**Содержание урока**

В данном уроке с помощью запросов корректировки для базы данных «Тестирование» будет реализован процесс подготовки вопросов для тестирования и обработки результатов попытки студента:

* [добавление новой попытки для определенного студента и определенной дисциплины](https://stepik.org/lesson/310422/step/2?unit=292728);
* [формирование тестовых вопросов для этой попытки;](https://stepik.org/lesson/310422/step/3?unit=292728)
* [вычисление результата тестирования](https://stepik.org/lesson/310422/step/4?unit=292728);
* [удаление неактуальных попыток](https://stepik.org/lesson/310422/step/5?unit=292728).

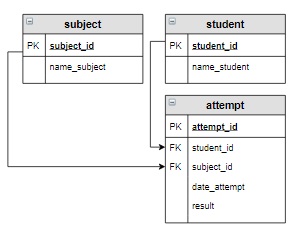
Структура таблиц и их наполнение как на [первом шаге](https://stepik.org/lesson/310421/step/1?unit=292727) предыдущего урока.

**=============================================**

**Задание**

В таблицу **attempt**включить новую попытку для студента Баранова Павла по дисциплине «Основы баз данных». Установить текущую дату в качестве даты выполнения попытки.

**Фрагмент логической схемы базы данных**



**Пояснение**

**Связанные шаги**

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

*В таблицу****attempt****включить новую попытку для студента Баранова Павла по дисциплине «Основы баз данных». Установить текущую дату в качестве даты выполнения попытки.*

**Результат**

**Наполнение таблиц subject, student, attempt (перед выполнением шага)**

таблица student таблица subject

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

| student\_id | name\_student | | subject\_id | name\_subject |

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

| 1 | Баранов Павел | | 1 | Основы SQL |

| 2 | Абрамова Катя | | 2 | Основы баз данных |

| 3 | Семенов Иван | | 3 | Физика |

| 4 | Яковлева Галина | +------------+-------------------+

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

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 |

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

1

/\*1

2

INSERT INTO attempt(student\_id, subject\_id, date\_attempt, result)

3

SELECT student.student\_id, subject.subject\_id, Now(), Null

4

FROM student

5

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

6

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

7

WHERE student.student\_id = 1 AND subject.subject\_id = 2;

8

SELECT\*FROM attempt;\*/

9

​

10

/\*2

11

INSERT INTO attempt(student\_id, subject\_id, date\_attempt, result)

12

SELECT student.student\_id, subject.subject\_id, Now(), Null

13

FROM attempt

14

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

15

  AND subject.name\_subject LIKE 'Основы баз данных'

16

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

17

  AND student.name\_student LIKE 'Баранов Павел';\*/

18

19

/\*3

20

\*/

21

INSERT INTO attempt(student\_id, subject\_id, date\_attempt, result)

22

VALUES((SELECT student\_id FROM student WHERE name\_student LIKE 'Баранов Павел'),

23

      (SELECT subject\_id FROM subject WHERE name\_subject LIKE 'Основы баз данных'),

24

      Now(),

25

      Null);

26

​

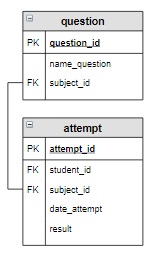
27

SELECT\*FROM attempt;

**Задание**

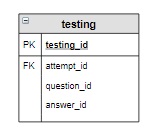
Случайным образом выбрать три вопроса ([запрос](https://stepik.org/lesson/310421/step/7?unit=292727)) по дисциплине, тестирование по которой собирается проходить студент, занесенный в таблицу **attempt** последним, и добавить их в таблицу**testing**.**id** последней попытки получить как максимальное значение**id** из таблицы **attempt**.

**Фрагмент логической схемы базы данных**



**Пояснение**

**Корректируемая таблица:**



**Связанные шаги**

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

*Случайным образом выбрать три вопроса ([запрос](https://stepik.org/lesson/310421/step/7?unit=292727)) по дисциплине, тестирование по которой собирается проходить студент, занесенный в таблицу****attempt****последним, и добавить их в таблицу****testing.id****последней попытки получить как максимальное значение****id****из таблицы****attempt****.*

**Результат**

**Наполнение таблиц question, attempt, testing(перед выполнением шага)**

1

/\*INSERT INTO testing(attempt\_id, question\_id, answer\_id)

