**S1**

1. Write the HTML5 code for generating the form as shown below. Apply the internal CSS to the following form to change the font size of the heading to 6pt and change the color to red and also change the background color to yellow.

ANS

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Styled Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

}

form {

width: 300px;

margin: 50px auto;

padding: 20px;

background-color: #fff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

font-size: 16pt;

color: red;

background-color: yellow;

padding: 10px;

text-align: center;

border-radius: 4px;

}

label {

display: block;

margin: 10px 0;

}

input {

width: 100%;

padding: 8px;

margin-bottom: 10px;

box-sizing: border-box;

}

button {

background-color: #4caf50;

color: #fff;

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<form>

<h2>Styled Form</h2>

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<button type="submit">Submit</button>

</form>

</body>

</html>

Q2) 1. Model the following Property system as a document database. Consider a set of Property, Owner. One owner can buy many properties.

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 05 documents in each collection Q

-- Create Owner table

CREATE TABLE Owner (

owner\_id INT PRIMARY KEY,

owner\_name VARCHAR(255) NOT NULL

);

-- Create Property table

CREATE TABLE Property (

property\_id INT PRIMARY KEY,

property\_name VARCHAR(255) NOT NULL,

area VARCHAR(100) NOT NULL,

rate INT NOT NULL,

owner\_id INT,

FOREIGN KEY (owner\_id) REFERENCES Owner(owner\_id)

);

-- Insert sample data into Owner table

INSERT INTO Owner (owner\_id, owner\_name) VALUES

(1, 'Mr. Patil'),

(2, 'Mrs. Sharma'),

(3, 'Mr. Singh'),

(4, 'Ms. Gupta'),

(5, 'Dr. Kumar');

-- Insert sample data into Property table

INSERT INTO Property (property\_id, property\_name, area, rate, owner\_id) VALUES

(101, 'Property1', 'Mumbai', 120000, 1),

(102, 'Property2', 'Nashik', 90000, 1),

(103, 'Property3', 'Pune', 150000, 2),

(104, 'Property4', 'Nashik', 80000, 3),

(105, 'Property5', 'Mumbai', 95000, 4);

Answer the following Queries

1. Display area wise property details. [3]

SELECT area, property\_name, rate, owner\_name

FROM Property

JOIN Owner ON Property.owner\_id = Owner.owner\_id;

1. Display property owned by 'Mr.Patil' having minimum rate [3]

SELECT property\_name, area, rate

FROM Property

JOIN Owner ON Property.owner\_id = Owner.owner\_id

WHERE owner\_name = 'Mr. Patil'

ORDER BY rate ASC

LIMIT 1;

1. Give the details of owner whose property is at “Nashik”. [4]

SELECT owner\_name, property\_name, area, rate

FROM Property

JOIN Owner ON Property.owner\_id = Owner.owner\_id

WHERE area = 'Nashik';

1. Display area of property whose rate is less than 100000. [4]

SELECT area, property\_name, rate

FROM Property

WHERE rate < 100000;

**S2**

Q1) Create a container add row inside it and add 3 columns inside row using BootStrap.

Ans

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<title>Bootstrap Container with Columns</title>

</head>

<body>

<div class="container">

<!-- Row -->

<div class="row">

<!-- Column 1 -->

<div class="col-md-4">

<div class="bg-light p-3">

<!-- Content for Column 1 -->

<p>Column 1 Content</p>

</div>

</div>

<!-- Column 2 -->

<div class="col-md-4">

<div class="bg-light p-3">

<!-- Content for Column 2 -->

<p>Column 2 Content</p>

</div>

</div>

<!-- Column 3 -->

<div class="col-md-4">

<div class="bg-light p-3">

<!-- Content for Column 3 -->

<p>Column 3 Content</p>

</div>

</div>

</div>

<!-- End Row -->

</div>

<!-- Bootstrap JS and Popper.js (Optional) -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

Q2) 1.Model the following system as a document database. Consider a database of newspaper, publisher, and city. Different publisher publishes various newspapers in different cities

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 5 documents in each collection. [3]

-- Create Publisher table

CREATE TABLE Publisher (

publisher\_id INT PRIMARY KEY,

publisher\_name VARCHAR(255) NOT NULL,

state VARCHAR(255) NOT NULL

);

