**STUDENT ID : 2300355705**

**FULL NAME : NEEHARIKA MADDIPATIO GOPIKRISHNA**

**SCRIPTS : MySQL**

**SET 1:**

**QUESTION 1:**

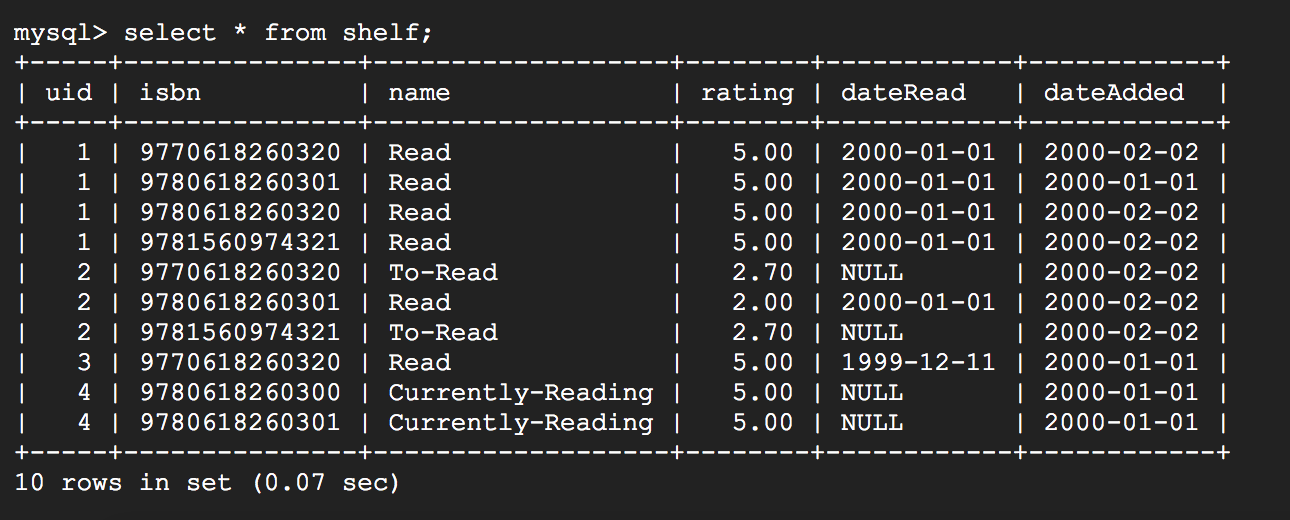
**QUERY:**

**Insert Query:**

INSERT INTO shelf

VALUES (2, 9780618260301,"Read", 2.00, "2000-01-01", "2000-02-02");

“shelf” table after inserting:



**Updation Query:**

UPDATE book

SET avgrating = (

SELECT z.sum\_rating/z.no\_of\_entries AS new\_avg

FROM (

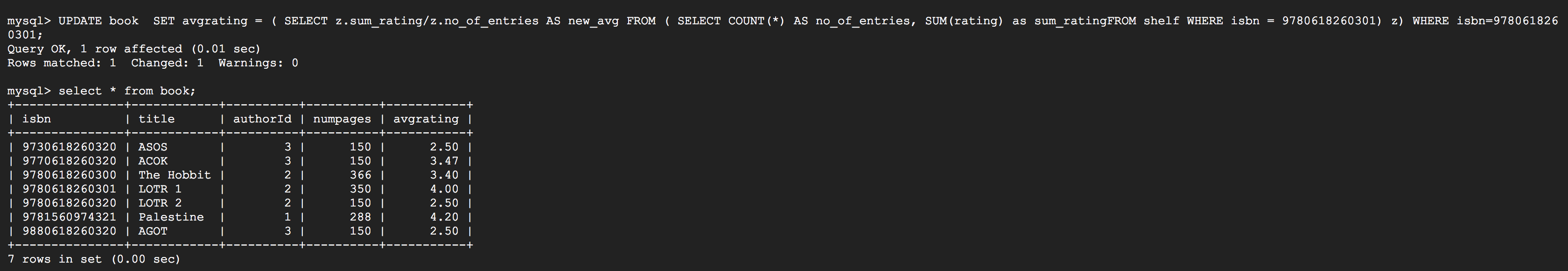
SELECT COUNT(\*) AS no\_of\_entries, SUM(rating) as sum\_rating FROM shelf

WHERE isbn = 9780618260301) z

)

WHERE isbn=9780618260301;

**OUTPUT:**



**EXPLANATION:**

The inner most query calculates the sum of all the ratings for that book and the total number of ratings. The next outer query calculate the average by dividing both the values. This new average is then updated in the table “book” for that particular isbn.

