**Note: "**Students will not be provided any database, you have to create the database as mentioned in the following project and work on the given questions**".**

**Create a database of the hotel named “Grosvenor”. Given are a few hints on how tables should be created and how the values should be inserted. After the creation of the database, answer the questions given at the end about the hotel.**

**The Database**

Hotel (Hotel\_No, Name, Address) Room (Room\_No, Hotel\_No, Type, Price) Booking (Hotel\_No,

Guest\_No, Date\_From, Date\_To, Room\_No) Guest (Guest\_No, Name, Address)

**Creating the Tables(Make sure to include primary and foreign keys also keeping in mind the normalization of tables).**

CREATE TABLE hotel ( hotel\_no CHAR(4) NOT NULL, name VARCHAR(20) NOT NULL, address

VARCHAR(50) NOT NULL);

CREATE TABLE room ( room\_no VARCHAR(4) NOT NULL, hotel\_no CHAR(4) NOT NULL, type CHAR(1)

NOT NULL, price DECIMAL(5,2) NOT NULL);

CREATE TABLE booking (hotel\_no CHAR(4) NOT NULL, guest\_no CHAR(4) NOT NULL, date\_from

DATETIME NOT NULL, date\_to DATETIME NULL, room\_no CHAR(4) NOT NULL); Dates: YYYY-MM-DD;

CREATE TABLE guest ( guest\_no CHAR(4) NOT NULL, name VARCHAR(20) NOT NULL, address

VARCHAR(50) NOT NULL);

**Populating the Tables(It is not necessary to insert only one row in each column, given here is just an example).**

INSERT INTO hotel VALUES ('H111', 'Grosvenor Hotel‘, 'London'); INSERT INTO room VALUES ('1', 'H111', 'S', 72.00); INSERT INTO guest VALUES ('G111', 'John Smith', 'London'); INSERT INTO booking VALUES ('H111', 'G111', DATE'1999-01-01', DATE'1999-01-02', '1');

**Updating the Tables**

UPDATE room SET price = price\*1.05;

* Create a separate table with the same structure as the Booking table to hold archive records.
* Using the INSERT statement, copy the records from the Booking table to the archive table relating to bookings before 1st January 2000. Delete all bookings before 1st January 2000 from the Booking table.

CREATE TABLE booking\_old ( hotel\_no CHAR(4) NOT NULL, guest\_no CHAR(4) NOT NULL, date\_from DATETIME NOT NULL, date\_to DATETIME NULL, room\_no VARCHAR(4) NOT NULL);

INSERT INTO booking\_old (SELECT \* FROM booking WHERE date\_to < DATE‘2000-01-01'); DELETE FROM booking WHERE date\_to < DATE‘2000-01-01';

Queries: Back to the Database Hotel (Hotel\_No, Name, Address) Room (Room\_No, Hotel\_No, Type,

Price) Booking (Hotel\_No, Guest\_No, Date\_From, Date\_To, Room\_No) Guest (Guest\_No, Name,

Address)

**Simple Queries**

1. List full details of all hotels.

2. List full details of all hotels in London.

3. List the names and addresses of all guests in London, alphabetically ordered by name.

4. List all double or family rooms with a price below £40.00 per night, in ascending order of price.

5. List the bookings for which no date\_to has been specified.

**Aggregate Functions**

1. How many hotels are there?

2. What is the average price of a room?

3. What is the total revenue per night from all double rooms?

4. How many different guests have made bookings for August?

**Subqueries and Joins**

1. List the price and type of all rooms at the Grosvenor Hotel.

2. List all guests currently staying at the Grosvenor Hotel.

3. List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the

room, if the room is occupied.

4. What is the total income from bookings for the Grosvenor Hotel today?

5. List the rooms that are currently unoccupied at the Grosvenor Hotel.

6. What is the lost income from unoccupied rooms at the Grosvenor Hotel?

**Grouping**

1. List the number of rooms in each hotel.

2. List the number of rooms in each hotel in London.

3. What is the average number of bookings for each hotel in August?

4. What is the most commonly booked room type for each hotel in London?

5. What is the lost income from unoccupied rooms at each hotel today?