-- Create City table

CREATE TABLE City (

city\_id INT PRIMARY KEY,

city\_name VARCHAR(255) NOT NULL

);

-- Create Newspaper table

CREATE TABLE Newspaper (

newspaper\_id INT PRIMARY KEY,

newspaper\_name VARCHAR(255) NOT NULL,

language VARCHAR(255) NOT NULL,

sale INT NOT NULL,

publisher\_id INT,

city\_id INT,

FOREIGN KEY (publisher\_id) REFERENCES Publisher(publisher\_id),

FOREIGN KEY (city\_id) REFERENCES City(city\_id)

);

-- Insert sample data into Publisher table

INSERT INTO Publisher (publisher\_id, publisher\_name, state) VALUES

(101, 'ABC Publishers', 'Maharashtra'),

(102, 'XYZ Publishers', 'Gujarat'),

(103, 'PQR Publishers', 'Maharashtra'),

(104, 'LMN Publishers', 'Gujarat'),

(105, 'EFG Publishers', 'Maharashtra');

-- Insert sample data into City table

INSERT INTO City (city\_id, city\_name) VALUES

(201, 'Nashik'),

(202, 'Mumbai'),

(203, 'Ahmedabad'),

(204, 'Pune'),

(205, 'Surat');

-- Insert sample data into Newspaper table

INSERT INTO Newspaper (newspaper\_id, newspaper\_name, language, sale, publisher\_id, city\_id) VALUES

(301, 'Marathi Times', 'Marathi', 50000, 101, 201),

(302, 'Gujarat News', 'Gujarati', 45000, 102, 203),

(303, 'Mumbai Herald', 'English', 60000, 103, 202),

(304, 'Pune Express', 'Marathi', 55000, 104, 204),

(305, 'Surat Tribune', 'Gujarati', 48000, 105, 205);

4. Answer the following Queries.

a. List all newspapers available “NASHIK” city [3]

SELECT newspaper\_name, language, sale

FROM Newspaper

WHERE city\_id = (SELECT city\_id FROM City WHERE city\_name = 'Nashik');

b. List all the newspaper of “Marathi” language [3]

SELECT newspaper\_name, language, sale

FROM Newspaper

WHERE language = 'Marathi';

c. Count no. of publishers of “Gujrat” state [4]

SELECT COUNT(DISTINCT publisher\_id) AS num\_publishers

FROM Publisher

WHERE state = 'Gujarat';

1. Write a cursor to show newspapers with highest sale in Maharashtra State [4]

DELIMITER //

CREATE PROCEDURE GetHighestSaleNewspapers()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE newspaper\_name VARCHAR(255);

DECLARE sale INT;

-- Declare cursor

DECLARE cur CURSOR FOR

SELECT newspaper\_name, sale

FROM Newspaper

WHERE city\_id IN (SELECT city\_id FROM City WHERE state = 'Maharashtra')

ORDER BY sale DESC;

-- Declare continue handler

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

-- Fetch and display data

FETCH cur INTO newspaper\_name, sale;

WHILE NOT done DO

SELECT newspaper\_name, sale;

FETCH cur INTO newspaper\_name, sale;

END WHILE;

CLOSE cur;

END //

DELIMITER ;

-- Call the stored procedure

CALL GetHighestSaleNewspapers();

**S3**

Q1) Write a bootstrap application to display thumbnails of the images.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Image Thumbnails</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-4">

<h2 class="text-center">Image Thumbnails</h2>

<div class="row">

<!-- Image 1 -->

<div class="col-md-4">

<div class="card">

<img src="https://placekitten.com/300/200" class="card-img-top" alt="Image 1">

<div class="card-body">

<p class="card-text">Image 1 description goes here.</p>

</div>

</div>

</div>

<!-- Image 2 -->

<div class="col-md-4">

<div class="card">

<img src="https://placekitten.com/300/201" class="card-img-top" alt="Image 2">

<div class="card-body">

<p class="card-text">Image 2 description goes here.</p>

</div>

</div>

</div>

<!-- Image 3 -->

<div class="col-md-4">

<div class="card">

<img src="https://placekitten.com/300/202" class="card-img-top" alt="Image 3">

<div class="card-body">

<p class="card-text">Image 3 description goes here.</p>

</div>

</div>

</div>

<!-- Add more image cards as needed -->

</div>

</div>

<!-- Bootstrap JS and Popper.js (Optional) -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

Q2)

1.Model the following system as a document database. Consider employee and department’s information.

