
HOTEL MANAGMENT

DBMS Project under:-
Dr. AMRITA DAHIYA



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

MEMBERS

NISHTHITA SINGH(102203265)
PREETMANNAT (102203283)
KHUSHI DAHIYA (102203339)
ADITYAVIR SINGH(102203278)
YUVRAJ SINGH BRAR(102203293)

INDEX

INTRODUCTION TO TOPIC

ER DIAGRAM

ER DIAGRAM TO TABLE

CREATION OF TABLE

SNAPSHOT

Background of the Project

The hospitality industry, particularly the hotel sector, relies heavily on efficient management systems to streamline operations, enhance guest experiences, and ensure smooth day-to-day functioning. A Hotel Management System (HMS) automates various aspects of hotel operations, including reservations, room assignments, billing, housekeeping, and reporting. Such systems play a crucial role in optimizing workflows, improving guest satisfaction, and maximizing revenue for hotel businesses.

Need of the Project

1. **Operational Efficiency:** An HMS helps in automating routine tasks such as reservation management, guest check-ins, and housekeeping schedules, thereby reducing manual effort and minimizing errors.
2. **Enhanced Guest Experience:** With features like online booking, quick check-ins, and personalized services, guests experience smoother interactions and better service quality.
3. **Centralized Information:** An HMS provides a centralized repository for guest data, room availability, billing details, and more, facilitating quick access to critical information.
4. **Reporting and Analytics:** The system generates comprehensive reports on occupancy rates, revenue streams, guest preferences, and other metrics, aiding in strategic decision-making.
5. **Revenue Optimization:** By efficiently managing room inventory, rates, and promotional offers, the system helps in maximizing revenue and occupancy rates.

Project Outcomes

1. **Functional Hotel Management System:** Develop a fully functional HMS capable of handling tasks like guest bookings, room management, billing, and reporting.
2. **User-Friendly Interface:** Create an intuitive user interface for hotel staff to efficiently manage day-to-day operations.

3. Improved Operational Efficiency: Automate routine tasks to reduce manual effort and improve overall operational efficiency.
4. Enhanced Guest Satisfaction: Implement features like quick reservations, personalized services, and streamlined check-ins to enhance guest satisfaction.
5. Insightful Reporting: Develop reporting modules to provide valuable insights into hotel performance, helping management make informed decisions.

PL/SQL in the Project

PL/SQL (Procedural Language/Structured Query Language) is an ideal choice for implementing a Hotel Management System due to its integration capabilities with Oracle databases and its procedural programming features. Here's how PL/SQL can be leveraged:

1. Database Operations: Use PL/SQL for database interactions such as retrieving guest details, updating room availability, and processing billing transactions.
2. Stored Procedures: Implement stored procedures to encapsulate complex business logic within the database, ensuring data integrity and security.
3. Data Validation: Utilize PL/SQL to enforce data validation rules, ensuring that only valid and consistent data is stored in the database.
4. Transaction Management: PL/SQL supports transaction management, allowing for atomic and consistent database operations, critical for financial transactions and booking updates.
5. Reporting and Analytics: PL/SQL can be used to generate custom reports and perform data analysis directly within the database, leveraging its computational capabilities.

By incorporating PL/SQL into your Hotel Management System, you can build a robust and scalable solution that efficiently manages hotel operations while leveraging the power of Oracle databases for data storage and retrieval. This combination ensures data integrity, performance, and security essential for a mission-critical application like an HMS.



TERMINOLOGIES AND SYMBOLS OF ER DIAGRAM:



Represents Entity



Represents Attribute



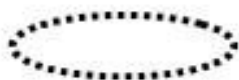
Represents Relationship



Links Attribute(s) to entity set(s) or
Entity set(s) to Relationship set(s)



Represents Multivalued Attributes



Represents Derived Attributes



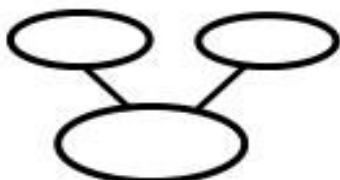
Represents Total Participation of Entity



Represents Weak Entity



Represents Weak Relationships



Represents Composite Attributes



Represents Key Attributes / Single Valued
Attributes

ER DIAGRAM



ER to Tables

HOTEL

hotel_id	Address	star	name	Email

STAFF

hotel_id	Staff_id	First_name	Last_name	Position	Email	Phone_no	salary	age

ROOMS

Room_no	Hotel_id	Price	Status	Type_id

ROOM TYPES

Type_id	Capacity	Price/night	Name

BOOKING

Booking_id	Hotel_id	Guest_id	Room_no	Check_in	Check_out	Status

PAYMENTS

Payment_id	Booking_id	Payment_method	Amount	Payment_date

GUEST

Guest_id	D.O.B	Name	Email	Address


```
CREATE TABLE BOOKING(  
    BOOKING_ID NUMBER PRIMARY KEY,  
    GUEST_ID NUMBER,  
    ROOM_NO NUMBER,  
    CHECKIN_DATE DATE,  
    CHECKOUT_DATE DATE,  
    HOTEL_ID NUMBER,  
    TOTAL_PRICE DECIMAL(10,2),  
    STATUS VARCHAR(20));
```

```
SQL> DESC BOOKING;
```

