Мамонов Антон 2ИСиП-19-1

Создаем таблицы

	T	
create table university	create table subject	create table lecturer
(((
univ_id int NOT NULL	subj_id int NOT NULL	lecturer_id int NOT NULL
auto_increment,	auto_increment,	auto_increment,
univ_name varchar(10),	subj_name varchar(20) NOT	surname varchar(20) NOT
rating int,	NULL,	NULL,
city varchar(20),	hours int,	name varchar(20) NOT NULL,
PRIMARY KEY (univ_id)	semestr int,	city varchar(20),
);	PRIMARY KEY (subj_id)	univ_id int,
/');	PRIMARY KEY (lecturer_id),
	<i>)</i> ,	FOREIGN KEY (univ_id)
		REFERENCES university(univ_id)
		on delete cascade on update
		cascade
):
amonto table avera montre	create table student	create table subj_lect
create table exam_marks		,
(((
exam_id int NOT NULL	student_id int NOT NULL	id int NOT NULL
auto_increment,	auto_increment,	auto_increment,
student_id INT NOT NULL,	surname varchar(20) NOT	lecturer_id int NOT NULL,
subj_id int NOT NULL,	NULL,	subj_id int NOT NULL,
mark TINYINT(4),	name varchar(20) NOT NULL,	PRIMARY KEY (id),
exam_date DATE,	stipend decimal(6,2),	FOREIGN KEY (lecturer_id)
PRIMARY KEY (exam_id),	kurs tinyint(1),	REFERENCES
FOREIGN KEY (student_id)	city varchar(20),	lecturer(lecturer_id) on delete
REFERENCES student(student_id)	birthday date,	cascade on update cascade,
on delete cascade on update	univ_id int,	FOREIGN KEY (subj_id)
cascade,	PRIMARY KEY (student_id),	REFERENCES subject(subj_id) on
FOREIGN KEY (subj_id)	FOREIGN KEY (univ_id)	delete cascade on update cascade
REFERENCES subject(subj_id) on	REFERENCES university(univ_id));
delete cascade on update cascade	on delete cascade on update	
);	cascade	
);	
	1 //	

Заполняем таблицы данными

INSERT INTO `university`	INSERT INTO `subjects` (`SUBJ_ID`,	INSERT INTO `subj_lect`
(`UNIV_ID`, `UNIV_NAME`,	`SUBJ_NAME`, `HOURS`, `SEMESTR`)	(`LECTURER_ID`,
`RATING`, `CITY`) VALUES	VALUES	`SUBJ_ID`) VALUES
(1, 'Академия н', 542, 'Москва'),	(1, 'Биология', 119, 1),	(1, 2),
(2, 'Московский', 470, 'Москва'),	(2, 'Химия', 65, 4),	(1, 5),
(3, 'Российская', 429, 'Москва'),	(3, 'Сочинение', 95, 2),	(2,3),
(4, 'Финансовая', 421, 'Москва'),	(4, 'География', 74, 2),	(2,5),
(5, 'Московский', 392, 'Москва'),	(5, 'Собеседование', 102, 4),	(3, 1),
(6, 'Самарский ', 383, 'Самара'),	(6, 'Изложение', 107, 4),	(3,4),
(7, 'Московская', 381, 'Москва'),	(7, 'Физика', 69, 1),	(3,5),
(10, 'ВГУ', 296, 'Воронеж'),	(8, 'История России', 73, 3),	(4, 5),
(11, 'НГУ', 345, 'Новосибирск'),	(9, 'Литература', 104, 3),	(4, 6),
(14, 'БГУ', 326, 'Белгород'),	(10, 'Информатика', 56, 1),	(5, 7),
(15, 'ТГУ', 368, 'Томск'),	(22, 'Физика', 34, 1),	(6, 3),
(18, 'ВГМА', 327, 'Вороне'),	(43, 'Математика', 56, 2),	(6, 7),
(22, 'МГУ', 606, 'Москва'),	(56, 'История', 34, 4),	(24, 22),
(32, 'РГУ', 416, 'Ростов'),	(73, 'Физкультура', 34, 5),	(46, 43),
(666, 'РТУ', 1000, 'Москва');	(94, 'Английский', 56, 3);	(74, 73),
		(108, 94),
		(276, 56),
		(328, 10);