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 5 documents in each collection.

-- Create Department table

CREATE TABLE Department (

dept\_id INT PRIMARY KEY,

dept\_name VARCHAR(255) NOT NULL

);

-- Create Employee table

CREATE TABLE Employee (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(255) NOT NULL,

salary DECIMAL(10, 2) NOT NULL,

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES Department(dept\_id)

);

-- Insert sample data into Department table

INSERT INTO Department (dept\_id, dept\_name) VALUES

(1, 'Sales'),

(2, 'HR'),

(3, 'IT');

-- Insert sample data into Employee table

INSERT INTO Employee (emp\_id, emp\_name, salary, department\_id) VALUES

(1, 'John', 60000, 1),

(2, 'Jane', 70000, 1),

(3, 'Bob', 80000, 2),

(4, 'Alice', 75000, 3),

(5, 'Charlie', 90000, 1);

4. Answer the following Queries.

a. Display name of employee who has highest salary [3]

SELECT emp\_name

FROM Employee

ORDER BY salary DESC

LIMIT 1;

b. Display biggest department with max. no. of employees [3]

SELECT d.dept\_name, COUNT(e.emp\_id) AS totalEmployees

FROM Department d

JOIN Employee e ON d.dept\_id = e.department\_id

GROUP BY d.dept\_id

ORDER BY totalEmployees DESC

LIMIT 1;

c. Write a cursor which shows department wise employee information [4]

DELIMITER //

CREATE PROCEDURE DepartmentWiseEmployeeInfo()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE dept\_id INT;

DECLARE dept\_name VARCHAR(255);

DECLARE emp\_id INT;

DECLARE emp\_name VARCHAR(255);

DECLARE salary DECIMAL(10, 2);

-- Declare cursor

DECLARE cur CURSOR FOR

SELECT d.dept\_id, d.dept\_name, e.emp\_id, e.emp\_name, e.salary

FROM Department d

LEFT JOIN Employee e ON d.dept\_id = e.department\_id

ORDER BY d.dept\_id, e.emp\_id;

-- Declare continue handler

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur;

-- Fetch and display data

FETCH cur INTO dept\_id, dept\_name, emp\_id, emp\_name, salary;

WHILE NOT done DO

IF dept\_id IS NOT NULL THEN

SELECT CONCAT('Department: ', dept\_name, ', Employee: ', emp\_name, ', Salary: ', salary) AS result;

END IF;

FETCH cur INTO dept\_id, dept\_name, emp\_id, emp\_name, salary;

END WHILE;

CLOSE cur;

END //

DELIMITER ;

-- Call the stored procedure

CALL DepartmentWiseEmployeeInfo();

d. List all the employees who work in Sales dept and salary > 50000

SELECT emp\_name

FROM Employee

WHERE department\_id = (SELECT dept\_id FROM Department WHERE dept\_name = 'Sales')

AND salary > 50000;

**S4**

Q1) Write a bootstrap program for the following “The .table class adds basic styling (light padding and only horizontal dividers) to a table” The table can have the first name, last name, and email id as columns

Ans

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<title>Bootstrap Table Example</title>

</head>

<body>

<div class="container mt-4">

<h2 class="text-center">User Information</h2>

<!-- Table with Bootstrap styling -->

<table class="table">

<thead>

<tr>

<th scope="col">First Name</th>

<th scope="col">Last Name</th>

<th scope="col">Email ID</th>

</tr>

</thead>

<tbody>

<!-- Sample data rows, replace with your actual data -->

<tr>

<td>John</td>

<td>Doe</td>

<td>john.doe@example.com</td>

</tr>

<tr>

<td>Jane</td>

<td>Smith</td>

<td>jane.smith@example.com</td>

</tr>

<!-- Add more rows as needed -->

</tbody>

</table>

</div>

<!-- Bootstrap JS and Popper.js (Optional) -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

Q2)

1.Model the following information system as a document database. Consider hospitals around Nashik. Each hospital may have one or more specializations like Pediatric, Gynaec, Orthopedic, etc. A person can recommend/provide review for a hospital. A doctor can give service to one or more hospitals.

