**Предметная область**

В университете реализуется on-line тестирование по нескольким дисциплинам. Каждая дисциплина включает некоторое количество вопросов. Ответы на вопрос представлены в виде вариантов ответов, один из этих вариантов правильный.

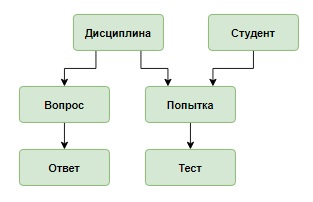
Студент регистрируется в системе, указав свое имя, фамилию и отчество. После этого он может проходить тестирование по одной или нескольким дисциплинам. Студент имеет несколько попыток для прохождения тестирования  (необходимо сохранять дату попытки). Каждому студенту случайным образом выбирается набор вопросов по дисциплине и формируется индивидуальный тест. Студент отвечает на вопросы, выбирая один из предложенных вариантов ответа.

После окончания тестирования  вычисляется и сохраняется результат (в процентах) попытки.

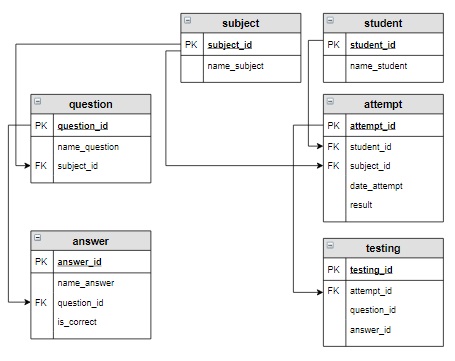
**Запросы для предметной области:**

1. [Вывести студентов, которые сдавали определенную дисциплину](https://stepik.org/lesson/310421/step/2?unit=292727).
2. [Вывести количество попыток и средний результат по каждой дисциплине](https://stepik.org/lesson/310421/step/3?unit=292727).
3. [Вывести студента (студентов), которые набрали максимальный результат при тестировании](https://stepik.org/lesson/310421/step/4?unit=292727).
4. [Если студент совершал несколько попыток по одной и той же дисциплине, то вывести разницу в днях между первой и последней попыткой](https://stepik.org/lesson/310421/step/5?unit=292727).
5. [Вывести количество студентов, которые проходили тестирование по каждой дисциплине](https://stepik.org/lesson/310421/step/6?unit=292727).
6. [Случайным образом отобрать 3 вопроса по определенной дисциплине.](https://stepik.org/lesson/310421/step/7?unit=292727)
7. [Вывести вопросы, на которые отвечал определенный студент в определенной попытке, ответ студента и результат (правильно или нет)](https://stepik.org/lesson/310421/step/8?unit=292727).
8. [Посчитать результаты тестирования для всех попыток](https://stepik.org/lesson/310421/step/9?unit=292727).
9. [Для каждого вопроса вывести процент успешных решений и общее количество ответов](https://stepik.org/lesson/310421/step/10?unit=292727).

**Концептуальная схема базы данных**



**Логическая схема базы данных**



**Структура и наполнение таблиц**

Таблица**subject:**

|  |  |
| --- | --- |
| **subject\_id** | **name\_subject** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(30)** |
| 1 | Основы SQL |
| 2 | Основы баз данных |
| 3 | Физика |

Таблица **student** :

|  |  |
| --- | --- |
| **student\_id** | **name\_student** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(50)** |
| 1 | Баранов Павел |
| 2 | Абрамова Катя |
| 3 | Семенов Иван |
| 4 | Яковлева Галина |

Таблица **attempt**(в таблице хранится информация о каждой попытке сдачи теста : **id** студента, **id** дисциплины,  дата попытки и результат в процентах):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **attempt\_id** | **student\_id** | **subject\_id** | **date\_attempt** | **result** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **INT** | **INT** | **DATE** | **INT** |
| 1 | 1 | 2 | 2020-03-23 | 67 |
| 2 | 3 | 1 | 2020-03-23 | 100 |
| 3 | 4 | 2 | 2020-03-26 | 0 |
| 4 | 1 | 1 | 2020-04-15 | 33 |
| 5 | 3 | 1 | 2020-04-15 | 67 |
| 6 | 4 | 2 | 2020-04-21 | 100 |
| 7 | 3 | 1 | 2020-05-17 | 33 |

Таблица **question**(таблица содержит вопросы по каждой дисциплине):

|  |  |  |
| --- | --- | --- |
| **question\_id** | **name\_question** | **subject\_id** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(100)** | **INT** |
| 1 | Запрос на выборку начинается с ключевого слова: | 1 |
| 2 | Условие, по которому отбираются записи, задается после ключевого слова: | 1 |
| 3 | Для сортировки используется: | 1 |
| 4 | Какой запрос выбирает все записи из таблицы student: | 1 |
| 5 | Для внутреннего соединения таблиц используется оператор: | 1 |
| 6 | База данных - это: | 2 |
| 7 | Отношение - это: | 2 |
| 8 | Концептуальная модель используется для | 2 |
| 9 | Какой тип данных не допустим в реляционной таблице? | 2 |

