DBMS Queries for practical

1	Identify primary keys and foreign keys for following database. Create tables and execute queries for given statements. employee(eid,ename,salary) assignment(projectid,eid) project(projectid,project_name,manager) manager(eid,ename) **Write queries for the following questions:** 1. Alter table to add address in employee table. 2. Display employee name and projects on which they are working/ 3. Display projectid, projectname and their managers. 4. Create view of employees working on 'Bank Management' project. 5. Print names of employees whose salary is greater than 40000 6. Update salary of each employee with increase of Rs.2000
2	Identify primary keys and foreign keys for following database. Create tables and execute queries for given statements. employee(eid, ename, salary) assignment(projectid,eid) project(projectid,project_name,manager) manager(eid,ename) **Write queries for the following questions:** 1. Modify eid to use auto_increment 2. Display Employees working in both projects 'Bank Management' and 'Content Management'. 3. Display average salary of organization. 4. Display employees who do not work on 'Bank Management' Project. 5. Delete employee whose id is 5. 6. Display employee having highest salary in oraganization.

3	Identify primary keys and foreign keys for following database. Create tables and execute queries for given statements.
	supplier(supplierid,sname,saddress)
	parts(part_id,part_name,color);
	catalog(supplierid,part_id,cost);
	Write queries for the following questions:
	1. Find name of supplier who supply ‘green’ parts.
	2. find name of suppliers who supply both blue and green parts.
	3. Find supplier who supply all parts.
	4. Find total cost of red parts.
	5. Find supplier who supply green parts with minimum cost.
	6. Update color of part having part_id = 4 and supplier_id = 2.

4	Identify primary keys and foreign keys for following database. Create tables and execute queries for given statements.
	emp(eid,ename,street,city);
	works(eid,company_name,salary);
	company(company_name,city);
	manages(eid,manager_id);
	Write queries for the following questions:
	1. Update company of employee name = ‘Prashant’ from ‘Infosys’ to ‘TCS’.
	2. Display names & cities of all employees who work for 'Infosys'
	3. Display names & Street address & of all employees who work in TCS cities and earn more than 20000.
	4. Find all employees in database who do not work for 'Infosys'.
	5. Find company wise total salary.
	6. Find names of all employees who work for 'Accenture'.

5	Identify primary keys and foreign keys for following database. Create tables and execute queries for given statements. employee(eid,ename,salary) assignment(projectid,eid) project(projectid,project_name,manager) manager(eid,ename) **Write queries for the following questions:** 1. Modify eid to use auto_increment 2. Display Employees working in both projects 'Bank Management' and 'Content Management'. 3. Display average salary of organization. 4. Display employees who do not work on 'Bank Management' Project. 5. Delete employee whose id is 5. 6. Display employee having highest salary in oraganization.

6	Identify primary keys and foreign keys for following database. Create tables and execute queries for given statements.
	supplier(supplierid,sname,saddress)
	parts(part_id,part_name,color);
	catalog(supplierid,part_id,cost);
	Write queries for the following questions:
	1. Find name of supplier who supply ‘green’ parts.
	2. Find name of suppliers who supply both blue and green parts.
	3. Find supplier who supply all parts.
	4. Find total cost of red parts.
	5. Find supplier who supply green parts with minimum cost.
	6. Update color of part having part_id = 4 and supplier_id = 2.

7	**Car Rental Database Management System** Customers (CustomerID, Name, Email, Phone, City) Cars (CarID, Model, Brand, Year, RentalPricePerDay, AvailabilityStatus) Rentals (RentalID, CustomerID, CarID, StartDate, EndDate, TotalAmount) **Write queries for the following questions:** 1. Create a Payments table with attributes: PaymentID, RentalID (FK), PaymentDate, AmountPaid, and PaymentMethod. 2. Update AvailabilityStatus of a car to 'Rented' for a specific CustomerID and CarID. 3. Retrieve Customer Name, Car Model, and Rental StartDate for rentals where RentalPricePerDay is above 1000. 4. Calculate the total rental amount collected per Car Brand. 5. Find the top 3 customers who have spent the most on rentals.
8	**Online Shopping System** 1. Customers (CustomerID, Name, Email, Phone, Address) 2. Products (ProductID, Name, Category, Price, StockQuantity) 3. Orders (OrderID, CustomerID, OrderDate, TotalAmount) 4. OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, Subtotal) **Write queries for the following questions:** 1. Create a Payments table with PaymentID, OrderID (FK), PaymentDate, AmountPaid, and PaymentMethod. 2. Update the stock quantity of a product after an order is placed. 3. Retrieve Customer Name, Order Date, and TotalAmount for orders where the total amount exceeds 5000. 4. Calculate the total sales per product category. 5. Find the top 5 customers who have spent the most on orders.
9	**Library Management System** 1. **Members** (MemberID, Name, Email, Phone, MembershipDate) 2. **Books** (BookID, Title, Author, Genre, CopiesAvailable) 3. **BorrowedBooks** (BorrowID, MemberID, BookID, BorrowDate, ReturnDate) **Write queries for the following questions:** 1. CREATE TABLE – Create a Fines table with FineID, MemberID (FK), Amount, Status, and FineDate. 2. UPDATE – Update CopiesAvailable when a book is borrowed or returned. 3. SELECT with JOIN & Operators – Retrieve Member Name, Book Title, and Borrow Date for books borrowed in the last month. 4. GROUP BY & Aggregate Function – Find the number of books borrowed per genre. 5. Joins & Aggregate Functions – Find the top 5 members who borrowed the most books.