INSERT INTO `exam_marks` (`EXAM_ID`, `STUDENT_ID`, `SUBJ_ID`, `MARK`, `EXAM_DATE`) VALUES (1, 1, 1, 5, '2021-01-13'),(2, 2, 9, 5, '2021-01-01'),(3, 4, 8, 4, '2021-01-09'),(4, 5, 7, 2, '2021-01-04'), (5, 6, 6, 3, '2021-01-12'),(6, 1, 5, 5, '2021-01-02'), (7, 1, 9, 5, '2021-01-01'),(10, 1, 1, 5, '2021-01-11'),(34, 32, 10, 4, '0000-00-00'), (43, 6, 22, 4, '0000-00-00'), (61, 32, 94, 5, '2021-01-09'), (62, 32, 1, 5, '2021-01-09'), (63, 32, 56, 5, '2021-01-09'), (75, 55, 10, 5, '0000-00-00'), (145, 12, 10, 5, '0000-00-00'), (238, 12, 22, 3, '0000-00-00'), (312, 32, 2, 2, '2021-03-17'), (313, 32, 2, 5, '2017-06-01'), (639, 55, 22, 5, '2021-02-28');

INSERT INTO `student` (`STUDENT_ID`, `SURNAME`, `NAME`, `STIPEND`, `KURS`, `CITY`, `BIRTHDAY`, `UNIV_ID`) **VALUES** (1, 'Сергеев', 'Сергей', '520.00', 1, 'Одинцово', '1983-03-21', 3), (2, 'Клюквина', 'Вера', '140.00', 3, 'Нижний Новгород', '1987-03-15', 6), (4, 'Водопьянова', 'Виктория', '170.00', 2, 'Костерево', '2000-01-09', 6), (5, 'Тимашов', 'Дмитрий', '90.00', 2, 'Горячий Ключ', '2000-01-09', 2), (6, 'Авраменко', 'Владислав', '480.00', 3, 'Павловское', '2000-01-09', 2), (10, 'Кузнецов', 'Борис', '0.00', 2, 'Брянск', '2021-10-12', 10), (12, 'Зайцева', 'Ольга', '250.00', 2, 'Липецк', '0000-00-00', 10), (30, 'Иванов', 'Иван', '150.00', 1, 'Орел', '0000-00-00', 10), (31, 'Петров', 'Петр', '200.00', 3, 'Курск', '2021-09-13', 10), (32, 'Котов', 'Павел', '150.00', 5, 'Белгород', NULL, 14), (55, 'Белкин', 'Вадим', '250.00', 5, 'Воронеж', '0000-00-00', 10), (60, 'Сидоров', 'Вадим', '150.00', 4, 'Москва', '2019-09-25', 22), (131, 'Васькин', 'В.', '132.00', 5, 'Брянск', '2021-03-01', 10), (265, 'Павлов', 'Андрей', '0.00', 3, 'Воронеж', '0000-00-00', 10), (276, 'Петров', 'Антон', '200.00', 4, NULL, '0000-00-00', 22), (654, 'Лукин', 'Артем', '200.00', 3, 'Воронеж', '0000-00-00', 10), (3231, NULL, NULL, '100.00', NULL, NULL, NULL, NULL), (32932, NULL, NULL, '300.00', NULL,

INSERT INTO `lecturer` (`LECTURER ID`, `SURNAME`, `NAME`, `CITY`, `UNIV ID`) **VALUES** (1, 'Бабкина', 'Надежда', 'Красный Холм', 1), (2, 'Бугрим', 'Наталья', 'Урюпинск', 3), (3, 'Каданцева', 'Наталья', 'Яхрома', 1), (4, 'Данилова', 'Елена', 'Мелединская', 5), (5, 'Красникова', 'Ольга', 'Семенов', 7), (6, 'Антонова', 'Наталья', NULL, 4), (7, 'Подкопаева', 'Елена', 'Пошехонье', 3), (8, 'Комова', 'Екатерина', 'Туапсе', 3), (9, 'Круглова', 'Екатерина', 'Бавлы', 3), (10, 'Кузнецова', 'Ольга', 'Никольск', 1), (11, 'Гашникова', 'Анастасия', 'Находка', 3), (24, 'Колесников', 'Борис', 'Воронеж', 10), (46, 'Никонов', 'Иван', 'Воронеж', 10), (74, 'Лагутин', 'Павел', 'Москва', 22), (108, 'Струков', 'Николай', 'Москва', 22), (276, 'Николаев', 'Виктор', 'Воронеж', 10), (328, 'Сорокин', 'Андрей', 'Орел', 10);

Теперь запросы. Сперва из День 1. Задание 1.