2

VALUES ((SELECT MAX(attempt\_id) FROM attempt),

3

      (SELECT question\_id FROM question, subject

4

        WHERE subject.subject\_id = question.subject\_id

5

        ORDER BY RAND()

6

        LIMIT 1), NULL),

7

      ((SELECT MAX(attempt\_id) FROM attempt),

8

      (SELECT question\_id FROM question, subject

9

        WHERE subject.subject\_id = question.subject\_id

10

        ORDER BY RAND()

11

        LIMIT 1), NULL),

12

      ((SELECT MAX(attempt\_id) FROM attempt),

13

      (SELECT question\_id FROM question, subject

14

        WHERE subject.subject\_id = question.subject\_id

15

        ORDER BY RAND()

16

        LIMIT 1), NULL);\*/

17

18

INSERT INTO testing(attempt\_id, question\_id, answer\_id)

19

SELECT attempt\_id, question\_id, NULL

20

FROM

21

  question

22

   JOIN attempt USING(subject\_id)

23

WHERE attempt\_id = (SELECT MAX(attempt\_id) FROM attempt)

24

ORDER BY RAND()

25

LIMIT 3;

26

​

27

SELECT\*FROM testing;

…………………….

SET @attempt\_id = (SELECT MAX(attempt\_id) FROM attempt);

INSERT INTO testing(attempt\_id, question\_id)

SELECT attempt\_id, question\_id

FROM question, attempt

/\*question

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

WHERE attempt\_id = @attempt\_id AND attempt.subject\_id = question.subject\_id

ORDER BY RAND()

LIMIT 3;

SELECT\*FROM testing;

## Задание

Студент прошел тестирование (то есть все его ответы занесены в таблицу **testing**), далее необходимо вычислить результат([запрос](https://stepik.org/lesson/310421/step/9?unit=292727)) и занести его в таблицу **attempt**для соответствующей попытки.  Результат попытки вычислить как количество правильных ответов, деленное на 3 (количество вопросов в каждой попытке) и умноженное на 100. Результат округлить до целого.

 Будем считать, что мы знаем **id**попытки,  для которой вычисляется результат, в нашем случае это 8. В таблицу **testing** занесены следующие ответы пользователя:

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

| testing\_id | attempt\_id | question\_id | answer\_id |

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

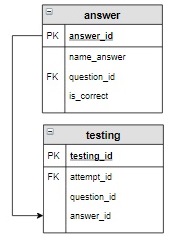
| 22 | 8 | 7 | 19 |

| 23 | 8 | 6 | 17 |

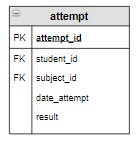
| 24 | 8 | 8 | 22 |

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

**Фрагмент логической схемы базы данных:**



**Корректируемая таблица:**



**Пояснение**

**Связанные шаги**

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

*Студент прошел тестирование (то есть все его ответы занесены в таблицу****testing****), далее необходимо вычислить результат([запрос](https://stepik.org/lesson/310421/step/9?unit=292727)) и занести его в таблицу****attempt****для соответствующей попытки.  Результат попытки вычислить как количество правильных ответов, деленное на 3 (количество вопросов в каждой попытке) и умноженное на 100. Результат округлить до целого.*

*Будем считать, что мы знаем****id****попытки,  для которой вычисляется результат, в нашем случае это 8.*

**Результат**

**Наполнение таблиц answer, attempt, testing(перед выполнением шага)**

1

SET @result\_null = (SELECT attempt\_id FROM attempt WHERE result IS NULL);

2

​

3

SET @result\_ = (SELECT ROUND(SUM(is\_correct /3\*100), 2) AS result

4

                FROM answer JOIN testing

5

                 ON answer.answer\_id = testing.answer\_id

6

                  WHERE attempt\_id = @result\_null);

7

8

UPDATE attempt

9

SET result = @result\_

10

WHERE attempt\_id = @result\_null;

11

SELECT\*FROM attempt;

12

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

13

UPDATE attempt a

14

SET result = (

15

   SELECT round(SUM(is\_correct) / 3 \* 100)

16

   FROM testing JOIN answer USING (answer\_id)

17

   WHERE attempt\_id = a.attempt\_id

18

   GROUP By attempt\_id

19

)

20

WHERE a.attempt\_id = 8;