10	**Hospital Management System** 1. Patients (PatientID, Name, Age, Gender, Contact) 2. Doctors (DoctorID, Name, Specialization, Contact) 3. Appointments (AppointmentID, PatientID, DoctorID, AppointmentDate, Status) 4. Bills (BillID, PatientID, Amount, PaymentStatus) **Write queries for the following questions:** 1. Create table – create a medicalrecords table with recordid, patientid (fk), diagnosis, prescription, and recorddate. 2. Update – update an appointment status to "completed" after a patient’s visit. 3. Select with join & operators – retrieve patient name, doctor name, and appointment date for patients who consulted a specific specialization. 4. Group by & aggregate function – find the total revenue collected per doctor. 5. Joins & aggregate functions – find the top 3 doctors who attended the highest number of appointments.
11	**University Database Management System** 1. Student Management: Store student details such as StudentID, Name, Age, Gender, Department, and Email. 2. Course Management: Maintain course details including CourseID, CourseName, Credits, and Department. 3. Enrollment System: Allow students to enroll in multiple courses, tracking StudentID, CourseID, EnrollmentDate, and Grade. 4. Professor Management: Store professor details like ProfessorID, Name, Department, and Email **Write queries for the following questions:** 1. Calculate percentage of students in each department 2. Detect duplicate enrollments (same student enrolled in same course in the same semester) 3. Find the semester with the highest average enrollments per course 4. List students with more than 3 enrollments 5. List all courses and the number of students enrolled in each

12	**University Database Management System** 1. Student Management: Store student details such as StudentID, Name, Age, Gender, Department, and Email. 2. Course Management: Maintain course details including CourseID, CourseName, Credits, and Department. 3. Enrollment System: Allow students to enroll in multiple courses, tracking StudentID, CourseID, EnrollmentDate, and Grade. 4. Professor Management: Store professor details like ProfessorID, Name, Department, and Email **Write queries for the following questions:** 1. List all courses a specific student is enrolled in (e.g., Pooja) 2. Identify students who failed more than 2 courses (assuming grade < 2.0 is fail) 3. Count the number of students in each department 4. Find courses with zero enrollments 5. Find the most popular course (course with the highest number of enrollments)
13	**University Database Management System** 1. Student Management: Store student details such as StudentID, Name, Age, Gender, Department, and Email. 2. Course Management: Maintain course details including CourseID, CourseName, Credits, and Department. 3. Enrollment System: Allow students to enroll in multiple courses, tracking StudentID, CourseID, EnrollmentDate, and Grade. 4. Professor Management: Store professor details like ProfessorID, Name, Department, and Email **Write queries for the following questions:** 1. Find students who have not enrolled in any course 2. Find students who are enrolled in more than 3 courses 3. Find the average grade of students per course 4. Retrieve the highest grade in each course 5. Get the department with the highest number of students

14	**Bank database Management System** 1. Customer (customer_id, name, address, phone, email) 2. Account (account_id, customer_id, account_type, balance, branch_id) 3. Branch (branch_id, branch_name, location, manager_id) 4. Transaction (transaction_id, account_id, transaction_type, amount, transaction_date) 5. Loan (loan_id, customer_id, amount, loan_type, status) 6. Employee (employee_id, name, position, branch_id, salary) **Write queries for the following questions:** 1. List all customers and their account details 2. Find the total balance in each branch 3. Find customers who have taken loans greater than Rs. 1,00,000 4. Retrieve transaction history for a specific account (e.g., Account ID: 101) 5. Find customers who have both a loan and an account 6. Create a view of high-value customers (balance > 1,00,000)
15	**Bank database Management System** 1. Customer (customer_id, name, address, phone, email) 2. Account (account_id, customer_id, account_type, balance, branch_id) 3. Branch (branch_id, branch_name, location, manager_id) 4. Transaction (transaction_id, account_id, transaction_type, amount, transaction_date) 5. Loan (loan_id, customer_id, amount, loan_type, status) 6. Employee (employee_id, name, position, branch_id, salary) **Write queries for the following questions:** 1. Find employees working in a specific branch (e.g., Branch ID: 3) 2. Get the details of the highest transaction made 3. Find accounts with a balance less than Rs. 5000 4. Update account balance after a deposit of Rs. 2000 in account ID 105 5. Delete inactive loan applications (status = 'Rejected') 6. Calculate the total loan amount per loan type