NULL, NULL, NULL);

1. Напишите запрос для вывода идентификатора (номера) предмета обучения, его наименования, семестра, в котором он читается, и количества отводимых на этот предмет часов для всех строк таблицы SUBJECT.

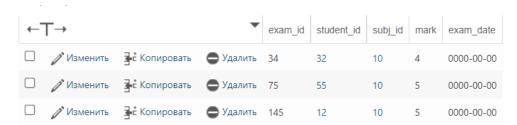
select * from subjects;

←_	Τ→		~	subj_id	subj_name	hours	semestr
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	1	Биология	119	1
	<i>№</i> Изменить	3 € Копировать	🖨 Удалить	2	Химия	65	4
	<i>№</i> Изменить	≩ € Копировать	🖨 Удалить	3	Сочинение	95	2
	<i>№</i> Изменить	≩ € Копировать	🖨 Удалить	4	География	74	2
	<i>№</i> Изменить	≩ € Копировать	🖨 Удалить	5	Собеседование	102	4
	<i>№</i> Изменить	≩ € Копировать	🖨 Удалить	6	Изложение	107	4
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	7	Физика	69	1
	<i>№</i> Изменить	≩ € Копировать	🖨 Удалить	8	История России	73	3
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	9	Литература	104	3
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	10	Информатика	56	1
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	22	Физика	34	1
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	43	Математика	56	2
	<i></i> ∕ Изменить	≩ € Копировать	🔵 Удалить	56	История	34	4

2. Напишите запрос, позволяющий вывести все строки таблицы EXAM_MARKS, в которых предмет обучения имеет номер (SUBJ_ID), равный 12.

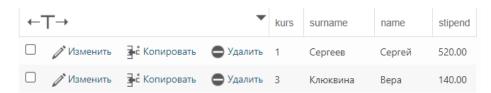
У меня нет такого id, использую 10

select * from exam_marks where subj_id = 10;



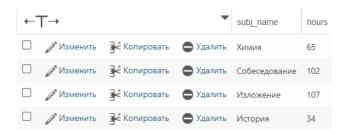
3. Напишите запрос, выбирающий все данные из таблицы STUDENT, расположив столбцы таблицы в следующем порядке: KURS, SURNAME, NAME, STIPEND

select kurs, surname, name, stipend from student;



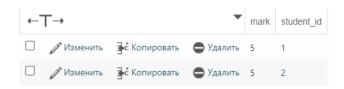
4. Напишите запрос SELECT, который выводит наименование предмета обучения (SUBJNAME) и количество часов (HOUR) для каждого предмета (SUBJECT) в 4-м семестре (SEMESTER).

select subj name, hours from subjects where semestr = 4;



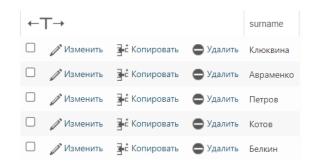
5. Напишите запрос, позволяющий получить из таблицы EXAM_MARKS значения столбца MARK (экзаменационная оценка) для всех студентов, исключив из списка повторение одинаковых строк.

select DISTINCT(mark), student id from exam marks;



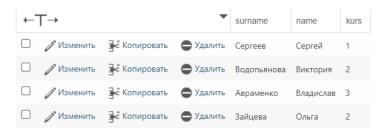
6. Напишите запрос, который выводит список фамилий студентов, обучающихся на третьем и последующих курсах.

select surname from student where kurs >2;



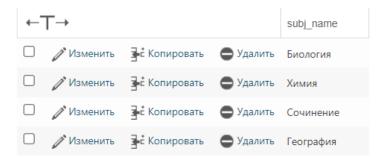
7. Напишите запрос, выбирающий данные о фамилии, имени и номере курса для студентов, получающих стипендию больше 140.

select surname, name, kurs from student where stipend > 140;