Таблица **answer**(в таблице содержатся варианты ответов на каждый вопрос, для правильного варианта столбец **is\_correct** имеет значение**true**):

|  |  |  |  |
| --- | --- | --- | --- |
| **answer\_id** | **name\_answer** | **question\_id** | **is\_correct** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(100)** | **INT** | **BOOL** |
| 1 | UPDATE | 1 | false |
| 2 | SELECT | 1 | true |
| 3 | INSERT | 1 | false |
| 4 | GROUP BY | 2 | false |
| 5 | FROM | 2 | false |
| 6 | WHERE | 2 | true |
| 7 | SELECT | 2 | false |
| 8 | SORT | 3 | false |
| 9 | ORDER BY | 3 | true |
| 10 | RANG BY | 3 | false |
| 11 | SELECT \* FROM student | 4 | true |
| 12 | SELECT student | 4 | false |
| 13 | INNER JOIN | 5 | true |
| 14 | LEFT JOIN | 5 | false |
| 15 | RIGHT JOIN | 5 | false |
| 16 | CROSS JOIN | 5 | false |
| 17 | совокупность данных, организованных по определенным правилам | 6 | true |
| 18 | совокупность программ для хранения и обработки больших массивов информации | 6 | false |
| 19 | строка | 7 | false |
| 20 | столбец | 7 | false |
| 21 | таблица | 7 | true |
| 22 | обобщенное представление пользователей о данных | 8 | true |
| 23 | описание представления данных в памяти компьютера | 8 | false |
| 24 | база данных | 8 | false |
| 25 | file | 9 | true |
| 26 | INT | 9 | false |
| 27 | VARCHAR | 9 | false |
| 28 | DATE | 9 | false |

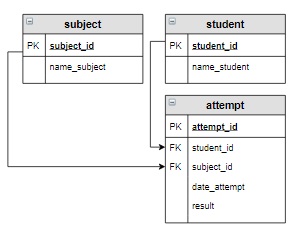
Таблица **testing** (в таблице хранится информация о каждой попытке студента: какие вопросы были заданы и какой ответ дал студент)

|  |  |  |  |
| --- | --- | --- | --- |
| **testing\_id** | **attempt\_id** | **question\_id** | **answer\_id** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **INT** | **INT** | **INT** |
| 1 | 1 | 9 | 25 |
| 2 | 1 | 7 | 19 |
| 3 | 1 | 6 | 17 |
| 4 | 2 | 3 | 9 |
| 5 | 2 | 1 | 2 |
| 6 | 2 | 4 | 11 |
| 7 | 3 | 6 | 18 |
| 8 | 3 | 8 | 24 |
| 9 | 3 | 9 | 28 |
| 10 | 4 | 1 | 2 |
| 11 | 4 | 5 | 16 |
| 12 | 4 | 3 | 10 |
| 13 | 5 | 2 | 6 |
| 14 | 5 | 1 | 2 |
| 15 | 5 | 4 | 12 |
| 16 | 6 | 6 | 17 |
| 17 | 6 | 8 | 22 |
| 18 | 6 | 7 | 21 |
| 19 | 7 | 1 | 3 |
| 20 | 7 | 4 | 11 |
| 21 | 7 | 5 | 16 |

**Задание**

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

**Фрагмент логической схемы базы данных**

​

**Связанные шаги**

**Текст задания** (чтобы не прокручивать страницу)

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

**Результат**

**Наполнение таблиц attempt, student, subject**

1

SELECT student.name\_student, attempt.date\_attempt, attempt.result

2

FROM

3

  subject

4

   JOIN attempt USING(subject\_id)

5

   JOIN student ON student.student\_id = attempt.student\_id

6

                   AND subject\_id = 2

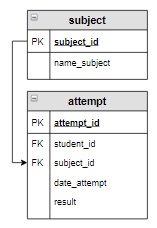
7

ORDER BY 3 DESC;

**Задание**

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.

**Фрагмент логической схемы базы данных**



**Пояснение**

**Связанные шаги**

**Текст задания**(чтобы не прокручивать страницу)

*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.*

**Результат**

**Структура и наполнение таблиц attempt и subject**

таблица attempt

+------------+------------+------------+--------------+--------+

| attempt\_id | student\_id | subject\_id | date\_attempt | result |

+------------+------------+------------+--------------+--------+

| 1 | 1 | 2 | 2020-03-23 | 67 |

| 2 | 3 | 1 | 2020-03-23 | 100 |

| 3 | 4 | 2 | 2020-03-26 | 0 |

| 4 | 1 | 1 | 2020-04-15 | 33 |

| 5 | 3 | 1 | 2020-04-15 | 67 |

| 6 | 4 | 2 | 2020-04-21 | 100 |

| 7 | 3 | 1 | 2020-05-17 | 33 |

+------------+------------+------------+--------------+--------+

таблица subject

+------------+-------------------+

| subject\_id | name\_subject |

+------------+-------------------+

| 1 | Основы SQL |

| 2 | Основы баз данных |

| 3 | Физика |

+------------+-------------------+

1

SELECT subject.name\_subject,

2

       COUNT(attempt.student\_id) AS 'Количество',

3

        ROUND(AVG(attempt.result), 2) AS 'Среднее'