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 10 documents in each collection

-- Create Hospital table

CREATE TABLE Hospital (

hospital\_id INT PRIMARY KEY,

hospital\_name VARCHAR(255) NOT NULL,

city VARCHAR(255) NOT NULL

);

-- Create Specialization table

CREATE TABLE Specialization (

specialization\_id INT PRIMARY KEY,

specialization\_name VARCHAR(255) NOT NULL

);

-- Create HospitalSpecialization table (to represent many-to-many relationship)

CREATE TABLE HospitalSpecialization (

hospital\_id INT,

specialization\_id INT,

PRIMARY KEY (hospital\_id, specialization\_id),

FOREIGN KEY (hospital\_id) REFERENCES Hospital(hospital\_id),

FOREIGN KEY (specialization\_id) REFERENCES Specialization(specialization\_id)

);

-- Create Doctor table

CREATE TABLE Doctor (

doctor\_id INT PRIMARY KEY,

doctor\_name VARCHAR(255) NOT NULL

);

-- Create DoctorHospital table (to represent many-to-many relationship)

CREATE TABLE DoctorHospital (

doctor\_id INT,

hospital\_id INT,

PRIMARY KEY (doctor\_id, hospital\_id),

FOREIGN KEY (doctor\_id) REFERENCES Doctor(doctor\_id),

FOREIGN KEY (hospital\_id) REFERENCES Hospital(hospital\_id)

);

-- Create Review table

CREATE TABLE Review (

review\_id INT PRIMARY KEY,

hospital\_id INT,

reviewer\_name VARCHAR(255) NOT NULL,

rating DECIMAL(2, 1) NOT NULL,

FOREIGN KEY (hospital\_id) REFERENCES Hospital(hospital\_id)

);

-- Insert sample data into Hospital table

INSERT INTO Hospital (hospital\_id, hospital\_name, city) VALUES

(1, 'City Hospital', 'Nashik'),

(2, 'Pediatric Care', 'Nashik'),

(3, 'Ortho Clinic', 'Mumbai'),

(4, 'Gynaecology Center', 'Nashik'),

(5, 'Specialty Hospital', 'Pune'),

(6, 'General Hospital', 'Nashik'),

(7, 'Ortho & More', 'Nashik'),

(8, 'Metro Hospital', 'Mumbai'),

(9, 'Women's Health Center', 'Nashik'),

(10, 'Child Wellness Clinic', 'Pune');

-- Insert sample data into Specialization table

INSERT INTO Specialization (specialization\_id, specialization\_name) VALUES

(1, 'Pediatric'),

(2, 'Gynaecology'),

(3, 'Orthopedic'),

(4, 'General');

-- Insert sample data into HospitalSpecialization table

INSERT INTO HospitalSpecialization (hospital\_id, specialization\_id) VALUES

(1, 1),

(2, 1),

(3, 3),

(4, 2),

(5, 4),

(6, 4),

(7, 3),

(8, 3),

(9, 2),

(10, 1);

-- Insert sample data into Doctor table

INSERT INTO Doctor (doctor\_id, doctor\_name) VALUES

(1, 'Dr. Deshmukh'),

(2, 'Dr. Patel'),

(3, 'Dr. Singh'),

(4, 'Dr. Sharma'),

(5, 'Dr. Gupta');

-- Insert sample data into DoctorHospital table

INSERT INTO DoctorHospital (doctor\_id, hospital\_id) VALUES

(1, 1),

(1, 2),

(2, 3),

(3, 4),

(4, 5),

(4, 6),

(5, 7),

(5, 8),

(5, 9),

(5, 10);

-- Insert sample data into Review table

INSERT INTO Review (review\_id, hospital\_id, reviewer\_name, rating) VALUES

(1, 1, 'Patient A', 4.5),

(2, 2, 'Patient B', 3.8),

(3, 3, 'Patient C', 4.2),

(4, 4, 'Patient D', 4.9),

(5, 5, 'Patient E', 3.5),

(6, 6, 'Patient F', 4.7),

(7, 7, 'Patient G', 3.0),

(8, 8, 'Patient H', 4.6),

(9, 9, 'Patient I', 3.9),

(10, 10, 'Patient J', 4.1);