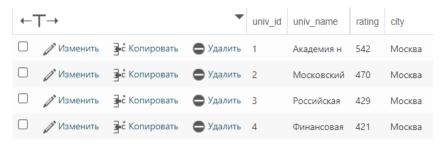
8. Напишите запрос, выполняющий выборку из таблицы SUBJECT названий всех предметов обучения, на которые отводится более 30 часов.

select subj_name from subjects where hours > 30;



9. Напишите запрос, который выполняет вывод списка университетов, рейтинг которых превышает 300 баллов.

select * from university where rating > 300;



10. Напишите запрос к таблице STUDENT для вывода списка фамилий (SURNAME), имен (NAME) и номера курса (KURS) всех студентов со стипендией, большей или равной 100, и живущих в Воронеже.

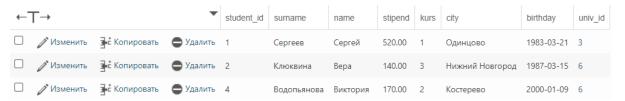
Со стипендией 100 никого нет, я поменял на 250

select surname, name, kurs from student where stipend >= 250 and city = 'Воронеж';

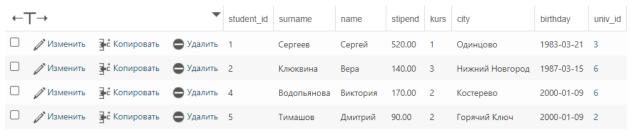


11. Какие данные будут получены в результате выполнения запроса? SELECT * FROM STUDENT WHERE (STIPEND < 100 OR NOT (BIRTHDAY >= '10/03/1980' AND STODENT_ID > 1003));

Запрос выдаст ошибку, так как неизвестное поле student_id, но если принять это за опечатку и исправить, то будет следующее:



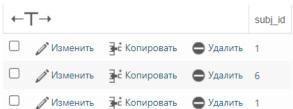
12. Какие данные будут получены в результате выполнения запроса? SELECT * FROM STUDENT WHERE NOT ((BIRTHDAY = '10/03/1980' OR STIPEND > 100) AND STUDENT_ID >= 1003);



13. Напишите запрос на вывод находящихся в таблице EXAM_MARKS номеров предметов обучения, экзамены по которым сдавались между 10 и 20 января 1999 года.

В 1999 году значений есть, я взял 2021

select subj_id from exam_marks where exam_date > '2021-0110' and exam_date < '2021-01-20';</pre>



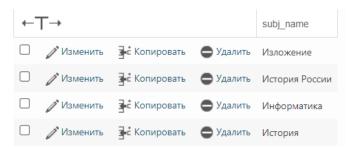
14. Напишите запрос, выбирающий данные обо всех предметах обучения, экзамены по которым сданы студентами, имеющими идентификаторы 12 и 32.

select * from subjects sub, exam_marks em where em.subj_id = sub.subj_
id and em.student_id in (12, 32);

subj_id	subj_name	hours	semestr	exam_id	student_id	subj_id	mark	exam_date
10	Информатика	56	1	34	32	10	4	0000-00-00
94	Английский	56	3	61	32	94	5	2021-01-09
1	Биология	119	1	62	32	1	5	2021-01-09

15. Напишите запрос на вывод названий предметов обучения, начинающихся на букву «И».

SELECT subj_name from subjects where subj_name like 'м%';



16. Напишите запрос, выбирающий сведения о студентах, у которых имена начинаются на буквы «И» или «С».

<u>select</u> * from student where name <u>like</u> 'μ%' <u>or</u> name <u>like</u> 'c%';

←_	Γ→		•	student_id	surname	name	stipend	kurs	city	birthday	univ_id
	<i>/</i> Изменить	💤 Копировать	🖨 Удалить	1	Сергеев	Сергей	520.00	1	Одинцово	1983-03-21	3
	Изменить	💤 Копировать	🖨 Удалить	30	Иванов	Иван	150.00	1	Орел	0000-00-00	10

17. Напишите запрос для выбора из таблицы EXAM_MARKS записей в которых отсутствуют значения оценок (поле MARK).

select * from exam_marks where mark is null;



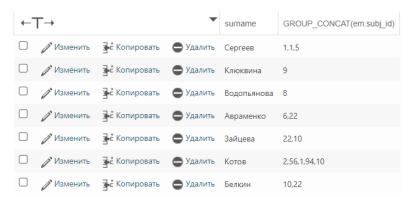
18. Напишите запрос на вывод из таблицы EXAM_MARKS записей, имеющих в поле MARK значения оценок.

