Московский авиационный институт

(Национальный исследовательский университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторная работа 1-3

по курсу «Базы данных»

Студенты: Макаренкова В.М.,

Вахрамян К.О.

Группа: 80-306Б-18

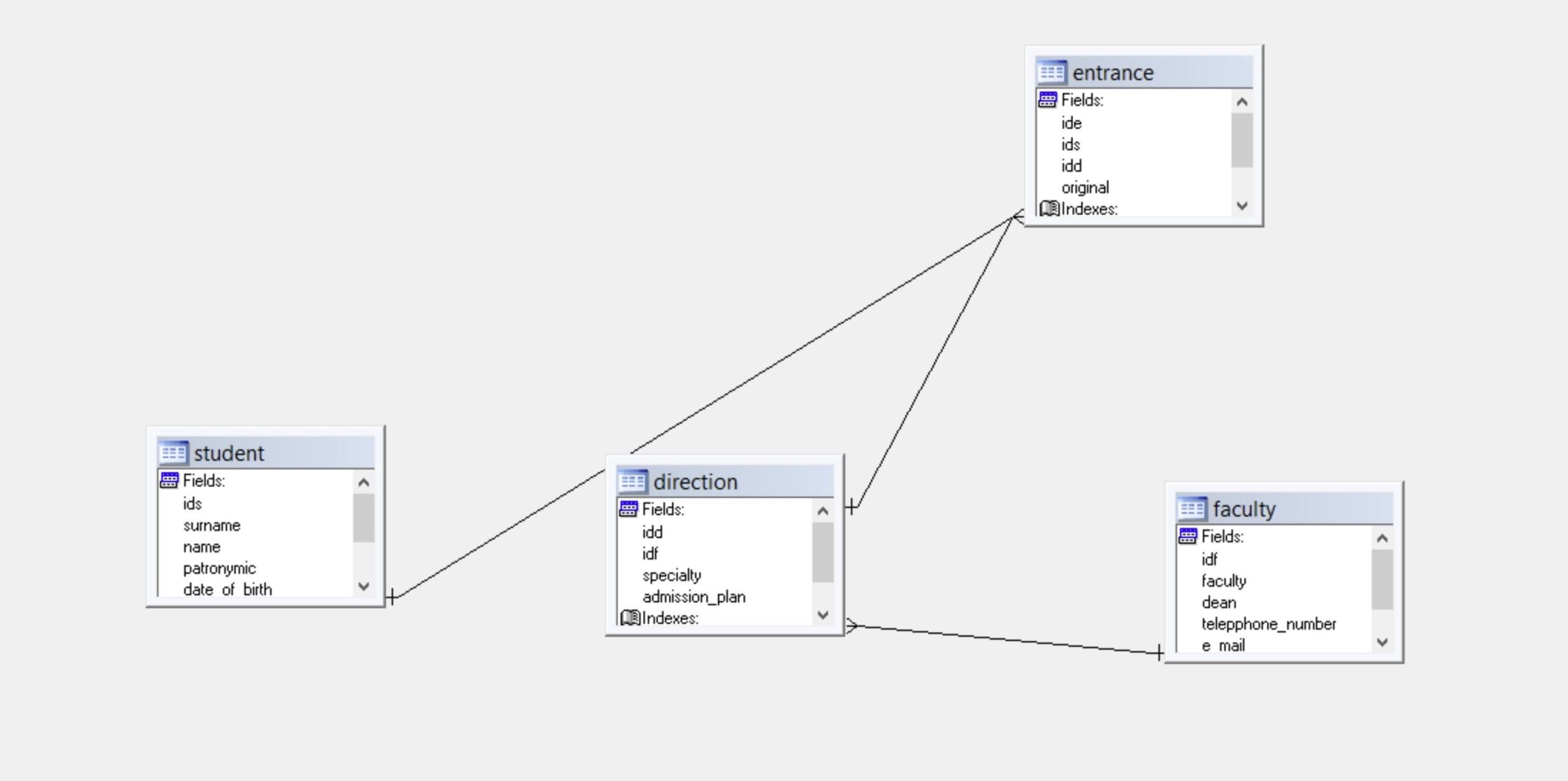
Преподаватель: Кузнецова Е.В.

