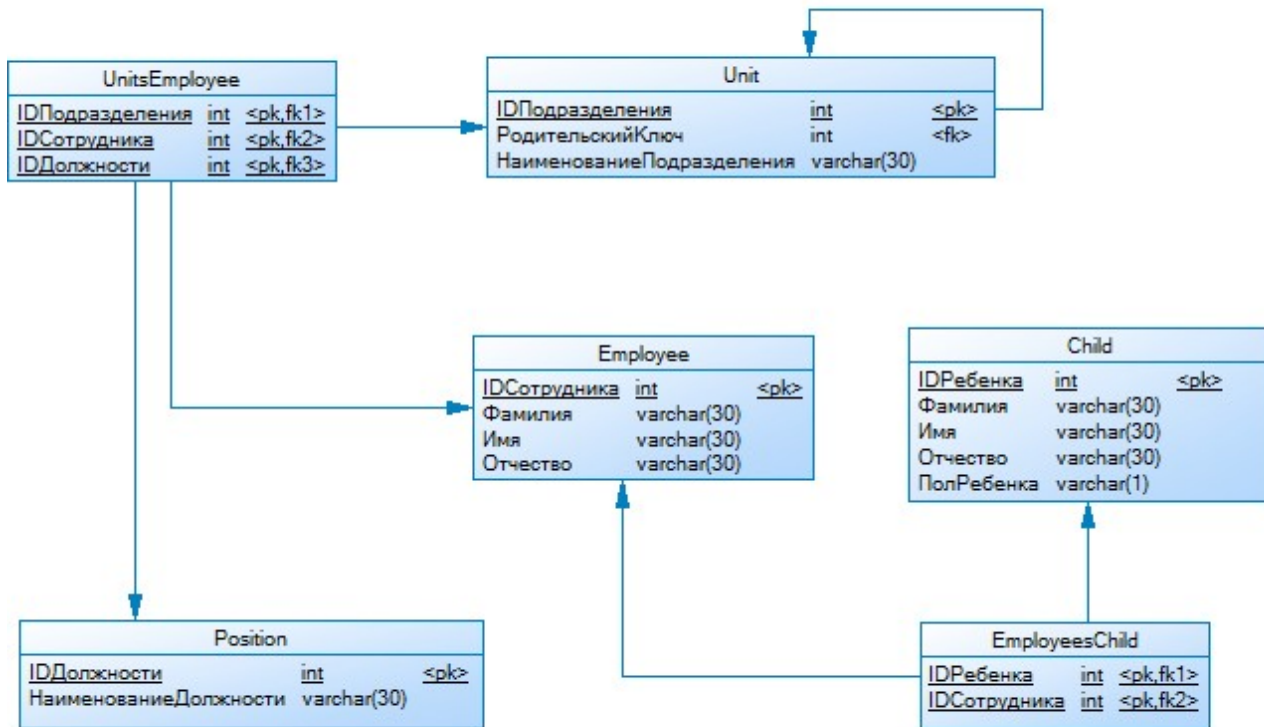
РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

подпись, дата

инициалы, фамилия

1. Схема БД.



2. Скрипт SQL для создания таблиц.

```
/*=====*/
/* DBMS name:      Microsoft SQL Server 2008      */
/* Created on:     08.03.2017 19:21:19             */
/*=====*/
create database mikel
go
use mikel
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('EmployeesChild') and o.name = 'FK_EMPLOYEE_REFERENCE_CHILD')
alter table EmployeesChild
  drop constraint FK_EMPLOYEE_REFERENCE_CHILD
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('EmployeesChild') and o.name = 'FK_EMPLOYEE_REFERENCE_EMPLOYEE')
alter table EmployeesChild
  drop constraint FK_EMPLOYEE_REFERENCE_EMPLOYEE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('Unit') and o.name = 'FK_UNIT_REFERENCE_UNIT')
alter table Unit
```

```

drop constraint FK_UNIT_REFERENCE_UNIT
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('UnitsEmployee') and o.name = 'FK_UNITSEMP_REFERENCE_UNIT')
alter table UnitsEmployee
  drop constraint FK_UNITSEMP_REFERENCE_UNIT
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('UnitsEmployee') and o.name = 'FK_UNITSEMP_REFERENCE_EMPLOYEE')
alter table UnitsEmployee
  drop constraint FK_UNITSEMP_REFERENCE_EMPLOYEE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('UnitsEmployee') and o.name = 'FK_UNITSEMP_REFERENCE_POSITION')
alter table UnitsEmployee
  drop constraint FK_UNITSEMP_REFERENCE_POSITION
go

if exists (select 1
  from sysobjects
  where id = object_id('Child')
  and type = 'U')
drop table Child
go

if exists (select 1
  from sysobjects
  where id = object_id('Employee')
  and type = 'U')
drop table Employee
go

if exists (select 1
  from sysobjects
  where id = object_id('EmployeesChild')
  and type = 'U')
drop table EmployeesChild
go

if exists (select 1
  from sysobjects
  where id = object_id('Position')
  and type = 'U')
drop table Position
go

if exists (select 1
  from sysobjects
  where id = object_id('Unit')
  and type = 'U')
drop table Unit
go

if exists (select 1
  from sysobjects
  where id = object_id('UnitsEmployee')
  and type = 'U')
drop table UnitsEmployee
go

/*=====*/
/* Table: Child */
/*=====*/