Name	Null?	Type
BOOKING_ID	NOT NULL	NUMBER
GUEST_ID		NUMBER
ROOM_NO		NUMBER
CHECKIN_DATE		DATE
CHECKOUT_DATE		DATE
HOTEL_ID		NUMBER
TOTAL_PRICE		NUMBER(10,2)
STATUS		VARCHAR2(20)

```
-----  
  
CREATE TABLE PAYMENT(  
    PAYMENT_ID NUMBER PRIMARY KEY,  
    BOOKING_ID NUMBER,  
    AMOUNT DECIMAL(10,2),  
    PAYMENT_DATE DATE,  
    PAYMENT_METHOD VARCHAR(50));
```

```
SQL> DESC PAYMENT;
```

Name	Null?	Type
PAYMENT_ID	NOT NULL	NUMBER
BOOKING_ID		NUMBER
AMOUNT		NUMBER(10,2)
PAYMENT_DATE		DATE
PAYMENT_METHOD		VARCHAR2(50)

```
CREATE TABLE GUEST_DETAILS(  
  GUEST_ID NUMBER PRIMARY KEY,  
  FIRST_NAME VARCHAR(50),  
  LAST_NAME VARCHAR(50),  
  DOB DATE,  
  ADDRESS VARCHAR(255),  
  PHONE VARCHAR(10),  
  EMAIL VARCHAR(100));
```

```
SQL> DESC GUEST_DETAILS;
```

Name	Null?	Type
GUEST_ID	NOT NULL	NUMBER
FIRST_NAME		VARCHAR2(50)
LAST_NAME		VARCHAR2(50)
DOB		DATE
ADDRESS		VARCHAR2(255)
PHONE		VARCHAR2(10)
EMAIL		VARCHAR2(100)

```
-----  
  
CREATE TABLE HOTEL(  
  HOTEL_ID NUMBER PRIMARY KEY,  
  NAME VARCHAR(50),  
  ADDRESS VARCHAR(50),  
  PHONE NUMBER(10),  
  EMAIL VARCHAR(30),  
  STARS NUMBER(7)  
);
```

```
SQL> DESC HOTEL;
```

Name	Null?	Type
HOTEL_ID	NOT NULL	NUMBER
NAME		VARCHAR2(50)
ADDRESS		VARCHAR2(50)
PHONE		NUMBER(10)
EMAIL		VARCHAR2(30)
STARS		NUMBER(7)

```
CREATE TABLE STAFF_DETAILS (  
    STAFF_ID NUMBER(20) PRIMARY KEY,  
    EMAIL VARCHAR(255),  
    HIREDATE DATE,  
    FIRST_NAME VARCHAR(50),  
    LAST_NAME VARCHAR(50),  
    PHONE VARCHAR(10),  
    POSITION VARCHAR(50),  
    DOB DATE,  
    SALARY DECIMAL(10,2),  
    HOTEL_ID NUMBER(20)  
);
```

```
SQL> DESC STAFF_DETAILS;
```

Name	Null?	Type
STAFF_ID	NOT NULL	NUMBER(20)
EMAIL		VARCHAR2(255)
HIREDATE		DATE
FIRST_NAME		VARCHAR2(50)
LAST_NAME		VARCHAR2(50)
PHONE		NUMBER
POSITION		VARCHAR2(50)
DOB		DATE
SALARY		NUMBER(10,2)
HOTEL_ID		NUMBER(20)

```
CREATE TABLE ROOMS (  
    ROOM_NO NUMBER,  
    HOTEL_ID NUMBER,
```

```

STATUS VARCHAR(30),
TYPE_ID NUMBER(10),
PRIMARY KEY(ROOM_NO,HOTEL_ID)
);

```

```

SQL> DESC ROOMS;

```

Name	Null?	Type
ROOM_NO	NOT NULL	NUMBER
HOTEL_ID	NOT NULL	NUMBER
STATUS		VARCHAR2(30)
PRICE		NUMBER(20)
TYPE_ID		NUMBER(10)

```

CREATE TABLE ROOM_TYPE (
    TYPE_ID NUMBER PRIMARY KEY,
    NAME VARCHAR2(50),
    PRICE_PER_NIGHT NUMBER,
    CAPACITY NUMBER );

```

```

SQL> DESC ROOM_TYPE;

```

Name	Null?	Type
TYPE_ID	NOT NULL	NUMBER
NAME		VARCHAR2(50)
PRICE_PER_NIGHT		NUMBER
CAPACITY		NUMBER

```

ALTER TABLE STAFF_DETAILS
ADD CONSTRAINT PHONE_CONST
CHECK (LENGTH(PHONE) =10);

```

```

ALTER TABLE STAFF_DETAILS
ADD CONSTRAINT CHK_EMAIL_STAFF
CHECK (EMAIL LIKE '%_@__%.__%');