Оценка:

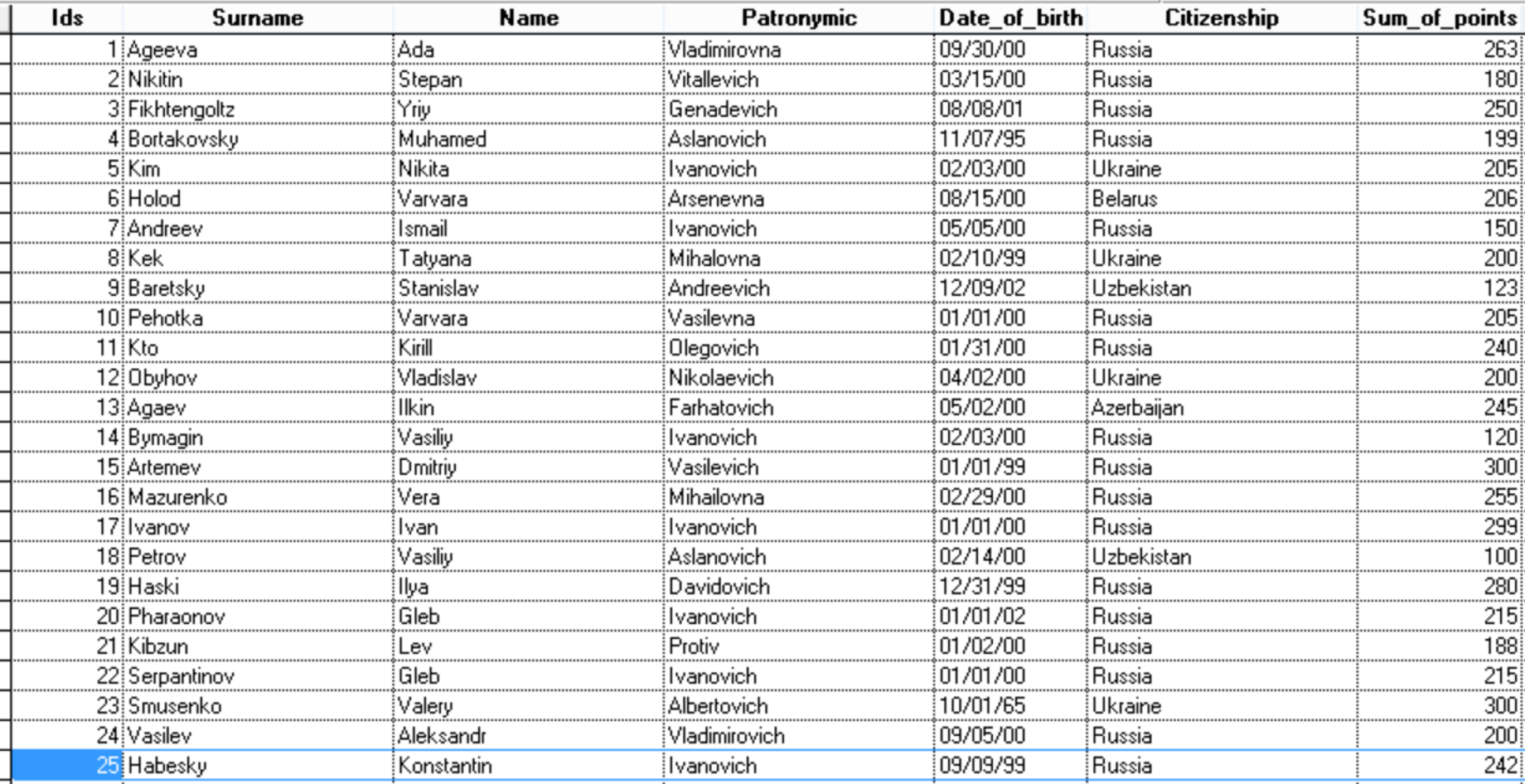
Москва, 2020

**Задание: БД приемной комиссии ВУЗа**

**База данных в FoxPro**