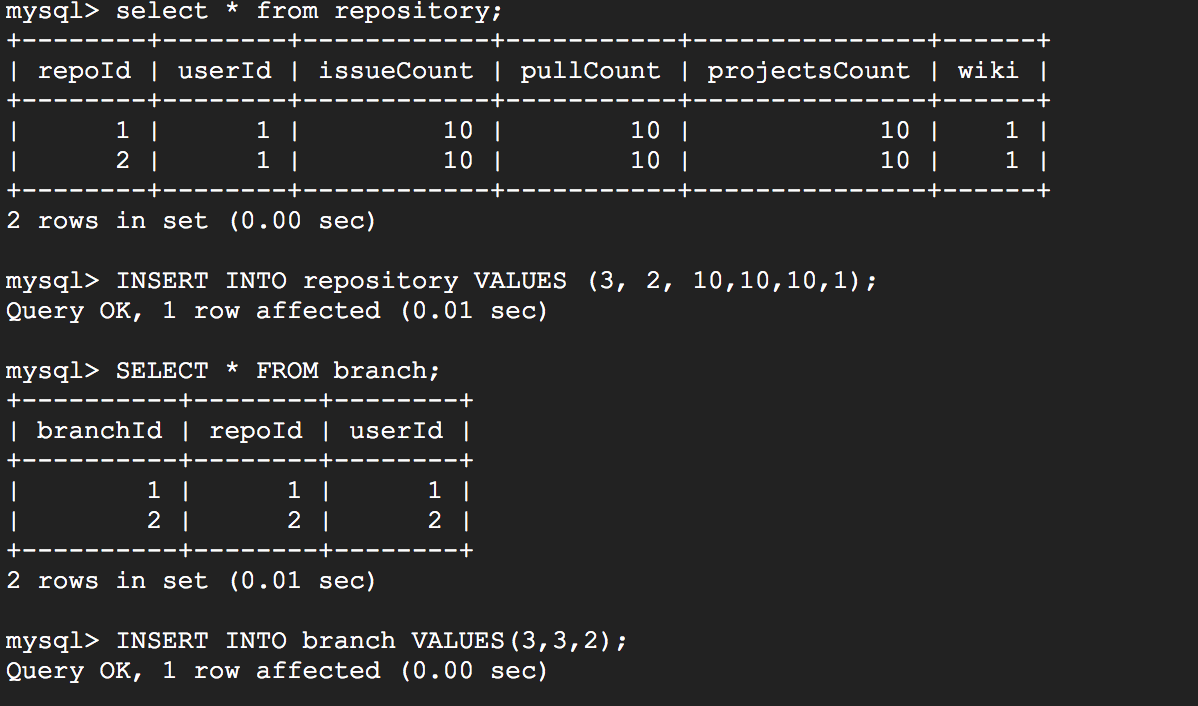
**QUESTION 2:**

**QUERY:**

**SET 2:**

**QUESTION 1:**

**ASSUMPTIONS:** Since there are no 3 different repoID’s in the given table I have added another repoID.

****

**QUERY:**

SELECT userId

FROM branch

WHERE (repoId=1) OR (repoId=2)