```

```
ALTER TABLE STAFF_DETAILS
ADD CONSTRAINT FK_STAFF
FOREIGN KEY (HOTEL_ID)
REFERENCES HOTEL(HOTEL_ID);

ALTER TABLE GUEST_DETAILS
ADD CONSTRAINT CHK_EMAIL_GUEST
CHECK (EMAIL LIKE '%_@__%.__%');
```

```
ALTER TABLE GUEST_DETAILS
ADD CONSTRAINT PHONE_CONSTRT
CHECK (LENGTH(PHONE) =10);
```

```
ALTER TABLE ROOMS
ADD CONSTRAINT FK_HOTEL_ID
FOREIGN KEY (HOTEL_ID)
REFERENCES HOTEL(HOTEL_ID);
```

```
ALTER TABLE ROOMS
ADD CONSTRAINT FK_ROOMS
FOREIGN KEY (TYPE_ID)
REFERENCES ROOM_TYPE(TYPE_ID);
```

```
ALTER TABLE BOOKING
ADD CONSTRAINT FK_GUEST_MONEY
FOREIGN KEY (GUEST_ID)
REFERENCES GUEST_DETAILS(GUEST_ID);
```

```
ALTER TABLE BOOKING
```

```
ADD CONSTRAINT FK_ROOM
```

```
FOREIGN KEY(ROOM_NO,HOTEL_ID)
```

```
REFERENCES ROOMS(ROOM_NO,HOTEL_ID);
```

```
ALTER TABLE BOOKING
```

```
ADD CONSTRAINT CHECK_TOTAL_PRICE
```

```
CHECK (TOTAL_PRICE > 0);
```

```
ALTER TABLE BOOKING
```

```
ADD CONSTRAINT CHECK_STATUS
```

```
CHECK (UPPER(STATUS) IN ('ACTIVE', 'CANCELLED', 'COMPLETED'));
```

```
ALTER TABLE ROOMS
```

```
ADD CONSTRAINT CHECK_STATUS_ROOMS
```

```
CHECK (UPPER(STATUS) IN ('AVAILABLE', 'BOOKED'));
```

--BOOKING TABLE CONSTRAINTS

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
SYS_C00157192487	P	-
CHECK_STATUS	C	UPPER(STATUS) IN ('ACTIVE', 'CANCELLED', 'COMPLETED')
FK_ROOM	R	-
CHECK_TOTAL_PRICE	C	TOTAL_PRICE > 0
FK_GUEST_MONEY	R	-

--PAYMENT TABLE CONSTRAINTS

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
SYS_C00157192488	P	-

--GUEST_DETAILS TABLE CONSTRAINTS

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
SYS_C00157192489	P	-
CHK_EMAIL_GUEST	C	EMAIL LIKE '%_@_%.__%'
PHONE_CONSTRT	C	LENGTH(PHONE) =10

--HOTEL

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
SYS_C00157192490	P	-

--STAFF_DETAILS

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
SYS_C00157192491	P	-
PHONE_CONST	C	LENGTH(PHONE) =10
CHK_EMAIL_STAFF	C	EMAIL LIKE '%_@_%.__%'
FK_STAFF	R	-

Procedures

Procedure to create Insert into rooms

```
CREATE OR REPLACE PROCEDURE Insert_Into_Rooms (  
    RoomNo IN NUMBER,  
    HotelID IN NUMBER,  
    Status IN VARCHAR2,  
   TypeID IN NUMBER  
)  
AS  
BEGIN  
    INSERT INTO ROOMS (ROOM_NO, HOTEL_ID, STATUS, TYPE_ID)  
    VALUES (RoomNo, HotelID, Status, TypeID);  
  
    COMMIT;  
EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
        DBMS_OUTPUT.PUT_LINE('Error: ROOM ALREADY EXIST');  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('Error: An unexpected error occurred. Please contact the administrator.');
```

END;

```
BEGIN
  Insert_Into_Rooms(101, 1, 'AVAILABLE', 1);
  Insert_Into_Rooms(102, 1, 'AVAILABLE', 1);
  Insert_Into_Rooms(103, 1, 'AVAILABLE', 2);
  Insert_Into_Rooms(104, 1, 'AVAILABLE', 2);
  Insert_Into_Rooms(105, 1, 'AVAILABLE', 3);
  Insert_Into_Rooms(106, 1, 'AVAILABLE', 3);
  Insert_Into_Rooms(107, 1, 'AVAILABLE', 4);
  Insert_Into_Rooms(108, 1, 'AVAILABLE', 4);
  Insert_Into_Rooms(109, 1, 'AVAILABLE', 5);
  Insert_Into_Rooms(110, 1, 'AVAILABLE', 5);
  Insert_Into_Rooms(111, 2, 'AVAILABLE', 1);
  Insert_Into_Rooms(112, 2, 'AVAILABLE', 1);
  Insert_Into_Rooms(113, 2, 'AVAILABLE', 2);
  Insert_Into_Rooms(114, 2, 'AVAILABLE', 2);
  Insert_Into_Rooms(115, 2, 'AVAILABLE', 3);
  Insert_Into_Rooms(116, 2, 'AVAILABLE', 3);
  Insert_Into_Rooms(117, 2, 'AVAILABLE', 4);
  Insert_Into_Rooms(118, 2, 'AVAILABLE', 4);
  Insert_Into_Rooms(119, 2, 'AVAILABLE', 5);
  Insert_Into_Rooms(120, 2, 'AVAILABLE', 5);
  Insert_Into_Rooms(121, 3, 'AVAILABLE', 1);
  Insert_Into_Rooms(122, 3, 'AVAILABLE', 1);
  Insert_Into_Rooms(123, 3, 'AVAILABLE', 2);
  Insert_Into_Rooms(124, 3, 'AVAILABLE', 2);
  Insert_Into_Rooms(125, 3, 'AVAILABLE', 3);
END;
```

