Assignment-5

Team-3

Joseph Krueger - jpk170030

Prudhveeraj Botta - pxb220018

Rohan Jayachandran - rxj220025

Airbnb System PL/SQL Stored Procedures and Triggers

Objective: Define two relevant stored procedures and two triggers (they should have a meaningful application in real-world cases).

Procedure 1:

```
/* Add_User adds a new row to the User table */
CREATE PROCEDURE Add_User(accountID IN User.accountid%TYPE, firstName IN
User.firstName%TYPE, lastName IN User.lastName%TYPE, birthDate IN User.birthDate%TYPE,
personalEmail IN User.personalEmail%TYPE, workEmail IN User.workEmail%TYPE,
phoneNumber IN User.phoneNumber%TYPE, profilePhotoLink IN User.profilePhotoLink%TYPE,
description IN User.description%TYPE, address IN User.address%TYPE) AS
BEGIN
	INSERT INTO User VALUES(accountID, firstName, lastName, birthDate, personalEmail,
workEmail, phoneNumber, profilePhotoLink, description, address);
END;
```

Procedure 2:

```
/* Update_House_Rules changes the houseRules attribute in a particular row of the
Rental_Listing table */
CREATE PROCEDURE Update_House_Rules(newRules IN Rental_Listing.houseRules%TYPE,
listingNum IN Rental_Listing.listingID%TYPE) AS
BEGIN
	UPDATE Rental_Listing SET houseRules = newRules WHERE listingID = listingNum;
END;
```

Trigger 1:

```
/* When row is added to Host_Review table, add one to reviewsAbout in Guest table and add
one to reviewsWritten in Host table */
```

```
CREATE TRIGGER Add_Host_Review
AFTER INSERT ON Host_Review
FOR EACH ROW
BEGIN
        UPDATE Guest
        SET reviewsAbout = reviewsAbout + 1
        WHERE accountID IN
                (SELECT accountID
                FROM Host_Review HR, BookingA BA, BookingC BC, Guest G
                WHERE HR.bookingID = BA.bookingID AND
                        BA.bookingID = BC.bookingID AND
                        BC.accountID = G.accountID);
        UPDATE Host
        SET reviewsWritten = reviewsWritten + 1
        WHERE accountID IN
                (SELECT accountID
                FROM Host_Review HR, BookingA BA, BookingC BC, ListingB LB, Host H
                WHERE HR.bookingID = BA.bookingID AND
                        BA.bookingID = BC.bookingID AND
                        BC.listingID = LB.listingID AND
                        LB.accountID = H.accountID);
END;
```

Trigger 2:

```
/* When new startDate/endDate added in BookingA, calculate new listPrice in BookingB */
/* listPrice = (endDate - startDate) * pricePerNight */
CREATE TRIGGER Calculate_List_Price
AFTER INSERT ON BookingA
FOR EACH ROW
DECLARE
        ppn number;
        start date;
        end date;
BEGIN
        SELECT pricePerNight into ppn
```

```
    FROM Rental_Listing RL, ListingA LA, BookingB BB, BookingC BC, BookingA BA
    WHERE RL.listingID = LA.listingID AND
        LA.listingID = BB.listingID AND
        LA.listingID = BC.listingID AND
        BC.bookingID = BA.bookingID AND
        BA.bookingID = :new.bookingID;

    SELECT startDate into start
    FROM BookingA BA, BookingC BC, ListingA LA, BookingB BB
    WHERE BA.bookingID = BC.bookingID AND
        BC.listingID = LA.listingID AND
        LA.listingID = BB.listingID AND
        BC.bookingID = BA.bookingID AND
        BA.bookingID = :new.bookingID;

    SELECT endDate into end
    FROM BookingA BA, BookingC BC, ListingA LA, BookingB BB
    WHERE BA.bookingID = BC.bookingID AND
        BC.listingID = LA.listingID AND
        LA.listingID = BB.listingID AND
        BC.bookingID = BA.bookingID AND
        BA.bookingID = :new.bookingID;

    UPDATE BookingB
    SET listPrice = (end - start) * ppn
    WHERE listingID IN
        (SELECT listingID
        FROM BookingB BB, ListingA LA, BookingC BC, BookingA BA
        WHERE BB.listingID = LA.listingID AND
            LA.listingID = BC.listingID AND
            BC.bookingID = BA.bookingID AND
            BA.bookingID = :new.bookingID);
END;
```