Share Itinerary Stored Procedure and Transaction

```sql
DELIMITER //
CREATE PROCEDURE ShareItinerary(
    IN p_itinerary_id INT,
    IN p_source_user_id INT,
    IN p_target_email VARCHAR(255),
    OUT p_new_itinerary_id INT,
    OUT p_target_user_name VARCHAR(255),
    OUT p_items_copied INT,
    OUT p_categories_count INT
)
BEGIN
    DECLARE v_target_user_id INT;
    DECLARE v_source_itinerary_exists BOOLEAN DEFAULT FALSE;
    DECLARE v_owner_name VARCHAR(255);
    DECLARE v_destination_city VARCHAR(100);
    DECLARE v_destination_state VARCHAR(100);
    DECLARE v_start_date DATE;
    DECLARE v_end_date DATE;
    DECLARE v_table_exists BOOLEAN DEFAULT FALSE;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    START TRANSACTION;

    SELECT
        COUNT(*) > 0,
        i.destination_city,
        i.destination_state,
        i.start_date,
        i.end_date,
        u.name
    INTO
        v_source_itinerary_exists,
        v_destination_city,
        v_destination_state,
        v_start_date,
        v_end_date,
```

```sql
    v_owner_name
FROM Itinerary i
JOIN User u ON i.user_id = u.user_id
WHERE i.itinerary_id = p_itinerary_id
AND i.user_id = p_source_user_id;

IF NOT v_source_itinerary_exists THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Itinerary not found or unauthorized';
END IF;

SELECT user_id, name INTO v_target_user_id, p_target_user_name
FROM User
WHERE email = p_target_email;

IF v_target_user_id IS NULL THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Target user not found';
END IF;

INSERT INTO Itinerary (
    user_id,
    destination_city,
    destination_state,
    start_date,
    end_date
) VALUES (
    v_target_user_id,
    v_destination_city,
    v_destination_state,
    v_start_date,
    v_end_date
);

SET p_new_itinerary_id = LAST_INSERT_ID();

INSERT INTO ItineraryItem (
    itinerary_id,
    attraction_id,
    day_number,
    start_time,
    end_time,
    notes
)
SELECT
    p_new_itinerary_id,
```

```
        attraction_id,
        day_number,
        start_time,
        end_time,
        notes
    FROM ItineraryItem
    WHERE itinerary_id = p_itinerary_id;

    SELECT
        COUNT(*) INTO p_items_copied
    FROM ItineraryItem
    WHERE itinerary_id = p_new_itinerary_id;

    SELECT
        COUNT(DISTINCT a.main_category) INTO p_categories_count
    FROM ItineraryItem ii
    JOIN Attraction a ON ii.attraction_id = a.attraction_id
    WHERE ii.itinerary_id = p_new_itinerary_id;

    SELECT COUNT(*) > 0 INTO v_table_exists
    FROM information_schema.tables
    WHERE table_schema = DATABASE()
    AND table_name = 'ItineraryShare';

    IF v_table_exists THEN
        INSERT INTO ItineraryShare (
            source_itinerary_id,
            target_itinerary_id,
            source_user_id,
            target_user_id,
            share_date,
            custom_message
        ) VALUES (
            p_itinerary_id,
            p_new_itinerary_id,
            p_source_user_id,
            v_target_user_id,
            NOW(),
            CONCAT('Shared from ', v_owner_name)
        );
    END IF;

    COMMIT;
END //
```

DELIMITER ;

Attraction Recommendation Stored Procedure

```
DELIMITER //
CREATE PROCEDURE GetRecommendationsForItinerary(
    IN p_itinerary_id INT,
    IN p_user_id INT,
    IN p_limit INT
)
BEGIN
    DECLARE v_city VARCHAR(100);
    DECLARE v_state VARCHAR(100);

    SELECT destination_city, destination_state
    INTO v_city, v_state
    FROM Itinerary
    WHERE itinerary_id = p_itinerary_id AND user_id = p_user_id;

    IF v_city IS NULL THEN
        SELECT NULL;