4. Answer the following Queries

a. List the names of hospitals with………… specialization. [3]

SELECT h.hospital\_name

FROM Hospital h

JOIN HospitalSpecialization hs ON h.hospital\_id = hs.hospital\_id

JOIN Specialization s ON hs.specialization\_id = s.specialization\_id

WHERE s.specialization\_name = 'Orthopedic';

b. List the Names of all hospital located in ……. city [3]

SELECT hospital\_name

FROM Hospital

WHERE city = 'Nashik';

c. List the names of hospitals where Dr. Deshmukh visits [4]

SELECT h.hospital\_name

FROM Hospital h

JOIN DoctorHospital dh ON h.hospital\_id = dh.hospital\_id

JOIN Doctor d ON dh.doctor\_id = d.doctor\_id

WHERE d.doctor\_name = 'Dr. Deshmukh';

d. List the names of hospitals whose rating >=4

SELECT h.hospital\_name, r.rating

FROM Hospital h

JOIN Review r ON h.hospital\_id = r.hospital\_id

WHERE r.rating >= 4;

**S5**

Q1) Write a HTML code, which generate the following output [ Apply border, border radius tags ] List of Persons Srno Person Name Age Country 1 2 3 10

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

table {

border-collapse: collapse;

width: 50%;

margin: 20px auto;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: center;

}

th {

background-color: #f2f2f2;

}

td:nth-child(2) {

border-radius: 10px;

}

</style>

<title>List of Persons</title>

</head>

<body>

<h2 style="text-align: center;">List of Persons</h2>

<table>

<thead>

<tr>

<th>Srno</th>

<th>Person Name</th>

<th>Age</th>

<th>Country</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>John Doe</td>

<td>25</td>

<td>USA</td>

</tr>

<tr>

<td>2</td>

<td>Jane Smith</td>

<td>30</td>

<td>Canada</td>

</tr>

<tr>

<td>3</td>

<td>Bob Johnson</td>

<td>28</td>

<td>UK</td>

</tr>

<tr>

<td>10</td>

<td>Alice Brown</td>

<td>22</td>

<td>Australia</td>

</tr>

</tbody>

</table>

</body>

</html>

Q2)

1Model the following database. Many employees working on one project. A company has various ongoing projects.

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 5 documents in each collection. [3]

-- Create Project table

CREATE TABLE Project (

project\_id INT PRIMARY KEY,

project\_name VARCHAR(255) NOT NULL,

project\_type VARCHAR(255) NOT NULL,

duration\_months INT NOT NULL

);

-- Create Employee table

CREATE TABLE Employee (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(255) NOT NULL

);

-- Create EmployeeProject table (to represent many-to-many relationship)

CREATE TABLE EmployeeProject (

emp\_id INT,

project\_id INT,

PRIMARY KEY (emp\_id, project\_id),

FOREIGN KEY (emp\_id) REFERENCES Employee(emp\_id),

FOREIGN KEY (project\_id) REFERENCES Project(project\_id)

);

-- Insert sample data into Project table

INSERT INTO Project (project\_id, project\_name, project\_type, duration\_months) VALUES

(1, 'Project A', 'Development', 6),

(2, 'Project B', 'Testing', 4),

(3, 'Project C', 'Design', 2),

(4, 'Project D', 'Development', 5),

(5, 'Project E', 'Testing', 3);

-- Insert sample data into Employee table

INSERT INTO Employee (emp\_id, emp\_name) VALUES

(101, 'Mr. Patil'),

(102, 'Mr. Kumar'),

(103, 'Mrs. Singh'),

(104, 'Ms. Joshi'),

(105, 'Mr. Deshmukh');

-- Insert sample data into EmployeeProject table

INSERT INTO EmployeeProject (emp\_id, project\_id) VALUES

(101, 1),

(102, 1),

(103, 2),

(104, 3),

(105, 4),

(105, 5);

4. Answer the following Queries

a. List all names of projects where Project\_type =….. [3]

SELECT project\_name

FROM Project

WHERE project\_type = 'Development';

b. List all the projects with duration greater than 3 months [3]

SELECT project\_name

FROM Project

WHERE duration\_months > 3;

c. Count no. of employees working on ……..project [4]

