Title: Transaction Integrity Check

The code performs an update operation on the `PawneeCommons.works` table and checks if the update violates certain constraints. If the constraints are violated, the transaction is rolled back; otherwise, it is committed.

Code Explanation:

1. Start the Transaction:
   - The code begins by initiating a transaction. A transaction is a series of one or more SQL statements executed as a single unit.

2. Update Operation:
   - An update operation is performed on the `PawneeCommons.works` table. The code attempts to set the job title to 'Producer' for a specific record identified by `pID` and `eID`.

3. Allowed Job Titles:
   - A set of allowed job titles (Cast, written_by, directed_by) is specified using the `@allowed_job_titles` variable.

4. Integrity Constraint Check:
   - The code checks if the updated job title ('Producer') is in the allowed list using the `FIND_IN_SET` function. The result is stored in the `@job_check` variable.

5. Condition Evaluation:
   - A variable, `@rollback_required`, is set to 0 initially. The code then checks if the job title violates the allowed titles and updates the variable accordingly.

6. Rollback or Commit:
   - Depending on the condition, the code determines whether to rollback or commit the transaction. If `@rollback_required` is 1, meaning the constraint is violated, the transaction is rolled back; otherwise, it is committed.

7. Result Output:
   - The final result is displayed as 'Transaction Rolled Back' or 'Transaction Committed' based on the outcome of the condition.

Conclusion:

This code checks that only specified job titles are allowed in the database, and any update with an invalid title will result in a rollback of the transaction. This is critical for data integrity and consistency of the database.

Example:Rollback:

```sql
1  -- Start the transaction
2  START TRANSACTION;
3
4  -- Update operation
5  UPDATE PawneeCommons.works
6  SET job = 'cook'
7  WHERE pID = 1 AND eID = 's7e12';
8
9  -- Set the allowed job titles
10 SET @allowed_job_titles := 'Cast,written_by,directed_by';
11
12 -- Check the integrity conditions
13 SET @job_check := FIND_IN_SET('cook', @allowed_job_titles COLLATE utf8mb4_general_ci);
14
15 -- Store the result of the condition
16 SET @rollback_required := 0;
17
18 -- Check the condition and set the variable
19 SELECT IF(@job_check = 0, @rollback_required := 1, @rollback_required);
20
21 -- Rollback or commit based on the condition
22 SELECT CASE
23     WHEN @rollback_required = 1 THEN
24         -- Rollback the transaction if the constraint is not satisfied
25         'Transaction Rolled Back'
26     ELSE
27         -- Commit the transaction if the constraint is satisfied
28         'Transaction Committed'
29 END AS Result;
```

>

IF(@job_check = 0, @rollback_required := 1, @rollback_required)

1 | 1

Result
1	Transaction Rolled Back

Example:Commit

localhost - PawneeCommons ⌄   Favorites ⌄   History ⌄   Run a

```
 1  -- Start the transaction
 2  START TRANSACTION;
 3
 4  -- Update operation
 5  UPDATE PawneeCommons.works
 6  SET job = 'written_by'
 7  WHERE pID = 1 AND eID = 's7e12';
 8
 9  -- Set the allowed job titles
10  SET @allowed_job_titles := 'Cast,written_by,directed_by';
11
12  -- Check the integrity conditions
13  SET @job_check := FIND_IN_SET('written_by', @allowed_job_titles COLLATE utf8mb4_general_ci);
14
15  -- Store the result of the condition
16  SET @rollback_required := 0;
17
18  -- Check the condition and set the variable
19  SELECT IF(@job_check = 0, @rollback_required := 1, @rollback_required);
20
21  -- Rollback or commit based on the condition
22  SELECT CASE
23      WHEN @rollback_required = 1 THEN
24          -- Rollback the transaction if the constraint is not satisfied
25          'Transaction Rolled Back'
26      ELSE
27          -- Commit the transaction if the constraint is satisfied
28          'Transaction Committed'
29  END AS Result;
```

IF(@job_check = 0, @rollback_required := 1, @rollback_required)

1 0

Result

1 Transaction Committed

Activate SQLPro Premium
to access results.