

TRIGGER

Code:

```
CREATE OR REPLACE FUNCTION new_user_song_update()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$

DECLARE
    songId INT;

BEGIN
    SELECT floor(random() * (SELECT MAX(songId) FROM songs) + 1)::int
    AS songId;

    INSERT INTO
        userSong(userId , songId)
        VALUES (NEW.userId , songId);

    RETURN NEW;

END;
$$
```

```
CREATE TRIGGER new_user_song  
AFTER INSERT  
ON users  
FOR EACH ROW  
EXECUTE PROCEDURE new_user_song_update();
```

FUNCTION

Code:

```
DECLARE
```

```
    latestDate DATE := NOW()::date;
```

```
    dayDiff INT;
```

```
BEGIN
```

```
    dayDiff = latestDate - lastChanged;
```

```
    IF(dayDiff > 7) THEN
```

```
        RETURN latestDate;
```

```
    ELSE
```

```
        RETURN lastChanged;
```

```
    END IF;
```

```
END;
```