        LEAVE GetRecommendationsForItinerary;
    END IF;

    SELECT
        a.*,
        CASE
            WHEN a.main_category = 'Park' THEN u.park_pref
            WHEN a.main_category = 'Historical Landmark' THEN u.historical_landmark_pref
            WHEN a.main_category = 'Historical Place Museum' THEN
u.historical_place_museum_pref
            WHEN a.main_category = 'Museum' THEN u.museum_pref
            WHEN a.main_category = 'History Museum' THEN u.history_museum_pref
            WHEN a.main_category = 'Tourist Attraction' THEN u.tourist_attraction_pref
            WHEN a.main_category = 'Wildlife Park' THEN u.wildlife_park_pref
            WHEN a.main_category = 'Art Museum' THEN u.art_museum_pref
            WHEN a.main_category = 'Aquarium' THEN u.aquarium_pref
            WHEN a.main_category = 'Monument' THEN u.monument_pref
            WHEN a.main_category = 'Hiking Area' THEN u.hiking_area_pref
            WHEN a.main_category = 'Zoo' THEN u.zoo_pref
            WHEN a.main_category = 'Catholic Cathedral' THEN u.catholic_cathedral_pref
            WHEN a.main_category = 'Nature Preserve' THEN u.nature_preserve_pref
            WHEN a.main_category = 'Amusement Park' THEN u.amusement_park_pref
            WHEN a.main_category = 'Garden' THEN u.garden_pref
            WHEN a.main_category = 'Theme Park' THEN u.theme_park_pref
            WHEN a.main_category = 'Water Park' THEN u.water_park_pref
```

```sql
            WHEN a.main_category = 'Scenic Spot' THEN u.scenic_spot_pref
            WHEN a.main_category = 'Observatory' THEN u.observatory_pref
            WHEN a.main_category = 'Castle' THEN u.castle_pref
            WHEN a.main_category = 'Archaeological Museum' THEN
u.archaeological_museum_pref
            WHEN a.main_category = 'Public Beach' THEN u.public_beach_pref
            WHEN a.main_category = 'National Forest' THEN u.national_forest_pref
            WHEN a.main_category = 'Catholic Church' THEN u.catholic_church_pref
            WHEN a.main_category = 'Heritage Museum' THEN u.heritage_museum_pref
            WHEN a.main_category = 'Beach' THEN u.beach_pref
            WHEN a.main_category = 'Synagogue' THEN u.synagogue_pref
            WHEN a.main_category = 'Ecological Park' THEN u.ecological_park_pref
            WHEN a.main_category = 'Wax Museum' THEN u.wax_museum_pref
            WHEN a.main_category = 'Hindu Temple' THEN u.hindu_temple_pref
            WHEN a.main_category = 'Wildlife Safari Park' THEN u.wildlife_safari_park_pref
            WHEN a.main_category = 'Buddhist Temple' THEN u.buddhist_temple_pref
            WHEN a.main_category = 'Animal Park' THEN u.animal_park_pref
            WHEN a.main_category = 'Wildlife Refuge' THEN u.wildlife_refuge_pref
            WHEN a.main_category = 'Heritage Building' THEN u.heritage_building_pref
            WHEN a.main_category = 'Vista Point' THEN u.vista_point_pref
            WHEN a.main_category = 'National Park' THEN u.national_park_pref
            WHEN a.main_category = 'Monastery' THEN u.monastery_pref
            WHEN a.main_category = 'Fortress' THEN u.fortress_pref
            WHEN a.main_category = 'Beach Pavilion' THEN u.beach_pavilion_pref
            ELSE 3
        END AS preference_score
    FROM
        Attraction a
        JOIN User u ON u.user_id = p_user_id
        LEFT JOIN (
            SELECT attraction_id
            FROM ItineraryItem
            WHERE itinerary_id = p_itinerary_id
        ) AS added ON a.attraction_id = added.attraction_id
    WHERE
        a.city = v_city
        AND a.state = v_state
        AND added.attraction_id IS NULL
    ORDER BY
        preference_score DESC,
        a.rating DESC
    LIMIT p_limit;
END //
DELIMITER ;
```

Trigger after adding item to itinerary

```
DELIMITER //
CREATE TRIGGER after_itinerary_item_insert
AFTER INSERT ON ItineraryItem
FOR EACH ROW
BEGIN
    DECLARE v_category VARCHAR(100);
    DECLARE v_user_id INT;

    SELECT a.main_category, i.user_id INTO v_category, v_user_id
    FROM Attraction a
    JOIN Itinerary i ON i.itinerary_id = NEW.itinerary_id
    WHERE a.attraction_id = NEW.attraction_id;

    IF v_category IS NOT NULL AND v_user_id IS NOT NULL THEN
        CASE LOWER(REPLACE(REPLACE(v_category, ' ', '_'), '-', '_'))
            WHEN 'park' THEN
                UPDATE User SET park_pref = LEAST(IFNULL(park_pref, 0) + 1, 100)
                WHERE user_id = v_user_id;
            WHEN 'historical_landmark' THEN
                UPDATE User SET historical_landmark_pref =
LEAST(IFNULL(historical_landmark_pref, 0) + 1, 100)
                WHERE user_id = v_user_id;
            WHEN 'historical_place_museum' THEN
                UPDATE User SET historical_place_museum_pref =
LEAST(IFNULL(historical_place_museum_pref, 0) + 1, 100)
                WHERE user_id = v_user_id;
            WHEN 'museum' THEN
                UPDATE User SET museum_pref = LEAST(IFNULL(museum_pref, 0) + 1, 100)
                WHERE user_id = v_user_id;
            WHEN 'history_museum' THEN
                UPDATE User SET history_museum_pref = LEAST(IFNULL(history_museum_pref, 0)
+ 1, 100)
                WHERE user_id = v_user_id;
            WHEN 'tourist_attraction' THEN
                UPDATE User SET tourist_attraction_pref = LEAST(IFNULL(tourist_attraction_pref, 0)
+ 1, 100)
                WHERE user_id = v_user_id;
            WHEN 'wildlife_park' THEN
                UPDATE User SET wildlife_park_pref = LEAST(IFNULL(wildlife_park_pref, 0) + 1,
100)
                WHERE user_id = v_user_id;
            WHEN 'art_museum' THEN
```

```sql
        UPDATE User SET art_museum_pref = LEAST(IFNULL(art_museum_pref, 0) + 1,
100)
        WHERE user_id = v_user_id;
      WHEN 'aquarium' THEN
        UPDATE User SET aquarium_pref = LEAST(IFNULL(aquarium_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'monument' THEN
        UPDATE User SET monument_pref = LEAST(IFNULL(monument_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'hiking_area' THEN
        UPDATE User SET hiking_area_pref = LEAST(IFNULL(hiking_area_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'zoo' THEN
        UPDATE User SET zoo_pref = LEAST(IFNULL(zoo_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'catholic_cathedral' THEN
        UPDATE User SET catholic_cathedral_pref =
LEAST(IFNULL(catholic_cathedral_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'nature_preserve' THEN
        UPDATE User SET nature_preserve_pref = LEAST(IFNULL(nature_preserve_pref, 0)
+ 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'amusement_park' THEN
        UPDATE User SET amusement_park_pref = LEAST(IFNULL(amusement_park_pref,
0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'garden' THEN
        UPDATE User SET garden_pref = LEAST(IFNULL(garden_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'theme_park' THEN
        UPDATE User SET theme_park_pref = LEAST(IFNULL(theme_park_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'water_park' THEN
        UPDATE User SET water_park_pref = LEAST(IFNULL(water_park_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'scenic_spot' THEN
        UPDATE User SET scenic_spot_pref = LEAST(IFNULL(scenic_spot_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'observatory' THEN
        UPDATE User SET observatory_pref = LEAST(IFNULL(observatory_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'castle' THEN
        UPDATE User SET castle_pref = LEAST(IFNULL(castle_pref, 0) + 1, 100)
```

```
          WHERE user_id = v_user_id;
        WHEN 'archaeological_museum' THEN
          UPDATE User SET archaeological_museum_pref =
LEAST(IFNULL(archaeological_museum_pref, 0) + 1, 100)
          WHERE user_id = v_user_id;
        WHEN 'public_beach' THEN
          UPDATE User SET public_beach_pref = LEAST(IFNULL(public_beach_pref, 0) + 1,
100)
          WHERE user_id = v_user_id;
        WHEN 'national_forest' THEN
          UPDATE User SET national_forest_pref = LEAST(IFNULL(national_forest_pref, 0) +
1, 100)
          WHERE user_id = v_user_id;
        WHEN 'catholic_church' THEN
          UPDATE User SET catholic_church_pref = LEAST(IFNULL(catholic_church_pref, 0) +
1, 100)
          WHERE user_id = v_user_id;
        WHEN 'heritage_museum' THEN
          UPDATE User SET heritage_museum_pref =
LEAST(IFNULL(heritage_museum_pref, 0) + 1, 100)
          WHERE user_id = v_user_id;
        WHEN 'beach' THEN
          UPDATE User SET beach_pref = LEAST(IFNULL(beach_pref, 0) + 1, 100)
          WHERE user_id = v_user_id;
        WHEN 'synagogue' THEN
          UPDATE User SET synagogue_pref = LEAST(IFNULL(synagogue_pref, 0) + 1, 100)
          WHERE user_id = v_user_id;
        WHEN 'ecological_park' THEN
          UPDATE User SET ecological_park_pref = LEAST(IFNULL(ecological_park_pref, 0) +
1, 100)
          WHERE user_id = v_user_id;
        WHEN 'wax_museum' THEN
          UPDATE User SET wax_museum_pref = LEAST(IFNULL(wax_museum_pref, 0) + 1,
100)
          WHERE user_id = v_user_id;
        WHEN 'hindu_temple' THEN
          UPDATE User SET hindu_temple_pref = LEAST(IFNULL(hindu_temple_pref, 0) + 1,
100)
          WHERE user_id = v_user_id;
        WHEN 'wildlife_safari_park' THEN
          UPDATE User SET wildlife_safari_park_pref =
LEAST(IFNULL(wildlife_safari_park_pref, 0) + 1, 100)
          WHERE user_id = v_user_id;
        WHEN 'buddhist_temple' THEN
```

```sql
        UPDATE User SET buddhist_temple_pref = LEAST(IFNULL(buddhist_temple_pref, 0)
+ 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'animal_park' THEN
        UPDATE User SET animal_park_pref = LEAST(IFNULL(animal_park_pref, 0) + 1,
100)
        WHERE user_id = v_user_id;
      WHEN 'wildlife_refuge' THEN
        UPDATE User SET wildlife_refuge_pref = LEAST(IFNULL(wildlife_refuge_pref, 0) + 1,
100)
        WHERE user_id = v_user_id;
      WHEN 'heritage_building' THEN
        UPDATE User SET heritage_building_pref = LEAST(IFNULL(heritage_building_pref,
0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'vista_point' THEN
        UPDATE User SET vista_point_pref = LEAST(IFNULL(vista_point_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'national_park' THEN
        UPDATE User SET national_park_pref = LEAST(IFNULL(national_park_pref, 0) + 1,
100)
        WHERE user_id = v_user_id;
      WHEN 'monastery' THEN
        UPDATE User SET monastery_pref = LEAST(IFNULL(monastery_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'fortress' THEN
        UPDATE User SET fortress_pref = LEAST(IFNULL(fortress_pref, 0) + 1, 100)
        WHERE user_id = v_user_id;
      WHEN 'beach_pavilion' THEN
        UPDATE User SET beach_pavilion_pref = LEAST(IFNULL(beach_pavilion_pref, 0) +
1, 100)
        WHERE user_id = v_user_id;
      ELSE
        BEGIN
        END;
    END CASE;
  END IF;
END //
DELIMITER ;
```