```

```

create table Child (
    IDРебенка          int          not null,
    Фамилия            varchar(30)   null,
    Имя                varchar(30)   null,
    Отчество           varchar(30)   null,
    ПолРебенка         varchar(1)    null, check (ПолРебенка in ('М', 'Ж')),
    primary key (IDРебенка)
)
go

/*=====*/
/* Table: Employee */
/*=====*/
create table Employee (
    IDСотрудника       int          not null,
    Фамилия            varchar(30)   null,
    Имя                varchar(30)   null,
    Отчество           varchar(30)   null,
    primary key (IDСотрудника)
)
go

/*=====*/
/* Table: EmployeesChild */
/*=====*/
create table EmployeesChild (
    IDРебенка          int          not null,
    IDСотрудника       int          not null,
    primary key (IDРебенка, IDСотрудника),
    foreign key (IDСотрудника)
        references Employee (IDСотрудника)
        on delete cascade
        on update cascade,
    foreign key (IDРебенка)
        references Child (IDРебенка)
        on delete cascade
        on update cascade
)
go

/*=====*/
/* Table: Position */
/*=====*/
create table Position (
    IDДолжности        int          not null,
    НаименованиеДолжности varchar(30) null,
    primary key (IDДолжности)
)
go

/*=====*/
/* Table: Unit */
/*=====*/
create table Unit (
    IDПодразделения    int          not null,
    РодительскийКлюч   int          null,
    НаименованиеПодразделения varchar(30) null,
    primary key (IDПодразделения),
    foreign key (РодительскийКлюч)
        references Unit (IDПодразделения)
        on delete no action
        on update no action
)
go

/*=====*/
/* Table: UnitsEmployee */
/*=====*/
create table UnitsEmployee (

```

```

IDПодразделения      int                not null,
IDСотрудника          int                not null,
IDДолжности           int                not null,
primary key (IDПодразделения, IDСотрудника, IDДолжности),
foreign key (IDДолжности)
references Position (IDДолжности)
on delete cascade
on update cascade,
foreign key (IDСотрудника)
references Employee (IDСотрудника)
on delete cascade
on update cascade,
foreign key (IDПодразделения)
references Unit (IDПодразделения)
on delete cascade
on update cascade
)
go

```

3. Описание структуры таблиц, ограничений на значения данных, ссылочной целостности, реализованных в БД.

- 3.1. **Таблица Child** содержит информацию о детях: уникальный ID ребенка (первичный ключ) исключительно числового типа, фамилию, имя и отчество ребенка, которые имеют символьный тип и имеют длину не более 30-ти символов и пол ребенка, который является строкой длиной в 1 символ и содержит проверку на данные (либо М, либо Ж). Данная таблица не имеет внешних ключей.
- 3.2. **Таблица Employee** содержит информацию о сотрудниках: уникальный ID сотрудника (первичный ключ) исключительно числового типа и фамилию, имя и отчество сотрудника, которые имеют символьный тип и имеют длину не более 30-ти символов. Данная таблица не имеет внешних ключей.
- 3.3. **Таблица EmployeesChild** представляет собой связующую таблицу из двух полей: ID сотрудника и ID ребенка, оба эти поля целочисленные и являются внешними первичными ключами для данной таблицы. Ссылочная целостность выстроена таким образом, что при изменении первичного ключа в таблицах Employee и Child (при удалении каких-либо сотрудников или детей сотрудников) будет удаляться и вся информация в отношении сотрудника (Employee) и ребенка (Child) и информация о несуществующем сотруднике и ребенке будет уже не актуальна. Также при изменении первичных ключей в таблицах Employee и Child будет изменена информация в таблице EmployeesChild.
- 3.4. **Таблица Position** содержит информацию о должностях: уникальный ID должности (первичный ключ) исключительно числового типа и наименование должности, которое имеет символьный тип и имеют длину не более 30-ти символов. Данная таблица не имеет внешних ключей.
- 3.5. **Таблица Unit** содержит информацию о подразделениях: уникальный ID подразделения (первичный ключ) исключительно числового типа, наименование подразделения, которое имеет символьный тип и имеют длину не более 30-ти символов и также имеет внешний ключ – родительский ключ. Ссылочная целостность выстроена таким образом, что изменение первичного ключа либо его удаление по умолчанию запрещено, так как нельзя удалить подразделение, если в нем есть другие подразделения.
- 3.6. **Таблица UnitsEmployee** представляет собой связующую таблицу из трех полей: ID сотрудника, ID должности и ID подразделения, все эти поля целочисленные и являются внешними первичными ключами для данной таблицы. Ссылочная целостность выстроена таким образом, что

при удалении первичного ключа в таблицах Employee, Unit и Position (при удалении каких-либо сотрудников, должностей) будет удаляться и вся информация в отношении сотрудника (Employee) , должности (Child) и подразделения(Unit). Также при изменении первичных ключей в таблицах Employee, Unit и Position будет изменена информация в таблице UnitsEmployee.

vk.com/club152685050

vk.com/id446425943