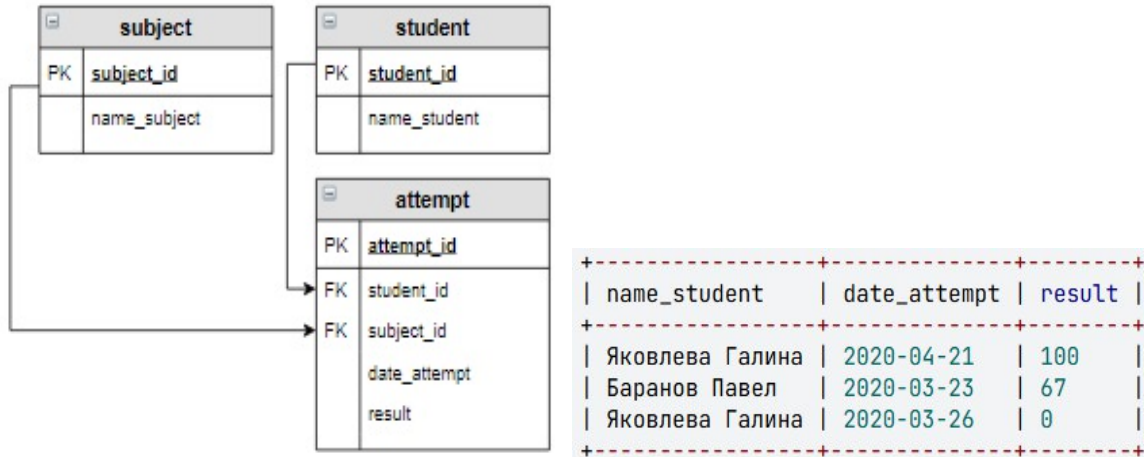


## Задание

## Запрос

Вывести студентов, которые сдавали дисциплину «Основы баз данных», указать дату попытки и результат. Информацию вывести по убыванию результатов тестирования.



```

SELECT name_student, date_attempt, result
FROM
    student
    INNER JOIN attempt USING(student_id)
    INNER JOIN subject USING(subject_id)
WHERE name_subject='Основы баз данных'
ORDER BY result DESC
  
```

Вывести, сколько попыток сделали студенты по каждой дисциплине, а также средний результат попыток, который округлить до 2 знаков после запятой. Под результатом попытки понимается процент правильных ответов на вопросы теста, который занесен в столбец **result**. В результат включить название дисциплины, а также вычисляемые столбцы **Количество** и **Среднее**. Информацию вывести по убыванию средних результатов.

```

SELECT name_subject,
COUNT(attempt.attempt_id) AS Количество,
ROUND(AVG(result), 2) AS Среднее
FROM subject LEFT JOIN attempt USING (subject_id)
GROUP BY name_subject
ORDER BY Среднее DESC
  
```

name_subject	Количество	Среднее
Основы SQL	4	58.25
Основы баз данных	3	55.67
Физика	0	NULL

Вывести студентов (различных студентов), имеющих максимальные результаты попыток. Информацию отсортировать в алфавитном порядке по фамилии студента. Максимальный результат не обязательно будет 100%.

name_student	result
Семенов Иван	100
Яковлева Галина	100

```
SELECT name_student, result
FROM student JOIN attempt USING(student_id)
WHERE result=(SELECT MAX(result) FROM attempt)
ORDER BY name_student
```

Если студент совершал **несколько попыток** по одной и той же дисциплине, то вывести разницу в днях между первой и последней попыткой. В результат включить фамилию и имя студента, название дисциплины и вычисляемый столбец **Интервал**. Информацию вывести по возрастанию разницы. Студентов, сделавших одну попытку по дисциплине, не учитывать.

name_student	name_subject	Интервал
Яковлева Галина	Основы баз данных	26
Семенов Иван	Основы SQL	55