Procedure to Insert into Hotel

```

CREATE OR REPLACE PROCEDURE Insert_Into_Hotel (
    HotelID IN NUMBER,
    Name IN VARCHAR2,
    Address IN VARCHAR2,
    Phone IN NUMBER,
    Email IN VARCHAR2,
    Stars IN NUMBER
)
AS
BEGIN
    INSERT INTO HOTEL (HOTEL_ID, NAME, ADDRESS, PHONE, EMAIL, STARS)
    VALUES (HotelID, Name, Address, Phone, Email, Stars);
    COMMIT;
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Error: Unique constraint violated.');
```

```

    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Error: Invalid value for numeric or date variable.');
```

```

    WHEN OTHERS THEN
        IF SQLCODE = -2291 THEN
            DBMS_OUTPUT.PUT_LINE('Error: Foreign key constraint violated.');
```

```

        ELSE
            DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        END IF;
END;

END;
```

```

BEGIN
    Insert_Into_Hotel(1, 'Grand View Hotel', '123 Ocean Blvd', 1234567890, 'grandview@example.com', 4);
    Insert_Into_Hotel(2, 'Cozy Corner Inn', '456 Forest Ave', 9876543210, 'cozycorner@example.com', 5);
    Insert_Into_Hotel(3, 'Lakeside Retreat', '789 Lakeview Dr', 1357924680, 'lakeside@example.com', 3);
    Insert_Into_Hotel(4, 'Mountaintop Lodge', '101 Summit Rd', 2468135790, 'mountaintop@example.com', 4);
    Insert_Into_Hotel(5, 'Seaside Escape', '202 Beachfront Blvd', 3698521470, 'seaside@example.com', 5);
END;
```