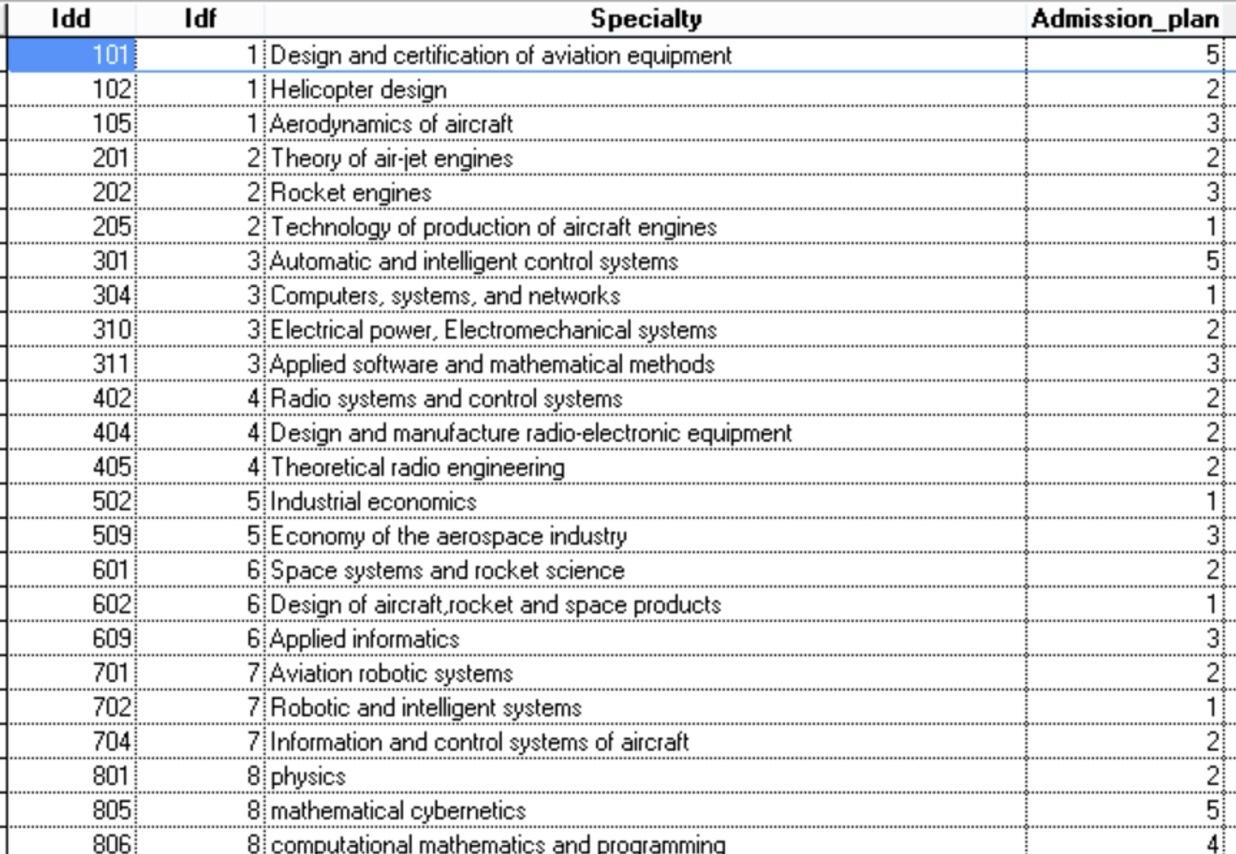
1. Таблица анкет абитуриентов : id, ФИО, дата рождения, гражданство, сумма баллов