SELECT COUNT(emp\_id) AS num\_employees

FROM EmployeeProject

WHERE project\_id = 1;

d. List the names of projects on which Mr. Patil is working

SELECT p.project\_name

FROM Project p

JOIN EmployeeProject ep ON p.project\_id = ep.project\_id

JOIN Employee e ON ep.emp\_id = e.emp\_id

WHERE e.emp\_name = 'Mr. Patil';

**S6**

Q1) Create a web page being rendered in the browser consists of many things - logo, informative text, pictures, hyperlinks, navigational structure and table

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sample Web Page</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

header {

text-align: center;

margin-bottom: 20px;

}

nav {

display: flex;

justify-content: center;

margin-bottom: 20px;

}

nav a {

margin: 0 10px;

text-decoration: none;

color: #333;

}

main {

max-width: 800px;

margin: 0 auto;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

</style>

</head>

<body>

<header>

<img src="logo.png" alt="Logo" width="100">

<h1>Company Name</h1>

</header>

<nav>

<a href="#home">Home</a>

<a href="#about">About Us</a>

<a href="#services">Services</a>

<a href="#contact">Contact</a>

</nav>

<main>

<section>

<h2>Welcome to Our Website</h2>

<p>This is some informative text about our company and what we do. Feel free to explore the content below.</p>

</section>

<section>

<h2>Gallery</h2>

<img src="image1.jpg" alt="Image 1" width="300">

<img src="image2.jpg" alt="Image 2" width="300">

<!-- Add more images as needed -->

</section>

<section>

<h2>Useful Links</h2>

<ul>

<li><a href="#link1">Link 1</a></li>

<li><a href="#link2">Link 2</a></li>

<li><a href="#link3">Link 3</a></li>

</ul>

</section>

<section>

<h2>Table Example</h2>

<table>

<thead>

<tr>

<th>Name</th>

<th>Age</th>

<th>Country</th>

</tr>

</thead>

<tbody>

<tr>

<td>John Doe</td>

<td>30</td>

<td>USA</td>

</tr>

<tr>

<td>Jane Smith</td>

<td>25</td>

<td>Canada</td>

</tr>

<!-- Add more rows as needed -->

</tbody>

</table>

</section>

</main>

</body>

</html>

Q2)

1Model the following information as a document database. A customer can take different policies and get the benefit. There are different types of policies provided by various companies

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 5 documents in each collection. [3]

-- Create Customer table

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255) NOT NULL

);

-- Create Policy table

CREATE TABLE Policy (

policy\_id INT PRIMARY KEY,

policy\_type VARCHAR(255) NOT NULL,

premium\_amount DECIMAL(10, 2) NOT NULL

);

-- Create CustomerPolicy table (to represent many-to-many relationship)

CREATE TABLE CustomerPolicy (

customer\_id INT,

policy\_id INT,

PRIMARY KEY (customer\_id, policy\_id),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (policy\_id) REFERENCES Policy(policy\_id)

);

-- Insert sample data into Customer table

INSERT INTO Customer (customer\_id, customer\_name) VALUES

(1, 'John Doe'),

(2, 'Jane Smith'),

(3, 'Bob Johnson'),

(4, 'Alice Brown'),

(5, 'Charlie Green');

-- Insert sample data into Policy table

INSERT INTO Policy (policy\_id, policy\_type, premium\_amount) VALUES

(101, 'Komal Jeevan', 5000),

(102, 'Term Insurance', 3000),

(103, 'Monthly', 100),

(104, 'Quarterly', 250),

(105, 'Half Yearly', 500);

-- Insert sample data into CustomerPolicy table

INSERT INTO CustomerPolicy (customer\_id, policy\_id) VALUES

(1, 101),

(2, 101),

(3, 102),

(4, 103),

(5, 105);

4. Answer the following Queries.

a. List the details of customers who have taken “Komal Jeevan” Policy [3]

SELECT c.customer\_name, p.policy\_type, p.premium\_amount

FROM Customer c

JOIN CustomerPolicy cp ON c.customer\_id = cp.customer\_id

JOIN Policy p ON cp.policy\_id = p.policy\_id

WHERE p.policy\_type = 'Komal Jeevan';

b. Display average premium amount [3]