Procedure to Insert Staff Details

```

CREATE OR REPLACE PROCEDURE Insert_Staff_Details (
    StaffID IN NUMBER,
    Email IN VARCHAR2,
    HireDate IN DATE,
    FirstName IN VARCHAR2,
    LastName IN VARCHAR2,
    phone IN VARCHAR,
    Position IN VARCHAR2,
    DOB IN DATE,
    Salary IN DECIMAL,
    HotelID IN NUMBER
)
AS
BEGIN
    INSERT INTO STAFF_DETAILS (STAFF_ID, EMAIL, HIREDATE, FIRST_NAME, LAST_NAME,PHONE,POSITION, DOB, SALARY, HOTEL_ID)
    VALUES (StaffID, Email, HireDate, FirstName, LastName,phone, Position, DOB, Salary, HotelID);

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Staff details inserted successfully.');
```

```

EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Error: Unique constraint violated.');
```

```

    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Error: Invalid value for numeric or date variable.');
```

```

    WHEN OTHERS THEN
        IF SQLCODE = -2291 THEN
            DBMS_OUTPUT.PUT_LINE('Error: Foreign key constraint violated.');
```

```

        ELSE
            DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        END IF;
END;
```

```

BEGIN
    Insert_Staff_Details(1, 'staff1@example.com', TO_DATE('2022-01-10', 'YYYY-MM-DD'), 'David', 'Brown', '1234567890', 'Manager', TO_DATE('1980-03-25', 'YYYY-MM-DD'), 50000, 1);
    Insert_Staff_Details(2, 'staff2@example.com', TO_DATE('2022-02-15', 'YYYY-MM-DD'), 'Sarah', 'Wilson', '9876543210', 'Receptionist', TO_DATE('1985-06-18', 'YYYY-MM-DD'), 35000, 2);
    Insert_Staff_Details(3, 'staff3@example.com', TO_DATE('2022-03-20', 'YYYY-MM-DD'), 'Mark', 'Taylor', '2468013579', 'Housekeeping', TO_DATE('1990-09-12', 'YYYY-MM-DD'), 30000, 3);
    Insert_Staff_Details(4, 'staff4@example.com', TO_DATE('2022-04-25', 'YYYY-MM-DD'), 'Emily', 'Anderson', '1592638470', 'Chef', TO_DATE('1987-04-05', 'YYYY-MM-DD'), 45000, 4);
    Insert_Staff_Details(5, 'staff5@example.com', TO_DATE('2022-05-30', 'YYYY-MM-DD'), 'Michael', 'Hernandez', '2468097531', 'Housekeeping', TO_DATE('1993-11-25', 'YYYY-MM-DD'), 30000, 5);
    Insert_Staff_Details(6, 'staff6@example.com', TO_DATE('2022-06-05', 'YYYY-MM-DD'), 'Emma', 'Garcia', '7894561230', 'Chef', TO_DATE('1983-11-05', 'YYYY-MM-DD'), 45000, 5);
    Insert_Staff_Details(7, 'staff7@example.com', TO_DATE('2022-07-10', 'YYYY-MM-DD'), 'Daniel', 'Martinez', '2468024680', 'Receptionist', TO_DATE('1995-02-18', 'YYYY-MM-DD'), 35000, 1);
    Insert_Staff_Details(8, 'staff8@example.com', TO_DATE('2022-08-15', 'YYYY-MM-DD'), 'Olivia', 'Brown', '3579135790', 'Housekeeping', TO_DATE('1991-06-14', 'YYYY-MM-DD'), 30000, 2);
    Insert_Staff_Details(9, 'staff9@example.com', TO_DATE('2022-09-20', 'YYYY-MM-DD'), 'William', 'Thomas', '4680246802', 'Manager', TO_DATE('1988-11-22', 'YYYY-MM-DD'), 50000, 3);
    Insert_Staff_Details(10, 'staff10@example.com', TO_DATE('2022-10-25', 'YYYY-MM-DD'), 'Samantha', 'Johnson', '5791357913', 'Receptionist', TO_DATE('1980-07-30', 'YYYY-MM-DD'), 35000, 4);
    Insert_Staff_Details(11, 'staff11@example.com', TO_DATE('2022-11-30', 'YYYY-MM-DD'), 'James', 'Taylor', '6802468024', 'Housekeeping', TO_DATE('1975-04-18', 'YYYY-MM-DD'), 30000, 5);
    Insert_Staff_Details(12, 'staff12@example.com', TO_DATE('2022-12-05', 'YYYY-MM-DD'), 'Isabella', 'Hernandez', '7913579135', 'Chef', TO_DATE('1984-02-09', 'YYYY-MM-DD'), 45000, 1);
    Insert_Staff_Details(13, 'staff13@example.com', TO_DATE('2023-01-10', 'YYYY-MM-DD'), 'Benjamin', 'Moore', '8024680246', 'Manager', TO_DATE('1987-09-27', 'YYYY-MM-DD'), 50000, 2);
    Insert_Staff_Details(14, 'staff14@example.com', TO_DATE('2023-02-15', 'YYYY-MM-DD'), 'Ava', 'Perez', '9135791357', 'Receptionist', TO_DATE('1983-12-05', 'YYYY-MM-DD'), 35000, 3);
    Insert_Staff_Details(15, 'staff15@example.com', TO_DATE('2023-03-20', 'YYYY-MM-DD'), 'Jacob', 'Gonzalez', '0246802468', 'Housekeeping', TO_DATE('1992-03-19', 'YYYY-MM-DD'), 30000, 4);
    Insert_Staff_Details(16, 'staff16@example.com', TO_DATE('2023-04-25', 'YYYY-MM-DD'), 'Mia', 'Walker', '1357913579', 'Chef', TO_DATE('1979-08-11', 'YYYY-MM-DD'), 45000, 5);
    Insert_Staff_Details(17, 'staff17@example.com', TO_DATE('2023-05-30', 'YYYY-MM-DD'), 'Ethan', 'Roberts', '2468024680', 'Manager', TO_DATE('1986-05-03', 'YYYY-MM-DD'), 50000, 1);
    Insert_Staff_Details(18, 'staff18@example.com', TO_DATE('2023-06-05', 'YYYY-MM-DD'), 'Charlotte', 'Clark', '3579135791', 'Receptionist', TO_DATE('1994-11-26', 'YYYY-MM-DD'), 35000, 2);
    Insert_Staff_Details(19, 'staff19@example.com', TO_DATE('2023-07-10', 'YYYY-MM-DD'), 'Alexander', 'Lewis', '4680246802', 'Housekeeping', TO_DATE('1981-01-17', 'YYYY-MM-DD'), 30000, 3);
    Insert_Staff_Details(20, 'staff20@example.com', TO_DATE('2023-08-15', 'YYYY-MM-DD'), 'Harper', 'Lee', '5791357913', 'Chef', TO_DATE('1990-07-09', 'YYYY-MM-DD'), 45000, 4);
    Insert_Staff_Details(21, 'staff21@example.com', TO_DATE('2023-09-20', 'YYYY-MM-DD'), 'William', 'Scott', '6802468024', 'Manager', TO_DATE('1976-04-30', 'YYYY-MM-DD'), 50000, 5);
    Insert_Staff_Details(22, 'staff22@example.com', TO_DATE('2023-10-25', 'YYYY-MM-DD'), 'Evelyn', 'King', '7913579135', 'Receptionist', TO_DATE('1985-02-13', 'YYYY-MM-DD'), 35000, 1);
    Insert_Staff_Details(23, 'staff23@example.com', TO_DATE('2023-11-30', 'YYYY-MM-DD'), 'Michael', 'Young', '8024680246', 'Housekeeping', TO_DATE('1993-10-24', 'YYYY-MM-DD'), 30000, 2);
    Insert_Staff_Details(24, 'staff24@example.com', TO_DATE('2023-12-05', 'YYYY-MM-DD'), 'Liam', 'Allen', '9135791357', 'Chef', TO_DATE('1988-07-07', 'YYYY-MM-DD'), 45000, 3);
    Insert_Staff_Details(25, 'staff25@example.com', TO_DATE('2024-01-10', 'YYYY-MM-DD'), 'Avery', 'Hill', '0246802468', 'Manager', TO_DATE('1982-12-19', 'YYYY-MM-DD'), 50000, 4);
END;
```