4

FROM

5

  subject

6

    LEFT JOIN attempt ON subject.subject\_id = attempt.subject\_id

7

GROUP BY subject.name\_subject

8

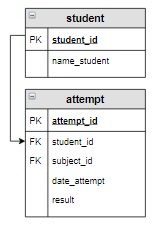
ORDER BY 3 DESC;

**Задание**

Вывести студентов (различных студентов), имеющих максимальные результаты попыток. Информацию отсортировать в алфавитном порядке по фамилии студента.

Максимальный результат не обязательно будет 100%, поэтому явно это значение в запросе не задавать.

**Фрагмент логической схемы базы данных:**



**Пояснение**

**Связанные шаги**

**Текст задания**(чтобы не прокручивать страницу)

*Вывести студентов (различных студентов), имеющих максимальные результаты попыток . Информацию отсортировать в алфавитном порядке по фамилии студента.*

**Результат**

**Структура и наполнение таблиц attempt и student**

1

SELECT student.name\_student, attempt.result

2

   FROM

3

  student

4

   JOIN attempt ON student.student\_id = attempt.student\_id

5

WHERE attempt.result = (SELECT MAX(result) FROM attempt)

6

ORDER BY 1;

7

​

8

SELECT\*FROM attempt;

9

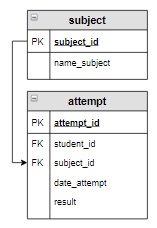
SELECT \* FROM student;

**Задание**

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.

**Пояснение**

**Фрагмент логической схемы базы данных**



**Пояснение**

**Связанные шаги**

**Текст задания**(чтобы не прокручивать страницу)

*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 не проходили, в этом случае указать количество студентов 0.*

**Результат**

**Структура и наполнение таблиц attempt и subject**

1

/\*SELECT name\_subject,

2

      COUNT(DISTINCT(attempt.student\_id)) AS 'Количество'

3

FROM

4

  attempt

5

  RIGHT JOIN subject ON subject.subject\_id = attempt.subject\_id

6

GROUP BY 1

7

ORDER BY 2 DESC, 1;\*/

8

​

9

SELECT name\_subject,

10

   COUNT(query\_in.student\_id) AS Количество

11

FROM (SELECT DISTINCT subject\_id, student\_id FROM attempt)query\_in

12

            right JOIN subject

13

            ON query\_in.subject\_id = subject.subject\_id

14

GROUP BY 1

15

ORDER BY 2 DESC, 1;

16

​

17

SELECT \* FROM attempt;

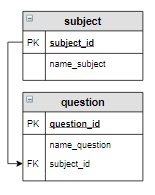
18

SELECT \* FROM subject;

**Задание**

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

**Фрагмент логической схемы базы данных**



**Пояснение**

**Связанные шаги**

**Текст задания**(чтобы не прокручивать страницу)

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

**Результат**

**Структура и наполнение таблиц question и subject**

таблица question

+-------------+-------------------------------------------------------------------------+------------+

| question\_id | name\_question | subject\_id |

+-------------+-------------------------------------------------------------------------+------------+

| 1 | Запрос на выборку начинается с ключевого слова: | 1 |

| 2 | Условие, по которому отбираются записи, задается после ключевого слова: | 1 |

| 3 | Для сортировки используется: | 1 |

| 4 | Какой запрос выбирает все записи из таблицы student: | 1 |

| 5 | Для внутреннего соединения таблиц используется оператор: | 1 |

| 6 | База данных - это: | 2 |

| 7 | Отношение - это: | 2 |

| 8 | Концептуальная модель используется для | 2 |

| 9 | Какой тип данных не допустим в реляционной таблице? | 2 |

+-------------+-------------------------------------------------------------------------+------------+

таблица subject

+------------+-------------------+

| subject\_id | name\_subject |

+------------+-------------------+

| 1 | Основы SQL |

| 2 | Основы баз данных |

| 3 | Физика |

+------------+-------------------+

1

SELECT question\_id, name\_question

FROM

question, subject

/\*JOIN subject ON subject.subject\_id = question.subject\_id\*/

/\*2 WHERE question.subject\_id = 2\*/

/\*3 WHERE subject\_id = (SELECT subject\_id

FROM subject

WHERE name\_subject = 'Основы баз данных')\*/

WHERE question.subject\_id = subject.subject\_id AND name\_subject LIKE 'Основы баз данных'

ORDER BY RAND()

LIMIT 3;

SELECT \* FROM question;

SELECT \* FROM subject;

====================================================================================