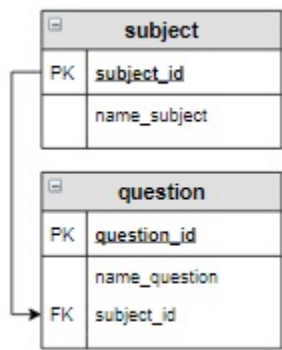
```
SELECT name_student, name_subject,
DATEDIFF(MAX(date_attempt), MIN(date_attempt)) AS Интервал
FROM student
INNER JOIN attempt USING(student_id)
INNER JOIN subject USING(subject_id)
GROUP BY name_student, name_subject
HAVING COUNT(*) > 1
ORDER BY Интервал
```

Студенты могут тестироваться по одной или нескольким дисциплинам (не обязательно по всем). Вывести дисциплину и количество уникальных студентов (столбец назвать **Количество**), которые по ней проходили тестирование. Информацию отсортировать сначала по убыванию количества, а потом по названию дисциплины. В результат включить и дисциплины, тестирование по которым студенты еще не проходили, в этом случае указать количество студентов 0.

name_subject	Количество
Основы SQL	2
Основы баз данных	2
Физика	0

```
SELECT name_subject, COUNT(DISTINCT student_id) AS Количество
FROM subject
LEFT JOIN attempt USING(subject_id)
GROUP BY name_subject
ORDER BY Количество DESC, name_subject
```

Случайным образом отберите 3 вопроса по дисциплине «Основы баз данных». В результат включите столбцы **question\_id** и **name\_question**.



question_id	name_question
8	Концептуальная модель используется для
7	Отношение - это:
6	База данных - это:

```
SELECT question_id, name_question
FROM question JOIN subject USING(subject_id)
WHERE name_subject='Основы баз данных'
ORDER BY RAND()
LIMIT 3
```

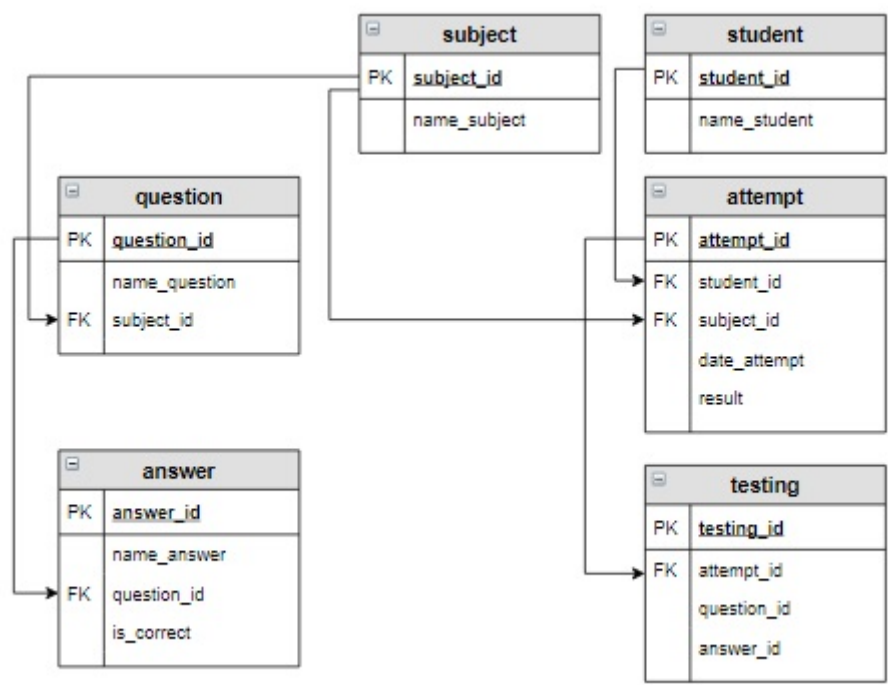
Вывести вопросы, которые были включены в тест для Семенова Ивана по дисциплине «Основы SQL» 2020-05-17 (значение **attempt\_id** для этой попытки равно 7). Указать, какой ответ дал студент и правильный он или нет (вывести **Верно** или **Неверно**). В результат включить вопрос, ответ и вычисляемый столбец **Результат**.