Procedure to Insert Guest Details

```

CREATE OR REPLACE PROCEDURE Insert_Guest_Details
(
    GuestID IN NUMBER,
    FirstName IN VARCHAR2,
    LastName IN VARCHAR2,
    DOB IN DATE,
    Address IN VARCHAR2,
    Phone IN VARCHAR2,
    Email IN VARCHAR2
)
AS
BEGIN
    INSERT INTO GUEST_DETAILS (GUEST_ID, FIRST_NAME, LAST_NAME, DOB, ADDRESS, PHONE, EMAIL)
    VALUES (GuestID, FirstName, LastName, DOB, Address, Phone, Email);
    COMMIT;
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Error: Duplicate GuestID found. Please choose a different ID.');
```

```

    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: An unexpected error occurred. Please contact the administrator.');
```

```

END;

BEGIN
    Insert_Guest_Details(1, 'John', 'Doe', TO_DATE('1990-05-15', 'YYYY-MM-DD'), '1234 Elm St', '1234567890', 'john@example.com');
    Insert_Guest_Details(2, 'Jane', 'Smith', TO_DATE('1985-08-20', 'YYYY-MM-DD'), '5678 Oak St', '9876543210', 'jane@example.com');
    Insert_Guest_Details(3, 'Michael', 'Johnson', TO_DATE('1978-12-10', 'YYYY-MM-DD'), '9101 Pine St', '2468013579', 'michael@example.com');
    Insert_Guest_Details(4, 'Emily', 'Brown', TO_DATE('1995-03-28', 'YYYY-MM-DD'), '2345 Maple St', '1357924680', 'emily@example.com');
    Insert_Guest_Details(5, 'Daniel', 'Wilson', TO_DATE('1982-09-03', 'YYYY-MM-DD'), '6789 Birch St', '3698521470', 'daniel@example.com');
    Insert_Guest_Details(6, 'Olivia', 'Martinez', TO_DATE('1991-06-14', 'YYYY-MM-DD'), '7890 Cedar St', '6543219870', 'olivia@example.com');
    Insert_Guest_Details(7, 'William', 'Anderson', TO_DATE('1988-11-22', 'YYYY-MM-DD'), '3456 Pine St', '1592638470', 'william@example.com');
    Insert_Guest_Details(8, 'Samantha', 'Thomas', TO_DATE('1980-07-30', 'YYYY-MM-DD'), '4567 Elm St', '8765432190', 'samantha@example.com');
    Insert_Guest_Details(9, 'James', 'Taylor', TO_DATE('1975-04-18', 'YYYY-MM-DD'), '7891 Oak St', '2468097531', 'james@example.com');
    Insert_Guest_Details(10, 'Isabella', 'Hernandez', TO_DATE('1984-02-09', 'YYYY-MM-DD'), '6789 Maple St', '1357924680', 'isabella@example.com');
    Insert_Guest_Details(11, 'Benjamin', 'Moore', TO_DATE('1987-09-27', 'YYYY-MM-DD'), '2345 Cedar St', '9876543210', 'benjamin@example.com');
    Insert_Guest_Details(12, 'Ava', 'Perez', TO_DATE('1983-12-05', 'YYYY-MM-DD'), '4567 Birch St', '6543219870', 'ava@example.com');
    Insert_Guest_Details(13, 'Jacob', 'Gonzalez', TO_DATE('1992-03-19', 'YYYY-MM-DD'), '3456 Oak St', '1592638470', 'jacob@example.com');
    Insert_Guest_Details(14, 'Mia', 'Walker', TO_DATE('1979-08-11', 'YYYY-MM-DD'), '1234 Pine St', '8765432190', 'mia@example.com');
    Insert_Guest_Details(15, 'Ethan', 'Roberts', TO_DATE('1986-05-03', 'YYYY-MM-DD'), '7890 Elm St', '2468097531', 'ethan@example.com');
    Insert_Guest_Details(16, 'Charlotte', 'Clark', TO_DATE('1994-11-26', 'YYYY-MM-DD'), '2345 Oak St', '1357924680', 'charlotte@example.com');
    Insert_Guest_Details(17, 'Alexander', 'Lewis', TO_DATE('1991-01-17', 'YYYY-MM-DD'), '4567 Cedar St', '9876543210', 'alexander@example.com');
    Insert_Guest_Details(18, 'Harper', 'Lee', TO_DATE('1990-07-09', 'YYYY-MM-DD'), '3456 Birch St', '6543219870', 'harper@example.com');
    Insert_Guest_Details(19, 'William', 'Scott', TO_DATE('1976-04-30', 'YYYY-MM-DD'), '1234 Oak St', '1592638470', 'william2@example.com');
    Insert_Guest_Details(20, 'Evelyn', 'King', TO_DATE('1985-02-13', 'YYYY-MM-DD'), '6789 Cedar St', '8765432190', 'evelyn@example.com');
    Insert_Guest_Details(21, 'Michael', 'Young', TO_DATE('1993-10-24', 'YYYY-MM-DD'), '7890 Birch St', '2468097531', 'michael2@example.com');
    Insert_Guest_Details(22, 'Liam', 'Allen', TO_DATE('1988-07-07', 'YYYY-MM-DD'), '2345 Elm St', '1357924680', 'liam@example.com');
    Insert_Guest_Details(23, 'Avery', 'Hill', TO_DATE('1982-12-19', 'YYYY-MM-DD'), '4567 Pine St', '9876543210', 'avery@example.com');
    Insert_Guest_Details(24, 'Emma', 'Baker', TO_DATE('1991-06-03', 'YYYY-MM-DD'), '7891 Elm St', '6543219870', 'emma@example.com');
    Insert_Guest_Details(25, 'Noah', 'Green', TO_DATE('1977-09-15', 'YYYY-MM-DD'), '3456 Cedar St', '1592638470', 'noah@example.com');
```

```

END;