select * from exam_marks where mark is not null;

	Τ→		_					exam_date
_	1 →			exam_id	student_id	Subj_id	mark	exam_date
	<i>№</i> Изменить	≩ с Копировать	🖨 Удалить	1	1	1	5	2021-01-13
	<i>№</i> Изменить	3 с Копировать	🖨 Удалить	2	2	9	5	2021-01-01

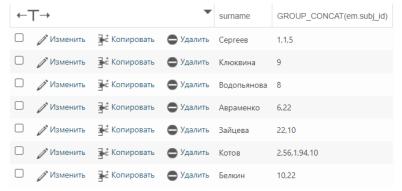
День 1. Задание 2

1. Напишите запрос, который выполняет вывод данных о фамилиях сдававших экзамены студентов (вместе с идентификаторами каждого сданного ими предмета обучения).



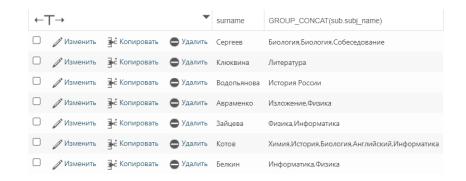
select st.surname, GROUP CONCAT(em.subj_id) from student st, exam_mark
s em where em.student_id = st.student_id and mark > 2 GROUP BY st.stud
ent_id;

2. Напишите запрос, который выполняет выборку значений фамилии всех студентов с указанием для студентов, сдававших экзамены, идентификаторов сданных ими предметов обучения.



select st.surname, GROUP CONCAT(em.subj_id) from student st, exam_marks em wh
ere em.student_id = st.student_id and mark > 2 GROUP BY st.student_id;

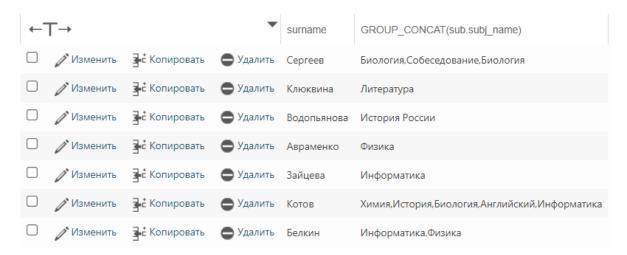
3. Напишите запрос, который выполняет вывод данных о фамилиях студентов, сдававших экзамены, вместе с наименованиями каждого сданного ими предмета обучения.



select st.surname, GROUP CONCAT(sub.subj_name) from student st, exam_m
arks em, subjects sub where em.student_id = st.student_id and sub.subj
_id = em.subj_id and mark > 2 GROUP BY st.student_id;

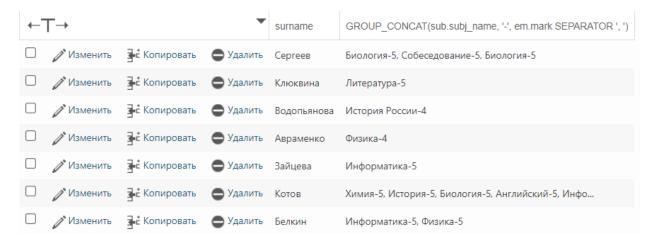
4. Напишите запрос на выдачу для каждого студента названий всех предметов обучения, по которым этот студент получил оценку 4 или 5.

select st.surname, GROUP CONCAT(sub.subj_name) from student st, exam_m
arks em, subjects sub where em.student_id = st.student_id and sub.subj
_id = em.subj_id and mark in (4,5) GROUP BY st.student_id;



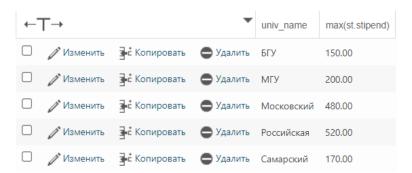
5. Напишите запрос на выдачу данных о названиях всех предметов, по которым студенты получили только хорошие (4 и 5) оценки. В выходных данных должны быть приведены фамилии студентов, названия предметов и оценка.

