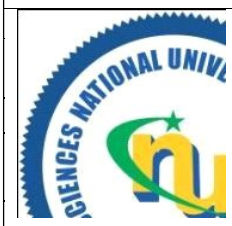
National University of Computer and Emerging Sciences, Lahore Campus				
	**Course Name:**	**Database Systems**	**Course Code:**	**CS2005**
			Semester:	**Spring 2024**
	Type		**Total Marks:**	**100**
	Assignment 01	do these queries with joing		
	Assignment 02	do q1 queries with subqueries	**DUE DATE:**	**24-Feb-24**
	Assignment 03	convert q1 into RA		

SUBMIT YOUR QUERIES IN SQL FILE.
PLAGIARISM WILL CAUSE ADVERSE CONSEQUENCES.

Note:- Each Query carries 3 mark

Schema: Café Management system
Customers: This table stores information about customers, including their name, date of birth, contact details, address, and email.

Staff: This table contains data about cafe staff members, including their name, date of birth, contact details, address, role (e.g., Chef, Waiter), and department they belong to.

Departments: This table represents different departments within the cafe, such as Kitchen, Service, and Management.

Orders: This table records orders made by customers with cafe staff. It includes the customer's ID, staff member's ID, order details (items ordered and quantity), order date and time, and status (e.g., Pending, Completed).

Transaction Records: This table contains transaction records for customer purchases, including the date of service, items purchased, total amount, and payment status (e.g., Paid, Unpaid).

Menu Items: This table stores information about menu items available in the cafe, including the item name, description, price, and category (e.g., Appetizers, Main Course, Desserts).

Ingredients: This table contains data related to ingredients used in preparing menu items, including the ingredient name, quantity on hand, and unit of measurement.

Suppliers: This table stores information about suppliers from whom the cafe purchases ingredients, including the supplier name, contact details, and address.

Tables: This table represents the tables available in the cafe for seating customers. It includes attributes such as table number, capacity, and status (e.g., Occupied, Available).

Scenario Usage:

In this Cafe Management System:

Customers can place orders with cafe staff.
Cafe staff members, including chefs and waiters, are associated with specific departments such as Kitchen and Service.
Transaction records are generated for each customer purchase to track payment status.
Menu items are available for ordering and are categorized for easy navigation.
Ingredients are managed to ensure availability for menu item preparation.
Suppliers provide ingredients necessary for preparing menu items.

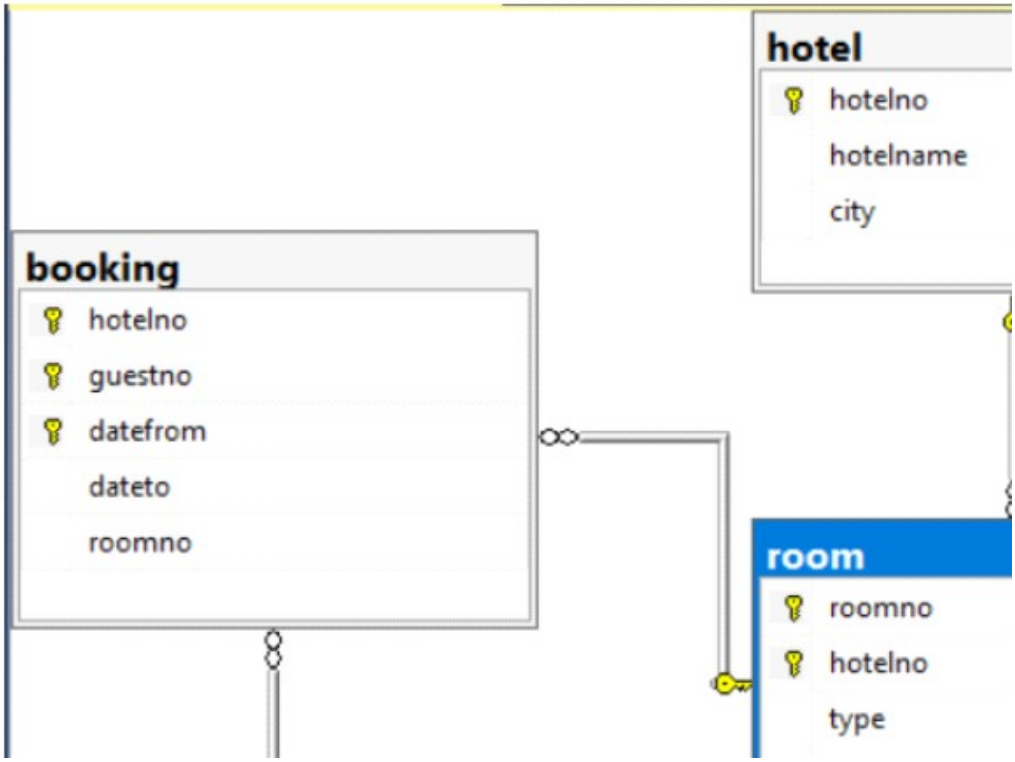
Create tables with proper datatypes and actions appropriate according to you.
Write sql queries for the following:

- Write a SQL query to retrieve the total number of orders placed by each customer, sorted in descending order of the number of orders.
- Can you construct a SQL query to calculate the total revenue generated by each menu category for the past month, including only categories that have generated revenue?
- Implement a SQL query to find the average price of menu items within each category, rounding the average price to two decimal places.
- Design a SQL query to identify the staff members who have served the highest total amount of transactions in the last quarter.
- Write a SQL query to retrieve the names of all menu items that contain the word "chocolate" in their description, disregarding case sensitivity.
- Construct a SQL query to find the top 3 most popular ingredients used across all menu items, based on the total quantity used.
- Implement a SQL query to calculate the average number of orders served per day for each staff member, considering only those staff members who have served at least 10 orders.
- Design a SQL query to identify any staff members who are assigned to multiple departments and list their names along with the departments they are assigned to.
- Develop a SQL query to find the total number of transactions made on weekends (Saturdays and Sundays) in the past three months.

