CREATE DATABASE University;

go

use University;

CREATE TABLE Department(

FAC\_ID int identity primary key,

FAC\_NAME nvarchar(20) NOT NULL

CHECK(FAC\_NAME LIKE '%[А-Яа-я]%'),

RATING int

)

GO

use University;

CREATE TABLE STUDENT(

ST\_ID bigint IDENTITY NOT NULL

CONSTRAINT PK\_STUDENT PRIMARY KEY

CHECK(ST\_ID BETWEEN 0 AND 9999999999),

Surname nvarchar(20) NOT NULL

CHECK(Surname LIKE '%[А-Яа-я]%'),

Name nvarchar(15) NOT NULL

CHECK(Name LIKE '%[А-Яа-я]%'),

STIPEND bigint NULL,

KURS int NOT NULL,

CITY nvarchar(30) NOT NULL

CHECK (CITY LIKE '%[А-Яа-я]%'),

BIRTHDAY datetime NOT NULL

CHECK(BIRTHDAY BETWEEN

CONVERT(datetime, '1/1/1753', 103) and CONVERT(datetime, '1/1/2018', 103)),

FAC\_ID int NOT NULL CONSTRAINT FAC\_ID

FOREIGN KEY REFERENCES Department(FAC\_ID)

ON UPDATE CASCADE

ON DELETE CASCADE

)

GO

use University;

CREATE TABLE LECTURE(

LECT\_ID bigint IDENTITY NOT NULL

CONSTRAINT PK\_LECTURE PRIMARY KEY

CHECK(LECT\_ID BETWEEN 0 AND 9999999999),

SURNAME nvarchar(20) NOT NULL

CHECK(SURNAME LIKE '%[А-Яа-я]%'),

NAME nvarchar(15) NOT NULL

CHECK(NAME LIKE '%[А-Яа-я]%'),

FAC\_ID int NOT NULL CONSTRAINT LECTURE\_FAC\_ID

FOREIGN KEY REFERENCES Department(FAC\_ID)

ON UPDATE CASCADE

ON DELETE CASCADE

)

GO

use University;

CREATE TABLE SUBJECT (

SUBJ\_ID bigint IDENTITY NOT NULL

CONSTRAINT PK\_SUBJECT PRIMARY KEY

CHECK(SUBJ\_ID BETWEEN 0 AND 9999999999),

SUBJ\_NAME nvarchar(20) NOT NULL

CHECK(SUBJ\_NAME LIKE '%[А-Яа-я]%'),

HOUR int NOT NULL,

SEMESTR int NOT NULL

)

GO

use University;

CREATE TABLE SUBJECTRELATIONS (

LECT\_ID bigint NOT NULL CONSTRAINT SUBJECTRELATIONS\_LECT\_ID

FOREIGN KEY REFERENCES LECTURE(LECT\_ID)

ON UPDATE CASCADE

ON DELETE CASCADE,

SUBJ\_ID bigint NOT NULL CONSTRAINT SUBJECTRELATIONS\_SUBJ\_ID

FOREIGN KEY REFERENCES SUBJECT(SUBJ\_ID)

ON UPDATE CASCADE

ON DELETE CASCADE

)

GO

---Факультеты------------------------------------------------------------------------

use University;

insert into Department values ('Математика',10)

insert into Department values('Физика',8)

insert into Department values('Химия',7)

insert into Department values('История',8)

insert into Department values('Экономика',9)

insert into Department values('Маркетинг',10)

insert into Department values('ВЭД',9)

insert into Department values('ИТ-технологии',10)

insert into Department values('ВЭД\_Торговля',7)

insert into Department values('ВЕБ\_дизайн',8)

GO

---Студенты--------------------------------------------------------------------------

use University;

insert into STUDENT values('Супрунова','Наталья',0,2,'Донецк','1986-04-09',2)

insert into STUDENT values('Николаев','Евгений',780,4,'Одесса','1986-13-10',12)

insert into STUDENT values('Леонова','Елена',0,1,'Запорожье','1986-24-05',5)

insert into STUDENT values('Жлобенко','Дмитрий',0,2,'Днепропетровск','1986-04-01',6)

insert into STUDENT values('Верещага','Екатерина',540,2,'Донецк','1985-30-06',4)

insert into STUDENT values('Степанова','Екатерина',540,2,'Львов','1985-20-08',3)