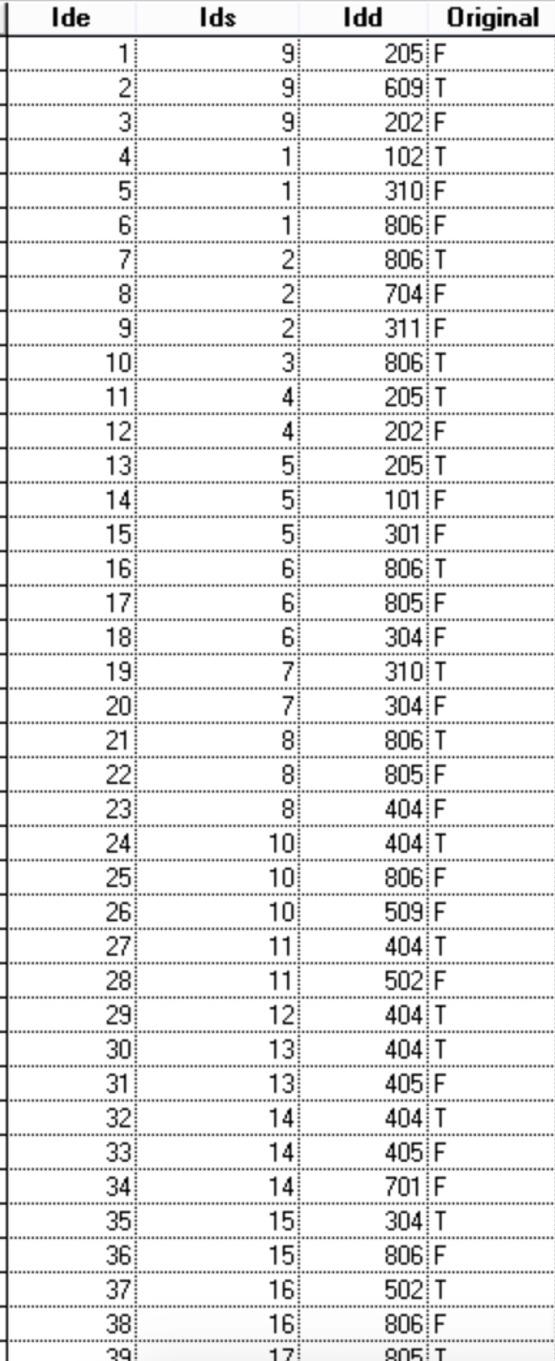
1. Таблица факультетов ВУЗа: id, название, декан, телефон, почта



3) Таблица направлений : id, id факультета, название, план приема(количество мест)