SELECT AVG(premium\_amount) AS average\_premium

FROM Policy;

c. Increase the premium amount by 5% for policy type=”Monthly” [4]

UPDATE Policy

SET premium\_amount = premium\_amount \* 1.05

WHERE policy\_type = 'Monthly';

d. Count no. of customers who have taken policy type “half yearly”

SELECT COUNT(cp.customer\_id) AS num\_customers

FROM CustomerPolicy cp

JOIN Policy p ON cp.policy\_id = p.policy\_id

WHERE p.policy\_type = 'Half Yearly';

**S7**

Q1) Create a 3D text, apply appropriate font, style, color. Use : Hover in the style selector so that the 3D effects appear only when you hover over the text

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>3D Text Effect</title>

<style>

.threeD-text {

font-family: 'Arial', sans-serif;

font-size: 2em;

font-weight: bold;

color: #3498db;

text-transform: uppercase;

position: relative;

display: inline-block;

transition: transform 0.5s;

}

.threeD-text:hover {

transform: perspective(1000px) rotateX(20deg);

}

</style>

</head>

<body>

<div class="threeD-text">Hover Me</div>

</body>

</html>

Q2)

1Model the following information as a document database. A customer operates his bank account, does various transactions and get the banking services

2. Assume appropriate attributes and collections as per the query requirements. [3]

3. Insert at least 5 documents in each collection. [3]

-- Create Customer table

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL

);

-- Create Account table

CREATE TABLE Account (

account\_id INT PRIMARY KEY,

customer\_id INT,

account\_type VARCHAR(255) NOT NULL,

open\_date DATE NOT NULL,

branch VARCHAR(255) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

-- Create Transaction table

CREATE TABLE Transaction (

transaction\_id INT PRIMARY KEY,

account\_id INT,

transaction\_date DATE NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (account\_id) REFERENCES Account(account\_id)

);

-- Insert sample data into Customer table

INSERT INTO Customer (customer\_id, first\_name, last\_name) VALUES

(1, 'John', 'Doe'),

(2, 'Jane', 'Smith'),

(3, 'Bob', 'Johnson'),

(4, 'Alice', 'Brown'),

(5, 'Charlie', 'Green');

-- Insert sample data into Account table

INSERT INTO Account (account\_id, customer\_id, account\_type, open\_date, branch) VALUES

(101, 1, 'Saving', '2020-01-01', 'Main'),

(102, 2, 'Checking', '2020-01-01', 'Downtown'),

(103, 3, 'Loan', '2020-03-15', 'Main'),

(104, 4, 'Saving', '2020-01-01', 'Downtown'),

(105, 5, 'Loan', '2020-02-10', 'Main');

-- Insert sample data into Transaction table

INSERT INTO Transaction (transaction\_id, account\_id, transaction\_date, amount) VALUES

(1001, 101, '2020-01-02', 500),

(1002, 102, '2020-01-05', -200),

(1003, 103, '2020-03-20', 1000),

(1004, 104, '2020-01-03', 300),

(1005, 105, '2020-02-15', -800);

4. Answer the following Queries.

a. List names of all customers whose first name starts with a “S” [3]

SELECT \* FROM Customer WHERE first\_name LIKE 'S%';

b. List all customers who has open an account on 1/1/2020 in \_\_\_branch [3]

SELECT c.\* FROM Customer c

JOIN Account a ON c.customer\_id = a.customer\_id

WHERE a.open\_date = '2020-01-01' AND a.branch = 'Main';

c. List the names customers where acctype=”Saving” [4]

SELECT c.\* FROM Customer c

JOIN Account a ON c.customer\_id = a.customer\_id

WHERE a.account\_type = 'Saving';

d. Count total no. of loan account holder of …….branch [4]

SELECT COUNT(\*) AS num\_loan\_accounts

FROM Account

WHERE account\_type = 'Loan' AND branch = 'Main';

**S8**

Q1) Create a button with different style (Secondary, Primary, Success, Error, Info, Warning, Danger) using BootStrap

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<title>Bootstrap Buttons</title>

</head>

<body>

<div class="container mt-5">

<h2>Bootstrap Buttons</h2>

<button type="button" class="btn btn-secondary">Secondary Button</button>

<button type="button" class="btn btn-primary">Primary Button</button>

<button type="button" class="btn btn-success">Success Button</button>

<button type="button" class="btn btn-danger">Danger Button</button>

<button type="button" class="btn btn-info">Info Button</button>

<button type="button" class="btn btn-warning">Warning Button</button>

<button type="button" class="btn btn-danger">Danger Button</button>

</div>

<!-- Bootstrap JS and Popper.js (Optional) -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

Q2)

1.Model the following inventory information as a document database. The inventory keeps track of various items. The items are tagged in various categories. Items may be kept in various warehouses and each warehouse keeps track of the quantity of the item.