- Construct a SQL query to calculate the total revenue generated by each staff member, considering both their own transactions and those they assisted with, sorted in descending order of revenue.
- Design a SQL query to find the average time taken to fulfill orders, excluding orders that were canceled or returned, rounded to the nearest minute.

- Develop a SQL query to list all customers who have placed orders for items from more than one category, along with the total number of categories they have ordered from.
- Write a SQL query to calculate the total cost of ingredients used for each menu item, considering the quantity of each ingredient used and its associated cost per unit.
- Construct a SQL query to find the average number of transactions per day during the busiest week of the year, considering all weeks within the last year .
- Design a SQL query to identify any staff members who have not served any orders in the past month, along with their respective roles and departments.

TOPIC: Relational Data Model
Q2. Consider the following database state: (total points =35)



Hotel :

Results Messages

	hotelno	hotelname	city
1	ch01	Omni Shoreham	Londor
2	ch02	Phoenix Park	Londor
3	dc01	Latham	Berlin
4	fb01	Grosvenor	Londor

Booking:

Results Messages

	hotelno	guestno	datefrom	dateto	roomno
1	ch01	10006	2021-04-04	NULL	1101
2	ch02	10002	2025-04-04	2006-04-05	801
3	dc01	10003	2020-04-05	NULL	1001
4	dc01	10007	2013-04-05	2015-04-05	1001
5	fb01	10001	2001-04-04	2008-04-04	501
6	fb01	10001	2001-04-05	NULL	701
7	fb01	10004	2015-04-04	2015-04-05	601
8	fb01	10005	2002-04-05	2007-04-05	501
9	fb02	10003	2005-04-04	2004-10-04	1001

Guest:

Results Messages

	guestno	guestname	guestaddress
1	10001	John Kay	56 High St, Lond
2	10002	Mike Ritchie	18 Tain St, Londo
3	10003	Mary Tregear	5 Tarbot Rd, Abe
4	10004	Joe Keogh	2 Fergus Dr, Abe
5	10005	Carol Farrel	6 Achray St, Glas
6	10006	Tina Murphy	63 Well St, Glasg

Room:

Results Messages

	roomno	hotelno	type	price
1	501	fb01	single	19.00
2	601	fb01	double	29.00
3	701	ch02	single	10.00
4	701	fb01	family	39.00
5	801	ch02	double	15.00
6	901	dc01	single	18.00
7	1001	ch01	single	29.99
8	1001	dc01	double	30.00
9	1001	fb02	single	58.00

Apply following operations on the above database. State if the operation would be carried out
successfully or not , give proper explaination with reasoning . In case of successful operation indicate the changes that will be made to the above.

database. Also state all the integrity constraints violated by each operation, if any. Note that all
following operations are independent.
FOREIGN KEY CONTRAINTS ARE FOLLOWING:
On update Cascade.
On delete set NULL.

INSERT:

Insert into hotel values('fb01', 'The Ritz', 'London');
Insert into room values(1201, 'fb01', 'single', 99.99);
Insert into booking values('fb01', 10001, '04-04-01', '04-04-08', 501);
Insert into hotel values('hw03', 'The Ritz', 'London');
Insert into booking values('hw03', 10008, '04-06-10', '04-06-15', 1201);

UPDATE:

Update guest Set guestaddress = '25 Maple Ave, London' where guestno = 10008;
Update room Set price = 109.99 where roomno = 1201 and hotelno = 'fb05';
Update guest Set guestaddress = '25 Maple Ave, London' where guestno = 10009;
Update booking Set roomno = 999 where hotelno = 'fb01' and guestno = 10001 and datefrom = '04-04-01';
Update room Set price = 109.99 where roomno = 1201 and hotelno = 'fb01';

DELETE:

Delete from room where hotelno = 'fb05';
Delete from guest where guestno = 10010;
Delete from room where roomno = 1301 and hotelno = 'dc01';
Delete from booking where hotelno = 'fb05';
Delete from booking where hotelno = 'fb01';
Delete from hotel where hotelno = 'fb02';

QUESTION NO 3 (TOTAL POINTS =20)

• Give detailed output for the following queries(also please show your working in the solution as well) and if any error then correct it.

Use the table of question 2 for solving the mentioned questions.

1. SELECT guestname, guestaddress
FROM guest
WHERE guestaddress LIKE '%Glasgow%' AND (guestname LIKE 'Tony%' OR guestname LIKE '%Farrel');

2. SELECT DISTINCT h.*
FROM hotel h
INNER JOIN room r1 ON h.hotelno = r1.hotelno AND r1.type = 'double'
INNER JOIN room r2 ON h.hotelno = r2.hotelno AND r2.type = 'family';

3. SELECT DISTINCT h.*
FROM hotel h
INNER JOIN room r1 ON h.hotelno = r1.hotelno AND r1.type = 'single'
INNER JOIN room r2 ON h.hotelno = r2.hotelno AND r2.type = 'double'
LEFT JOIN room r3 ON h.hotelno = r3.hotelno AND r3.type = 'family'
WHERE r3.hotelno IS NULL;

4. SELECT *
FROM booking

```
WHERE guestno IN (10003, 10007) AND guestno != 1001;
```