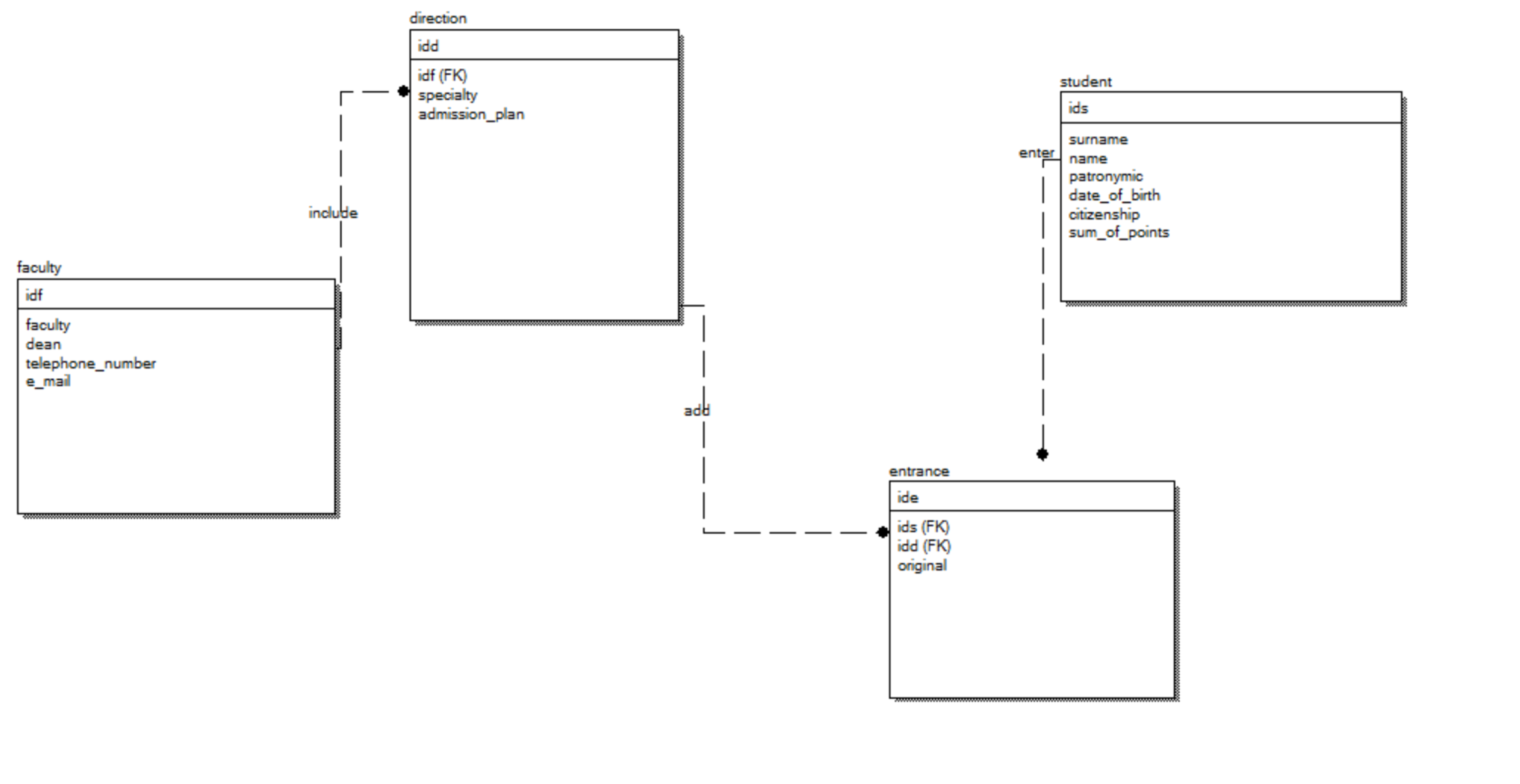


4) Таблица подачи документов абитуриентов на специальности:id, id студента, id специальности, подан ли оригинал аттестата