name_question	name_answer	Результат
Запрос на выборку начинается с ключевого слова:	INSERT	Неверно
Какой запрос выбирает все записи из таблицы student:	SELECT * FROM student	Верно
Для внутреннего соединения таблиц используется оператор:	CROSS JOIN	Неверно

```
SELECT name_question, name_answer,
IF(is_correct=1, 'Верно', 'Неверно') AS Результат
FROM testing
    INNER JOIN answer USING(answer_id)
    INNER JOIN question ON answer.question_id=question.question_id
WHERE attempt_id=7
```



Посчитать результаты тестирования. Результат попытки вычислить как количество правильных ответов, деленное на 3 (количество вопросов в каждой попытке) и умноженное на 100. Результат округлить до двух знаков после запятой. Вывести фамилию студента, название предмета, дату и результат. Последний столбец назвать **Результат**. Информацию отсортировать сначала по фамилии студента, потом по убыванию даты попытки.



name_student	name_subject	date_attempt	Результат
Баранов Павел	Основы SQL	2020-04-15	33.33
Баранов Павел	Основы баз данных	2020-03-23	66.67
Семенов Иван	Основы SQL	2020-05-17	33.33
Семенов Иван	Основы SQL	2020-04-15	66.67
Семенов Иван	Основы SQL	2020-03-23	100.00
Яковлева Галина	Основы баз данных	2020-04-21	100.00
Яковлева Галина	Основы баз данных	2020-03-26	0.00

```

SELECT name_student, name_subject,
date_attempt, ROUND((SUM(is_correct)/3)*100, 2) AS Результат
FROM answer
JOIN testing USING(answer_id)
JOIN attempt USING(attempt_id)
JOIN subject USING(subject_id)
JOIN student USING(student_id)
GROUP BY name_student, name_subject, date_attempt
ORDER BY name_student, date_attempt DESC
    
```

Для каждого вопроса вывести процент успешных решений, то есть отношение количества верных ответов к общему количеству ответов, значение округлить до 2-х знаков после запятой. Также вывести название предмета, к которому относится вопрос, и общее количество ответов на этот вопрос. В результат включить название дисциплины, вопросы по ней (столбец назвать **Вопрос**), а также два вычисляемых столбца **Всего\_ответов** и **Успешность**. Информацию отсортировать сначала по названию дисциплины, потом по убыванию успешности, а потом по тексту вопроса в алфавитном порядке.

Поскольку тексты вопросов могут быть длинными, обрезать их 30 символов и добавить многоточие "...".

name_subject	Вопрос	Всего_ответов	Успешность
Основы SQL	Условие, по которому отбираютс...	1	100.00
Основы SQL	Запрос на выборку начинается с...	4	75.00
Основы SQL	Какой запрос выбирает все запи...	3	66.67
Основы SQL	Для сортировки используется:...	2	50.00
Основы SQL	Для внутреннего соединения таб...	2	0.00
Основы баз данных	База данных - это:...	3	66.67
Основы баз данных	Какой тип данных не допустим в...	2	50.00
Основы баз данных	Концептуальная модель использу...	2	50.00
Основы баз данных	Отношение - это:...	2	50.00

Придумайте один или несколько запросов на выборку для предметной области «Тестирование» (в таблицы занесены данные, как на [первом шаге](#) урока).

```
SELECT name_subject, CONCAT(LEFT(name_question, 30), '...') AS Вопрос,
COUNT(testing.answer_id) AS Всего_ответов,
ROUND(SUM(is_correct)/COUNT(testing.answer_id) * 100, 2) AS Успешность
FROM
    subject
JOIN question USING(subject_id)
JOIN testing USING(question_id )
LEFT JOIN answer USING(answer_id)
GROUP BY name_subject, name_question
ORDER BY name_subject, Успешность DESC, Вопрос
```

```
SELECT name_student, name_subject, date_attempt,
SUM(is_correct) AS 'Правильные ответы'
FROM
    student
JOIN attempt ON student.student_id = attempt.student_id
JOIN subject ON attempt.subject_id = subject.subject_id
JOIN testing ON attempt.attempt_id = testing.attempt_id
JOIN answer ON testing.answer_id = answer.answer_id
JOIN question ON testing.question_id = question.question_id
WHERE name_subject='Основы SQL' AND date_attempt='2020-04-15'
GROUP BY name_student
ORDER BY name_student
```

## Логическая схема базы данных:

