Phase – 5

Apex Trigger & Handler for Booking Seat Management

1. Purpose

The Apex Trigger and Handler manage **Package Slot seat availability** automatically:

- Reduces seats when a booking is **confirmed**.
- Restores seats when a booking is **cancelled** or **deleted**.
- Adjusts seats if the **Number_of_people__c** field changes in an existing booking.
- Ensures **bulk-safe operations** and prevents inconsistencies.

---

2. Objects & Fields

Object	API Name	Key Fields
Booking	Booking__c	Package_Slot__c (Lookup), Number_of_people__c (Number), Booking_Status__c (Picklist)
Package Slot	Package_Slot__c	Available_Seats__c (Number), Capacity__c (Number)

Number_of_people__c replaces Num_Pax__c from previous versions.

---

3. Apex Trigger: **BookingTrigger**

Events Triggered:

- before insert, before update → optional validation (currently unused)
- after insert, after update → update seat counts
- after delete → restore seats

Code Summary:

```
trigger BookingTrigger on Booking__c (before insert, before update, after insert, after update, after delete) {
    if (Trigger.isBefore) {
        if (Trigger.isInsert || Trigger.isUpdate) {
            BookingTriggerHandler.beforeSave(Trigger.new, Trigger.oldMap);
        }
    }
```

```
    if (Trigger.isAfter) {
        if (Trigger.isInsert || Trigger.isUpdate) {
            BookingTriggerHandler.afterSave(Trigger.new, Trigger.oldMap);
        }
        if (Trigger.isDelete) {
            BookingTriggerHandler.afterDelete(Trigger.old);
        }
    }
}
```

---

4. Handler Class: **BookingTriggerHandler**

Responsibilities:

1. **beforeSave()**
    - Placeholder for validation logic.
    - Can implement overbooking checks using addError().
2. **afterSave()**
    - Reduces or restores seats after insert/update.
    - Handles status transitions: Confirmed → Cancelled, Cancelled → Confirmed.
    - Adjusts Available_Seats__c if Number_of_people__c changes.
    - Uses **Map<Id, Package_Slot__c>** to bulkify updates.
3. **afterDelete()**
    - Restores seats for deleted confirmed bookings.
    - Bulk-safe: updates all affected slots in one DML operation.

Bulkification Strategy:

- Collect all affected Package Slot IDs in a **Set**.
- Query all relevant slots **once**.
- Apply seat changes in memory.
- Update slots in a single DML operation.

---

5. Test Class: **BookingTriggerTest**

Test Scenarios Covered:

1. Insert confirmed booking → reduces available seats.
2. Cancel booking → restores available seats.
3. Delete booking → restores seats.
4. Insert and delete additional bookings to validate bulk behavior.
5. (Optional) Validate overbooking prevention if implemented in beforeSave.

Assertions Example:

```
System.assertEquals(8, slot.Available_Seats__c, 'Seats should reduce by 2');
System.assertEquals(10, slot.Available_Seats__c, 'Seats restored after cancellation');
```

---

6. Key Points / Best Practices

- One trigger per object; delegate logic to a handler class.
- Bulk-safe operations using **Maps** and **Sets**.
- Seat adjustments consider all booking status transitions.
- Test class ensures **minimum 75% coverage**.