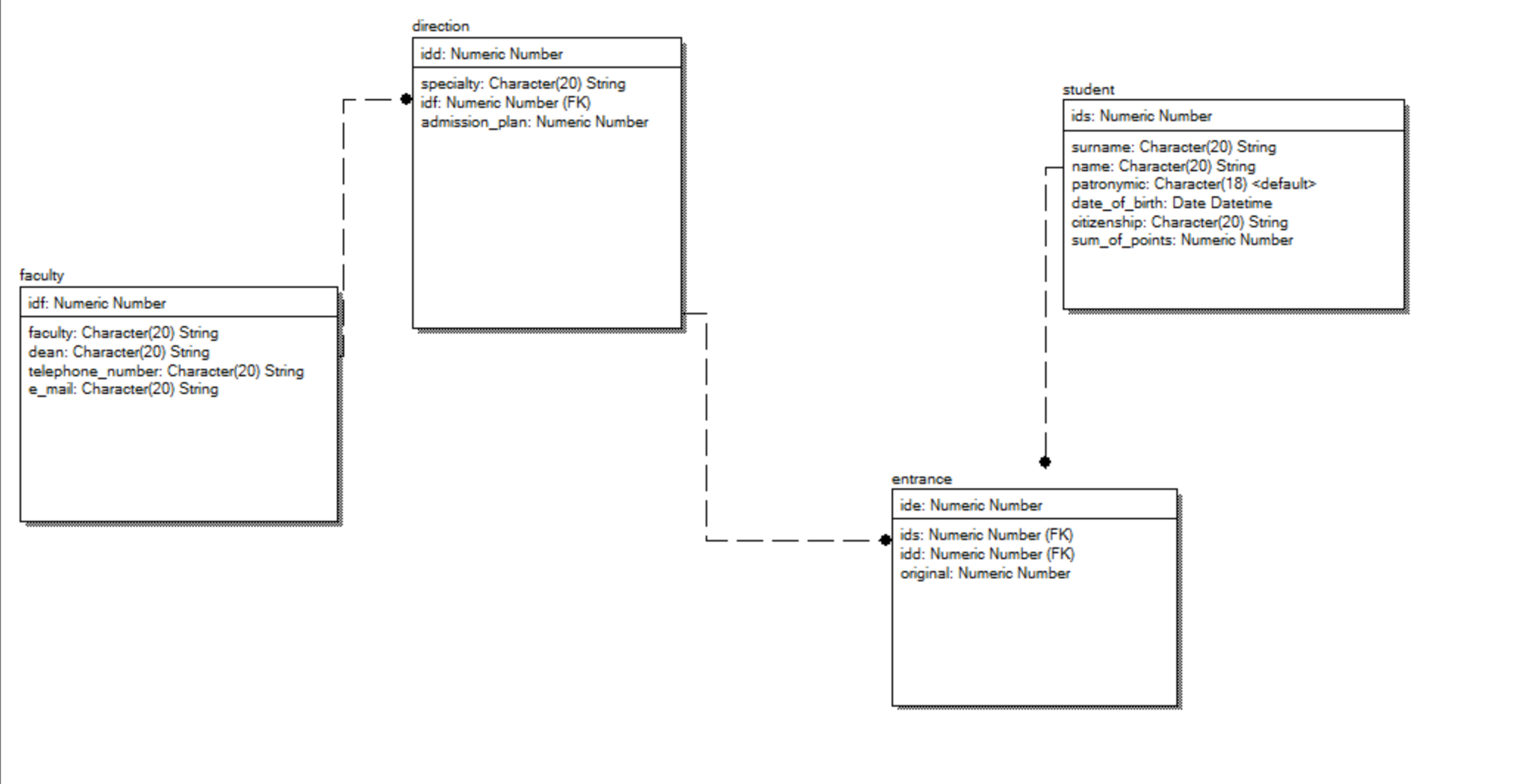


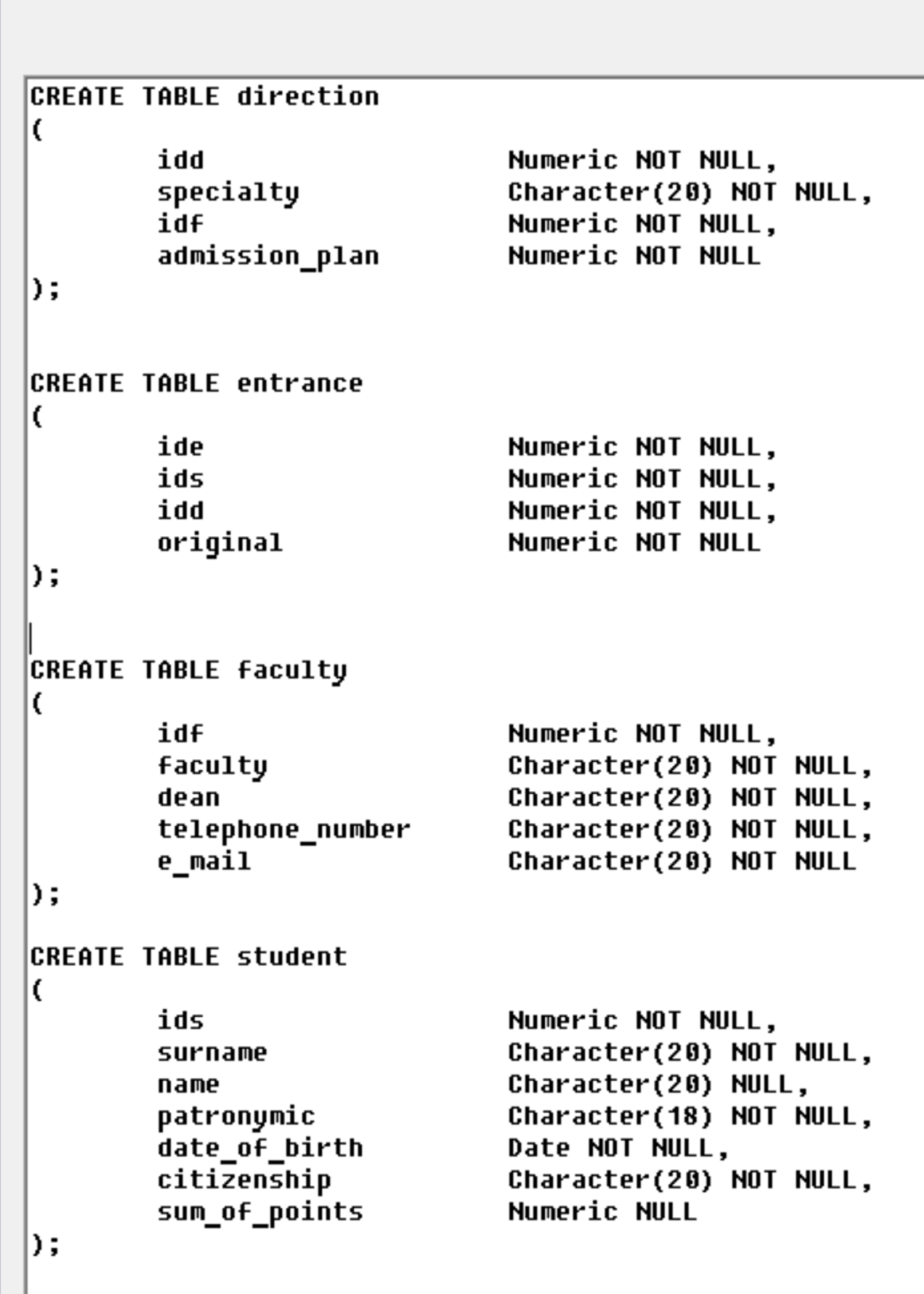
**Модель ErWin**

1. **Логическая модель**



1. **Физическая модель**





**Сложные запросы**

1. **Пары студентов одного направления для студентов заданного года рождения.**

SELECT student.ids, student.surname, student.date\_of\_birth, entrance.idd;

FROM student, entrance INTO CURSOR tmp;

where student.ids = entrance.ids

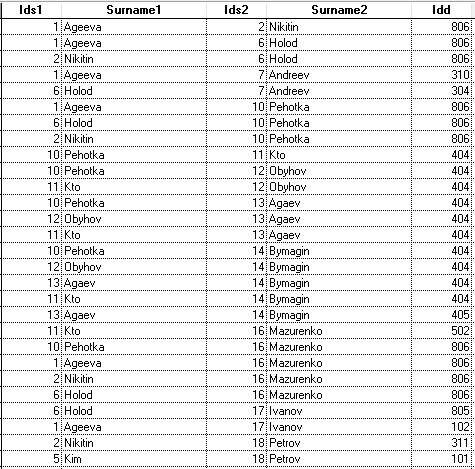
SELECT a.ids as ids1, a.surname as surname1, b.ids as ids2, b.surname as surname2, a.idd;

FROM tmp as a, tmp as b;

where(((a.date\_of\_birth between {^2000/01/01} AND {^2000/12/31});

AND b.date\_of\_birth between {^2000/01/01} AND {^2000/12/31});

AND a.ids < b.ids AND a.idd=b.idd)



1. **Кто из студентов заданного года рождения не подавал документы на заданное направление.**

select ids, surname from student;

where (date\_of\_birth between {^2000/01/01} AND {^2000/12/31});

and ids not in (select ids from entrance where idd = 806)



1. **Определить факультет, на которое подано максимальное количество док-тов в оригинале.**

SELECT direction.idf, faculty.faculty, COUNT(entrance.original) as orig;

FROM direction, entrance, faculty INTO CURSOR tmp;

WHERE entrance.idd= direction.idd AND entrance.original = .T. and direction.idf = faculty.idf;

group BY direction.idf, faculty.faculty

SELECT tmp.idf, tmp.faculty, tmp.orig FROM tmp;

where orig = (select MAX(orig) FROM tmp)



1. **Определить направление, на котором абитуриентов заданного возраста выше среднего**

select entrance.idd, direction.specialty;

FROM student, entrance, direction INTO CURSOR tmp;

where student.ids = entrance.ids and entrance.idd = direction.idd and YEAR(DATE()) - year(date\_of\_birth) = 20

SELECT idd, specialty, COUNT(specialty) FROM tmp;

into CURSOR tmp2;

group BY idd, specialty

SELECT \* FROM tmp2;

where cnt\_specialty > (SELECT AVG(tmp2.cnt\_specialty) FROM tmp2)