21

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

22

UPDATE attempt,

23

(SELECT attempt\_id, round(count(testing.question\_id)/3\*100, 2) AS res

24

FROM testing INNER JOIN question

25

            ON testing.question\_id = question.question\_id

26

                       INNER JOIN answer

27

                       ON testing.answer\_id=answer.answer\_id and answer.is\_correct = True

28

GROUP BY attempt\_id) query\_in

29

SET attempt.result = query\_in.res

30

WHERE attempt.attempt\_id = 8;

## Задание

Удалить из таблицы**attempt**все попытки, выполненные раньше 1 мая 2020 года. Также удалить и все соответствующие этим попыткам вопросы из таблицы**testing**, которая создавалась следующим запросом:

CREATE TABLE testing (

testing\_id INT PRIMARY KEY AUTO\_INCREMENT,

attempt\_id INT,

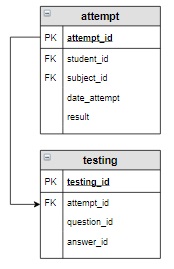
question\_id INT,

answer\_id INT,

FOREIGN KEY (attempt\_id) REFERENCES attempt (attempt\_id) ON DELETE CASCADE

);

**Фрагмент логической схемы базы данных:**



**Связанные шаги**

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

*Удалить из таблицы****attempt****все попытки, выполненные раньше 1 мая 2020 года. Также удалить и все соответствующие этим попыткам вопросы из таблицы****testing****.*

**Результат**

**Наполнение таблиц attempt, testing(перед выполнением шага)**

1

SET @delete\_date = (DATE('2020-05-01'));

2

​

3

DELETE FROM attempt

4

WHERE DATE(date\_attempt) < @delete\_date;

5

​

6

SELECT \* FROM attempt;

7

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

8

DELETE FROM attempt

9

WHERE DATEDIFF(date\_attempt, '2020-05-01')<0;

10

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

11

DELETE FROM attempt

12

WHERE MONTH(date\_attempt) < 5 AND YEAR(date\_attempt) <= 2020;

**Задание**

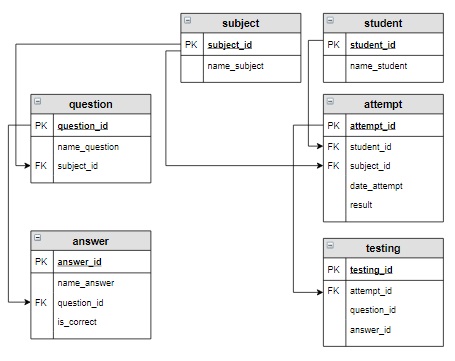
Придумайте один или несколько запросов корректировки данных для предметной области «Тестирование» (в таблицы занесены данные, как на первом шаге урока). Проверьте, правильно ли они работают.

При желании можно формулировку запросов  разместить в комментариях.

Размещенные задания можно реализовать для закрепления материала урока.

Оценивайте понравившиеся Вам запросы.

**Логическая схема базы данных:**



**Наполнение таблиц**

1

/\*1. Добавьте себя любимого (любимую) в таблицу студентов.

2

2. Вставьте в таблицу attempt:

3

2.1 свой student\_id,

4

2.2 все три предмета которые вы якобы сдали.

5

2.3 случайную дату для каждой попытки, такую чтобы дата рассчитывалась как: (сегодняшний день - случайное число дней от 1 до 12) . К примеру, если сегодня 15 января, дата сдачи любого вашего теста была от 3 до 15 января, выбранная случайным образом. Ибо система распознает жуликов, которые получают все три сертификата в один день.

6

2.4 В результат воткните себе 100 балов, чтобы все ботаны, кавказцы и евреи обзавидовались, а ваша мамка расцвела от гордости за своего ребёнка.

7

Примечания:

8

п.2.2 - используйте функцию CROSS JOIN, вставляйте только SELECTOM вставлять словом values - ниже вашего достоинства.

9

п. 2.3 - вычитать дату можно функцией

10

DATE\_ADD(NOW(), interval -10 DAY) # вернет дату, которая была 10 дней назад\*/

11

INSERT INTO student(name\_student)

12

SELECT 'Valentyn Mozul';

13

SELECT\*FROM student;

14

​

15

INSERT INTO attempt(student\_id, subject\_id, date\_attempt, result)

16

SELECT student\_id, subject\_id,

17

      DATE\_ADD(NOW(), INTERVAL FLOOR(RAND() \* 12) DAY), NULL

18

FROM student CROSS JOIN subject

19

WHERE name\_student = 'Valentyn Mozul';

20

SELECT\*FROM attempt;