Практическое задание

Урок 7. Расширенные возможности SQL

Задание

Работаем с базой данных учителей teachers.db. Для каждого задания надо сдать только код, который выполняется для получения результата, в текстовом файле.

1. Создайте представление, которое для каждого курса выводит название, номер последнего потока, дату начала обучения последнего потока и среднюю успеваемость курса по всем потокам.

2. Удалите из базы данных всю информацию, которая относится к преподавателю с идентификатором, равным 3. Используйте транзакцию.

3. Создайте триггер для таблицы успеваемости, который проверяет значение успеваемости на соответствие диапазону чисел от 0 до 5 включительно.

4. Дополнительное задание. Создайте триггер для таблицы потоков, который проверяет, что дата начала потока больше текущей даты, а номер потока имеет наибольшее значение среди существующих номеров. При невыполнении условий необходимо вызвать ошибку с информативным сообщением.

**

sqlite> .open teachers6.db

sqlite> .tables

courses grades streams teachers

```
sqlite> SELECT * FROM streams;
id  course_id  number  started_at  finished_at  students_amount
--  ---------  ------  ----------  -----------  ---------------
1   3          165     2020-08-18  0            34
2   2          178     2020-10-02  0            37
3   1          203     2020-11-12  0            35
4   1          210     2020-12-03  0            41

sqlite> SELECT * FROM grades;
id_teacher  id_stream  grade
----------  ---------  -----
3           1.0        4.7
2           2.0        4.9
1           3.0        4.8
1           4.0        4.9

sqlite> SELECT * FROM courses;
id  name
--  ----------------------
1   Базы данных
2   Основы Python
3   Linux. Рабочая станция

sqlite> SELECT * FROM teachers;
id  surname  name      email
--  -------  --------  ----------------------
1   Николай  Савельев  saveliev.n@mail.ru
2   Наталья  Петрова   petrova.n@yandex.ru
3   Елена    Малышева  malisheva.e@google.com
```

1. Создайте представление, которое для каждого курса выводит название, номер последнего потока, дату начала обучения последнего потока и среднюю успеваемость курса по всем потокам.

```
sqlite> CREATE VIEW course_info AS
   ...> SELECT courses.name,
   ...> MAX(streams.number),
   ...> streams.started_at,
   ...> AVG(grades.grade)
   ...> FROM courses
   ...> LEFT JOIN streams ON courses.id = streams.course_id
   ...> LEFT JOIN streams ON streams.id = grades.id_stream
   ...> GROUP BY name;
sqlite> SELECT * FROM CourseInfo;
```

2. Удалите из базы данных всю информацию, которая относится к преподавателю с идентификатором, равным 3. Используйте транзакцию.

```
sqlite> SELECT * FROM teachers;
id  surname  name      email
--  -------  --------  ---------------------
1   Николай  Савельев  saveliev.n@mail.ru
2   Наталья  Петрова   petrova.n@yandex.ru
3   Елена    Малышева  malisheva.e@google.com

sqlite> BEGIN TRANSACTION;
sqlite> DELETE FROM grades WHERE id_teacher = 3;
sqlite> DELETE FROM teachers WHERE id = 3;
sqlite> COMMIT;
```

```
sqlite> SELECT * FROM teachers;
id  surname  name      email
--  -------  --------  -------------------
1   Николай  Савельев  saveliev.n@mail.ru
2   Наталья  Петрова   petrova.n@yandex.ru
```

3. Создайте триггер для таблицы успеваемости, который проверяет значение успеваемости на соответствие диапазону чисел от 0 до 5 включительно.

```
sqlite> CREATE TRIGGER check_grade BEFORE INSERT
   ...> ON grades
   ...> BEGIN
   ...> SELECT CASE
   ...> WHEN
   ...> (NEW.grade NOT BETWEEN 1 AND 5)
   ...> THEN
   ...> RAISE(ABORT, 'grade should be between 1 and 5 !')
   ...> END;
   ...> END;
INSERT INTO grades (stream_id, teacher_id, grade) VALUES (5, 1, 2);
SELECT * FROM grades;
sqlite> INSERT INTO grades(id_stream, id_teacher, grade) VALUES (5,1,2);
sqlite> SELECT * FROM grades;
id_teacher  id_stream  grade
----------  ---------  -----
2           2.0        4.9
1           3.0        4.8
1           4.0        4.9
1           5.0        2.0
sqlite> INSERT INTO grades(id_stream, id_teacher, grade) VALUES (6,1,7);
Runtime error: grade should be between 1 and 5 ! (19)
```

4. Дополнительное задание. Создайте триггер для таблицы потоков, который проверяет, что дата начала потока больше текущей даты, а номер потока имеет наибольшее значение среди существующих номеров. При невыполнении условий необходимо вызвать ошибку с информативным сообщением.

...