1. **Определить для каждого направления каждого факультета количество студентов, набравших баллов выше среднего.**

SELECT direction.idf, faculty.faculty, entrance.idd, entrance.ids, direction.specialty, student.sum\_of\_points;

FROM direction, entrance, student, faculty;

INTO CURSOR tmp;

where direction.idd = entrance.idd ;

AND entrance.ids = student.ids ;

and direction.idf = faculty.idf ;

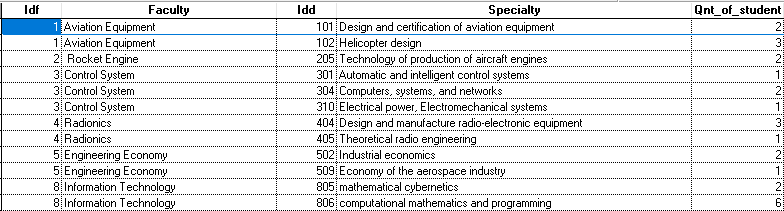
and entrance.idd = direction.idd

SELECT idf , faculty, idd, specialty, COUNT(tmp.ids) as qnt\_of\_student;

FROM tmp;

WHERE tmp.sum\_of\_points > (SELECT AVG(tmp.sum\_of\_points) FROM tmp );

GROUP BY idf , faculty, idd, specialty



1. **Определить для заданного фак-тета заданного направления проходной балл в текущем году.**

SELECT direction.idf, faculty.faculty, entrance.idd, direction.specialty, student.sum\_of\_points, student.surname;

FROM student, entrance, direction, faculty;

into CURSOR tmp;

where entrance.idd = 404 and direction.idf = 4 and faculty.idf = 4 and direction.idd = 404 AND entrance.original = .T. and entrance.ids = student.ids;

order BY student.sum\_of\_points DESC

SELECT RECNO() as idt, \* FROM tmp;

into CURSOR tmp

select \* from tmp;

into cursor tmp;

where idt <= (select direction.admission\_plan from direction;

where idd = 404)

select idf, faculty, idd, specialty, sum\_of\_points, surname from (select \*, ;

(select count(\*) from tmp as t2 where t2.sum\_of\_points <= t1.sum\_of\_points) as num;

from tmp as t1) tmp;

where num = 1



1. **Придумать и реализовать запрос на предикаты All, Any, Exist.**

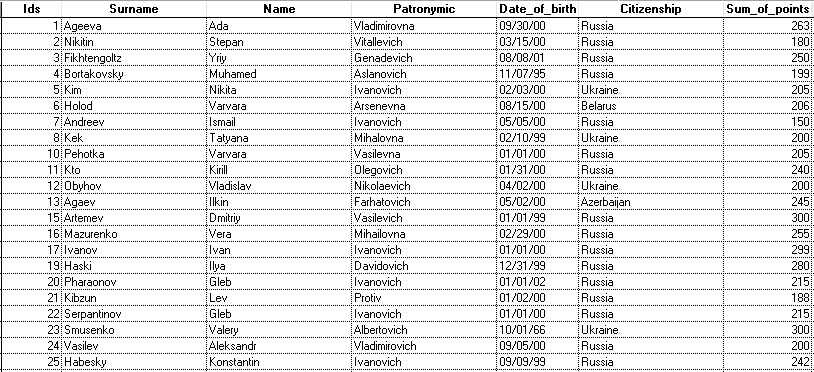
**Предикат ALL**

Выбрать всех студентов, набравших баллов больше, чем любой студент из Узбекистана.

select \* from student;

where student.sum\_of\_points > ALL(select student.sum\_of\_points from student;

where student.citizenship = 'Uzbekistan')



**предикат EXISTS**

Выбрать студентов, подавших заявление на 806 кафедру.

select \* from student;

where exists(select entrance.ids from entrance;

where student.ids = entrance.ids and entrance.idd = 806)



**предикат ANY**

Аналогичный запрос с предикатом ANY

select \* from student;

where student.ids = ANY(select entrance.ids from entrance where entrance.idd = 806)