```

Procedure to Insert into Room Types

```

CREATE OR REPLACE PROCEDURE Insert_Into_RoomType (
   TypeID IN NUMBER,
    Name IN VARCHAR2,
    PricePerNight IN NUMBER,
    Capacity IN NUMBER
)
AS
BEGIN
    INSERT INTO ROOM_TYPE (TYPE_ID, NAME, PRICE_PER_NIGHT, CAPACITY)
    VALUES (TypeID, Name, PricePerNight, Capacity);

    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Error: TypeID does not exist.');
```

```

    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: An unexpected error occurred. Please contact the administrator.');
```

```

END;
```

```

BEGIN
    Insert_Into_RoomType(1, 'Single', 1100, 1);
    Insert_Into_RoomType(2, 'Double', 1800, 2);
    Insert_Into_RoomType(3, 'Suite', 2750, 4);
    Insert_Into_RoomType(4, 'Penthouse', 5500, 6);
    Insert_Into_RoomType(5, 'Family', 2200, 3);
END;
```

Procedure to Update Staff Details

```

CREATE OR REPLACE PROCEDURE UPDATE_STAFF_DETAILS(
    STAFF_ID_PARAM IN NUMBER,
    OPTION_PARAM IN NUMBER,
    UPDATED_ATTRIBUTE VARCHAR2
)
AS
    CNT NUMBER := 0;
BEGIN
    SELECT COUNT(*)
    INTO CNT
    FROM STAFF_DETAILS
    WHERE STAFF_ID = STAFF_ID_PARAM;

    IF CNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('ENTER A VALID STAFF ID');
    ELSE
        IF OPTION_PARAM = 1 THEN
            UPDATE STAFF_DETAILS
            SET PHONE = UPDATED_ATTRIBUTE
            WHERE STAFF_ID = STAFF_ID_PARAM;
        ELSIF OPTION_PARAM = 2 THEN
            UPDATE STAFF_DETAILS
            SET EMAIL = UPDATED_ATTRIBUTE
            WHERE STAFF_ID = STAFF_ID_PARAM;
        ELSIF OPTION_PARAM = 3 THEN
            UPDATE STAFF_DETAILS
            SET POSITION = UPDATED_ATTRIBUTE
            WHERE STAFF_ID = STAFF_ID_PARAM;
        ELSE
            DBMS_OUTPUT.PUT_LINE('WRONG OPTION!');
        END IF;
        COMMIT;
    END IF;
END;
/

```

Procedure to Update Guest Details

```

CREATE OR REPLACE PROCEDURE UPDATE_GUEST_DETAILS(
    GUEST_ID_PARAM IN NUMBER,
    OPTION_PARAM IN NUMBER,
    UPDATED_ATTRIBUTE VARCHAR2
)
AS
    CNT NUMBER := 0;
BEGIN
    SELECT COUNT(*)
    INTO CNT
    FROM GUEST_DETAILS
    WHERE GUEST_ID = GUEST_ID_PARAM;

    IF CNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('ENTER A VALID GUEST ID');
    ELSE
        IF OPTION_PARAM = 1 THEN
            UPDATE GUEST_DETAILS
            SET PHONE = UPDATED_ATTRIBUTE
            WHERE GUEST_ID = GUEST_ID_PARAM;
        ELSIF OPTION_PARAM = 2 THEN
            UPDATE GUEST_DETAILS
            SET EMAIL = UPDATED_ATTRIBUTE
            WHERE GUEST_ID = GUEST_ID_PARAM;
        ELSIF OPTION_PARAM = 3 THEN
            UPDATE GUEST_DETAILS
            SET ADDRESS = UPDATED_ATTRIBUTE
            WHERE GUEST_ID = GUEST_ID_PARAM;
        ELSE
            DBMS_OUTPUT.PUT_LINE('WRONG OPTION!');
        END IF;
        COMMIT;
    END IF;
END;