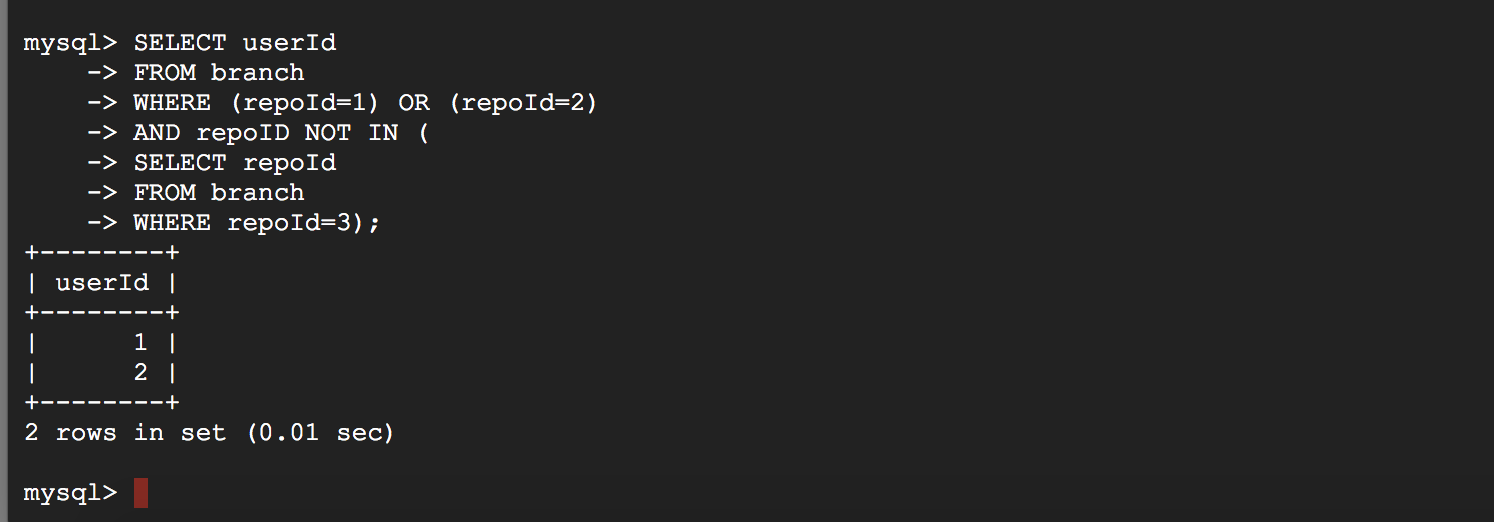
AND repoID NOT IN

( SELECT repoId

FROM branch

WHERE repoId=3);

**OUTPUT:**

****

**EXPLANATION:**

X,Y and Z are hardcoded as 1,2 and 3 respectively. The inner query collects all the users with repoId =3 . The main select outputs the users who made branches of repositories1 and 2 but not repository 3 (all the users not included in the subquery results.)

**QUESTION 2:**

**QUERY:**

SELECT \*

FROM commits

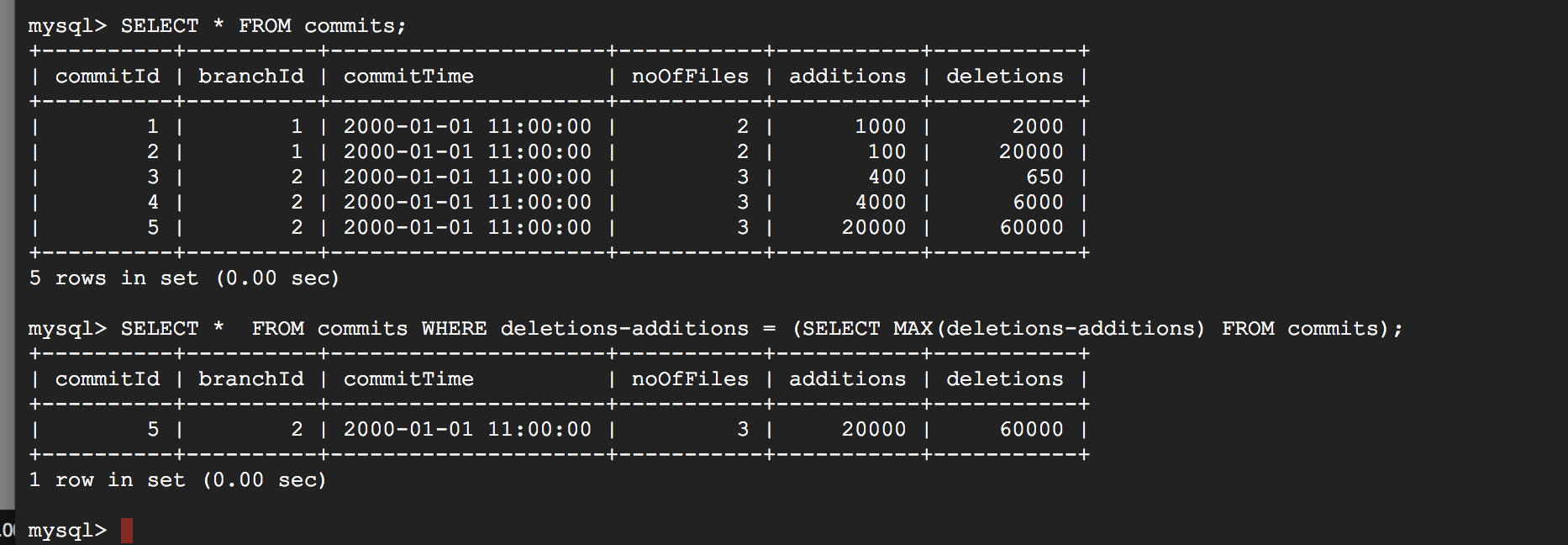
WHERE deletions-additions = (

SELECT MAX(deletions-additions)

FROM commits

);

**OUTPUT:**

****

**EXPLANATION:**

The inner query calculates the highest lines of code reduced but subtracting additions from deletions. The final select displays all the commits with highest lines of code reduced.

SELECT userId, count(\*) as creations,

FROM User u

JOIN Issues I ON u.userID=I.creatorId