insert into STUDENT values('Ткаченко','Антон',0,1,'Николаев','1986-01-12',7)

insert into STUDENT values('Морозов','Александр',0,2,'Ужгород','1986-23-01',2)

insert into STUDENT values('Битюкова','Инна',0,3,'Тернополь','1986-15-03',8)

insert into STUDENT values('Левена','Любовь',0,2,'Ивано-Франковск','1986-09-12',9)

insert into STUDENT values('Дорожко','Евгений',780,2,'Сумы','1986-04-05',11)

insert into STUDENT values('Типаев','Игорь',780,2,'Харьков','1990-02-12',2)

insert into STUDENT values('Кущ','Игорь',1250000,4,'Киев','1971-06-08',8)

insert into STUDENT values('Висковатых','Наталина',5250000,4,'Киев','1986-14-02',8)

insert into STUDENT values('Висковатых','Елена',2350630,1,'Гонконг','1988-2-2',8)

GO

---Лекторы---------------------------------------------------------------------------

use University;

insert into LECTURE values('Волкова','Елена',2)

insert into LECTURE values('Дорина','Мария',3)

insert into LECTURE values('Становая','Евгения',5)

insert into LECTURE values('Жданов','Михаил',4)

insert into LECTURE values('Ставриков','Николай',6)

insert into LECTURE values('Дроздь','Дарина',7)

insert into LECTURE values('Коликова','Светлана',7)

insert into LECTURE values('Мудрая','Марина',8)

insert into LECTURE values('Феолетовая','Екатерина',11)

insert into LECTURE values('Лапухова','Степанида',12)

insert into LECTURE values('Дарун','Николай',9)

GO

---Предметы--------------------------------------------------------------------------

use University;

insert into SUBJECT values('Математика',15,1)

insert into SUBJECT values('Астрономия',20,2)

insert into SUBJECT values('Производства',30,1)

insert into SUBJECT values('Физика',50,1)

insert into SUBJECT values('Экономика',40,2)

insert into SUBJECT values('Политология',30,2)

insert into SUBJECT values('Дизайн',40,1)

insert into SUBJECT values('Мэнеджмент',30,2)

insert into SUBJECT values('Маркетинг',140,1)

insert into SUBJECT values('История',130,2)

insert into SUBJECT values('Химия',140,1)

insert into SUBJECT values('ВЭД\_торговля',150,1)

insert into SUBJECT values('Управление',140,2)

go

--Отношения-------------------------------------------------------------------------

use University;

insert into SUBJECTRELATIONS values(15,1)

----a. подсчитывает и показывает количество студентов, получающих стипендию, определенного размера.

--- При отсутствии таковых вы-дает соответствующее сообщение.

ALTER PROC FindStipendia --назвали процедуру

@stipendia int--добавили аргумент

AS

BEGIN

IF NOT EXISTS ( ------ если не нашли IF NOT EXISTS

SELECT COUNT (S.ST\_ID) [#]

FROM dbo.STUDENT AS S

WHERE S.STIPEND=@stipendia--добавили условие

)

select 'Not have this stipendia'

ELSE --- если все нашли ELSE

SELECT COUNT (S.ST\_ID) [#]

FROM dbo.STUDENT AS S

WHERE S.STIPEND=@stipendia--добавили условие

END

go

exec FindStipendia '5250000'

exec FindStipendia '0'

exec FindStipendia '540'

exec FindStipendia '10'

--b. Подсчитывает и показывает количество студентов, живущих в определенном городе.

-- При отсутствии таковых выдает соответ-ствующее сообщение.

CREATE PROC FindAdress --назвали процедуру

@adress nvarchar (30)--добавили аргумент

AS

BEGIN

IF NOT EXISTS ( ------ если не нашли IF NOT EXISTS

SELECT COUNT (S.CITY) [Количество студентов]

FROM dbo.STUDENT AS S

WHERE S.CITY=@adress--добавили условие

)

select 'Not have this city'

ELSE --- если все нашли ELSE

SELECT COUNT (S.CITY) [Количество студентов]

FROM dbo.STUDENT AS S

WHERE S.CITY=@adress--добавили условие

END

go

exec FindAdress 'Донецк'

exec FindAdress 'Харьков'

exec FindAdress 'Киев'

exec FindAdress 'Бердянск'

--e. Изменение оценок всем студентам указанного курса (увеличение на 1).

--f. Удаление преподавателей, не читающих ни одного предмета.