```

Procedure to Book Room


```

SQL> CREATE OR REPLACE PROCEDURE BOOK_ROOM(
2   CHECKIN_DATE_PARAM IN DATE,
3   CHECKOUT_DATE_PARAM IN DATE,
4   ROOM_NUMBER_PARAM IN NUMBER,
5   HOTEL_ID_PARAM IN NUMBER,
6   GUEST_ID_PARAM IN NUMBER
7 )
8 IS
9     COUNT_ROOM NUMBER := 0;
10    NO_OF_DAYS NUMBER;
11    ROOMTYPE NUMBER;
12    TOTAL_PRICE NUMBER;
13 BEGIN
14     -- Check room availability for the given check-in and check-out dates
15     SELECT COUNT(*)
16     INTO COUNT_ROOM
17     FROM ROOMS R
18     LEFT JOIN BOOKING B ON R.ROOM_NO = B.ROOM_NO
19     WHERE R.ROOM_NO = ROOM_NUMBER_PARAM
20     AND R.HOTEL_ID = HOTEL_ID_PARAM
21     AND R.STATUS = 'AVAILABLE'
22     AND (B.CHECKIN_DATE IS NULL OR B.CHECKOUT_DATE <= CHECKIN_DATE_PARAM OR B.CHECKIN_DATE >= CHECKOUT_DATE_PARAM);
23
24     IF COUNT_ROOM = 0 THEN
25         DBMS_OUTPUT.PUT_LINE('ERROR: Room is not available for the selected dates.');

Procedure created.


```

```

602 v begin
603     book_room(TO_DATE('2024-01-28','YYYY-MM-DD'), TO_DATE('2024-01-30','YYYY-MM-DD'), 101, 1, 4);
604     book_room(TO_DATE('2024-02-10','YYYY-MM-DD'), TO_DATE('2024-02-15','YYYY-MM-DD'), 102, 1, 5);
605     book_room(TO_DATE('2024-03-05','YYYY-MM-DD'), TO_DATE('2024-03-08','YYYY-MM-DD'), 103, 1, 6);
606     book_room(TO_DATE('2024-04-20','YYYY-MM-DD'), TO_DATE('2024-04-25','YYYY-MM-DD'), 111, 2, 7);
607     book_room(TO_DATE('2024-05-15','YYYY-MM-DD'), TO_DATE('2024-05-20','YYYY-MM-DD'), 105, 1, 8);
608 end;
609
610 select* from booking;
611

```

Statement processed.
Room booked successfully.
Room booked successfully.
Room booked successfully.
Room booked successfully.
Room booked successfully.

BOOKING_ID	GUEST_ID	ROOM_NO	CHECKIN_DATE	CHECKOUT_DATE	HOTEL_ID	TOTAL_PRICE	STATUS
1	4	101	28-JAN-24	30-JAN-24	1	2200	COMPLETED
2	5	102	10-FEB-24	15-FEB-24	1	5500	COMPLETED
3	6	103	05-MAR-24	08-MAR-24	1	5400	COMPLETED
4	7	111	20-APR-24	25-APR-24	2	5500	COMPLETED
5	8	105	15-MAY-24	20-MAY-24	1	13750	COMPLETED

Procedure To Cancel Room

```

SQL> CREATE OR REPLACE PROCEDURE CANCEL_ROOM(
2     BOOKING_ID_PARAM IN NUMBER
3 )
4 AS
5     ROOM_NUMBER NUMBER;
6     HOTEL_ID NUMBER;
7     COUNT_BOOKING NUMBER := 0;
8 BEGIN
9     -- Check if the booking ID exists
10    SELECT COUNT(*)
11    INTO COUNT_BOOKING
12    FROM BOOKING
13    WHERE BOOKING_ID = BOOKING_ID_PARAM;
14
15    IF COUNT_BOOKING = 0 THEN
16        DBMS_OUTPUT.PUT_LINE('ERROR: Booking ID does not exist.');

```

Procedure created.

```

612 v begin
613     cancel_room(1);
614 end;
615

```

Statement processed.
Room booking canceled successfully.

BOOKING_ID	GUEST_ID	ROOM_NO	CHECKIN_DATE	CHECKOUT_DATE	HOTEL_ID	TOTAL_PRICE	STATUS
1	4	101	28-JAN-24	30-JAN-24	1	2200	CANCELLED
2	5	102	10-FEB-24	15-FEB-24	1	5500	COMPLETED
3	6	103	05-MAR-24	08-MAR-24	1	5400	COMPLETED
4	7	111	20-APR-24	25-APR-24	2	5500	COMPLETED
5	8	105	15-MAY-24	20-MAY-24	1	13750	COMPLETED

Procedure to Check Available Rooms

```
Connected.
SQL>
SQL> CREATE OR REPLACE PROCEDURE CHECK_AVAILABLE_ROOMS
 2 AS
 3 BEGIN
 4     DBMS_OUTPUT.PUT_LINE('Available rooms:');
 5     FOR room_rec IN (
 6         SELECT ROOM_NO, HOTEL_ID
 7         FROM ROOMS
 8         WHERE STATUS = 'AVAILABLE'
 9     ) LOOP
10         DBMS_OUTPUT.PUT_LINE('Room Number: ' || room_rec.ROOM_NO || ', Hotel ID: ' || room_rec.HOTEL_ID);
11     END LOOP;
12 EXCEPTION
13     WHEN OTHERS THEN
14         DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
15 END;
16 /

Procedure created.
```

Thank You!