select st.surname, GROUP CONCAT(sub.subj_name, '', em.mark SEPARATOR ',
') from student st, exam_marks em, subjects sub where em.student_id =
st.student_id and sub.subj_id = em.subj_id and mark in (4,5) GROUP BY
st.student id;



6. Напишите запрос, который выполняет вывод списка университетов с рейтингом, превышающим 300, вместе со значением максимального размера стипендии, получаемой студентами в этих университетах.

select u.univ_name, max(st.stipend) from university u, student st wher
e rating > 300 and st.univ_id = u.univ_id GROUP BY u.univ_name;



7. Напишите запрос на выдачу списка фамилий студентов (в алфавитном порядке) вместе со значением рейтинга университета, где каждый из них учится, включив в список и тех студентов, для которых в базе данных не указано место их учебы.

name 🔺 1	univ_name	rating
	NULL	NULL
Андрей	ВГУ	296
Антон	МГУ	606
Артем	ВГУ	296
Борис	ВГУ	296

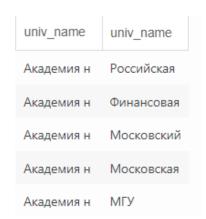
select st.name, u.univ_name, u.rating from student st <u>left</u> join univer sity u on st.univ id = u.univ id ORDER by st.name;

8. Написать запрос, выполняющий вывод списка всех пар фамилий студентов, проживающих в одном городе. При этом не включать в список комбинации фамилий студентов самих с собой (то есть комбинацию типа «Иванов-Иванов») и комбинации фамилий студентов, отличающиеся порядком следования (то есть включать одну из двух комбинаций типа «Иванов-Петров» и «Петров-Иванов»).

surname	surname
Кузнецов	Васькин
Белкин	Павлов
Белкин	Лукин
Павлов	Лукин

select st1.surname, st2.surname from student st1, student st2 where (s
t1.city = st2.city) and (st1.student_id < st2.student_id);</pre>

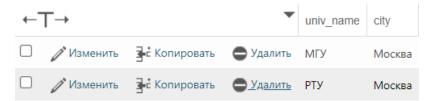
9. Написать запрос, выполняющий вывод списка всех пар названий университетов, расположенных в одном городе, не включая в список комбинации названий университетов самих с собой и пары названий университетов, отличающиеся порядком следования.



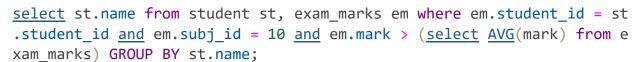
```
select u1.univ_name, u2.univ_name from university u1, university u2 wh
ere (u1.city = u2.city) and (u1.univ_id < u2.univ_id);</pre>
```

10. Написать запрос, который позволяет получить данные о названиях университетов и городов, в которых они расположены, с рейтингом, равным или превышающим рейтинг ВГУ.

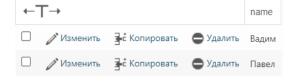
select univ_name, city from university where rating >= (select rating from university where univ_name = 'ΒΓΥ');



- 11. Напишите запрос, выбирающий данные об именах всех студентов, имеющих по предмету с идентификатором 101 балл выше общего среднего балла.
- 101 идентификатора нет, я поменял на 10



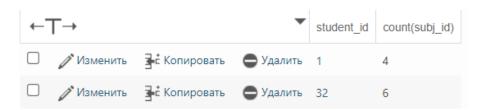
12. Напишите запрос, который выполняет выборку имен всех студентов, имеющих по предмету с идентификатором 102 балл ниже общего среднего балла.



select st.name from student st, exam_marks em where em.student_id = st
.student_id and em.subj_id = 10 and em.mark < (select AVG(mark) from e
xam_marks) GROUP BY st.name;</pre>

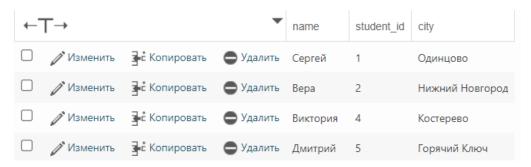
13. Напишите запрос, выполняющий вывод количества предметов, по которым экзаменовался каждый студент, сдававший более 20 предметов.

select student_id, count(subj_id) from exam_marks GROUP BY student_id
HAVING COUNT(subj_id)>2;



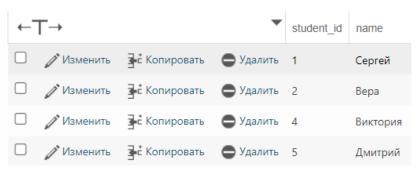
14. Напишите команду SELECT, использующую связанные подзапросы и выполняющую вывод имен и идентификаторов студентов, у которых стипендия совпадает с максимальным значением стипендии для города, в котором живет студент.

select st.name, st.student_id, st.city from student st where stipend =
 (SELECT max(stipend) from student st1 where st1.city = st.city);



15. Напишите запрос, который позволяет вывести имена и идентификаторы всех студентов, для которых точно известно, что они проживают в городе, где нет ни одного университета.

 $\underline{\text{select}}$ st.student_id, st.name from student st where city $\underline{\text{not}}$ in ($\underline{\text{selec}}$ $\underline{\text{t}}$ city from university);



16. Напишите два запроса, которые позволяют вывести имена и идентификаторы всех студентов, для которых точно известно, что он проживают не в том городе, где расположен их университет. Один запрос с использованием соединения, а другой — с использованием связанного подзапроса.

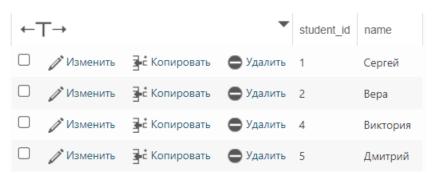
Соединение

select st.student_id, st.name, u.city, st.city from student st inner j
oin university u on u.univ_id = st.univ_id where u.city <> st.city;

student_id	name	city	city
1	Сергей	Москва	Одинцово
2	Вера	Самара	Нижний Новгород
4	Виктория	Самара	Костерево

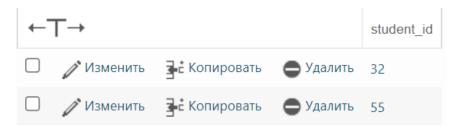
Подзапрос

select st.student_id, st.name from student st where city <> (select u.
city from university u where u.univ_id = st.univ_id);



17. Извлечь из таблицы EXAM_MARK данные о студентах, получивших хотя бы одну неудовлетворительную оценку, среди тех студентов у которых средний балл по остальным предметам больше 4.

select em.student_id from exam_marks em where (select COUNT(mark) from
exam_marks em1 where em1.student_id = em.student_id and em1.mark = 2)
>= 1 and (select AVG(mark) from exam_marks em2 where em.subj_id <> em
2.subj_id and em2.student_id = em.student_id) > 4 GROUP BY em.student_
id;



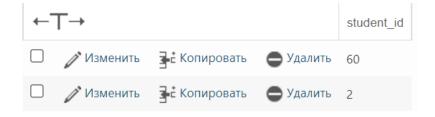
18. Напишите команду SELECT, использующую связанные подзапросы и выполняющую вывод имен и идентификаторов студентов, у которых стипендия совпадает с максимальным значением стипендии для города, в котором живет студент.

select st.student_id, st.name from student st where st.stipend = (sele ct max(st1.stipend) from student st1 where st.city = st1.city);

←	Τ→		•	student_id	name
	/ Изменить	3- Е Копировать	🕳 Удалить	1	Сергей
	Изменить	3- с Копировать	🖨 Удалить	2	Вера
	/ Изменить	3 - с Копировать	🖨 Удалить	4	Виктория

19. Напишите запрос с EXISTS, выбирающий сведения обо всех студентах, для которых в том же городе, где живет студент, существуют университеты, в которых он не учит

select st.student_id from student st where EXISTS(select u.univ_id fro
m university u where st.univ_id <> u.univ_id and u.city = st.city);



20. Напишите запрос, выбирающий из таблицы EXAM_MARKS данные о названиях предметов обучения, для которых значение полученных на экзамене оценок (поле MARK) превышает любое значение оценки для предмета, имеющего идентификатор, равный 105.

<u>select</u> sub.subj_id, sub.subj_name, em.mark from subjects sub inner joi n exam_marks em on sub.subj_id = em.subj_id where em.mark > (\underline{select} \underline{ma} \underline{x} (mark) from exam_marks where subj_id = 105) GROUP BY sub.subj_id, sub

.subj name, em.mark;

subj_id	subj_name	mark
1	Биология	5
2	Химия	5
5	Собеседование	5