2. Assume appropriate attributes and collections as per the query requirements [3]

3. Insert at least 5 documents in each collection. [3]

-- Create Item table

CREATE TABLE Item (

item\_id INT PRIMARY KEY,

item\_name VARCHAR(255) NOT NULL,

tags INT NOT NULL,

status VARCHAR(1) NOT NULL,

height DECIMAL(5, 2) NOT NULL

);

-- Create Warehouse table

CREATE TABLE Warehouse (

warehouse\_id INT PRIMARY KEY,

warehouse\_name VARCHAR(255) NOT NULL

);

-- Create Inventory table (to represent the many-to-many relationship between Item and Warehouse)

CREATE TABLE Inventory (

item\_id INT,

warehouse\_id INT,

quantity INT NOT NULL,

PRIMARY KEY (item\_id, warehouse\_id),

FOREIGN KEY (item\_id) REFERENCES Item(item\_id),

FOREIGN KEY (warehouse\_id) REFERENCES Warehouse(warehouse\_id)

);

-- Insert sample data into Item table

INSERT INTO Item (item\_id, item\_name, tags, status, height) VALUES

(1, 'Laptop', 3, 'A', 10.5),

(2, 'Planner', 4, 'B', 8.2),

(3, 'Headphones', 2, 'C', 6.5),

(4, 'Chair', 5, 'A', 12.0),

(5, 'Desk', 3, 'B', 9.8);

-- Insert sample data into Warehouse table

INSERT INTO Warehouse (warehouse\_id, warehouse\_name) VALUES

(101, 'Main Warehouse'),

(102, 'Backup Warehouse'),

(103, 'Local Storage');

-- Insert sample data into Inventory table

INSERT INTO Inventory (item\_id, warehouse\_id, quantity) VALUES

(1, 101, 400),

(2, 102, 20),

(2, 103, 30),

(3, 101, 150),

(4, 102, 80),

(5, 103, 25);

4. Answer the following Queries.

a. List all the items qty is greater than 300 [3]

SELECT i.\*

FROM Item i

JOIN Inventory inv ON i.item\_id = inv.item\_id

WHERE inv.quantity > 300;

b. List all items which have tags less than 5 [3]

SELECT \*

FROM Item

WHERE tags < 5;

c. List all items having status equal to “B” or having quantity less than 50 and height of the product should be greater than 8 [4]

SELECT \*

FROM Item

WHERE status = 'B' OR (quantity < 50 AND height > 8);

d. Find all warehouse that keeps item “Planner” and having in stock quantity less than 20

SELECT w.\*

FROM Warehouse w

JOIN Inventory inv ON w.warehouse\_id = inv.warehouse\_id

JOIN Item i ON inv.item\_id = i.item\_id

WHERE i.item\_name = 'Planner' AND inv.quantity < 2;

**S9**

Q1) Write an HTML 5 program for student registration form for college admission. Use input type like search, email, date etc

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Student Registration Form</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

form {

max-width: 600px;

margin: 0 auto;

}

label {

display: block;

margin-bottom: 8px;

}

input, select {

width: 100%;

padding: 8px;

margin-bottom: 16px;

box-sizing: border-box;

}

button {

background-color: #4CAF50;

color: white;

padding: 10px 15px;

border: none;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<h2>Student Registration Form</h2>

<form action="#" method="post">

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<label for="dob">Date of Birth:</label>

<input type="date" id="dob" name="dob" required>

<label for="gender">Gender:</label>

<select id="gender" name="gender" required>

<option value="male">Male</option>

<option value="female">Female</option>

<option value="other">Other</option>

</select>

<label for="address">Address:</label>

<input type="text" id="address" name="address" required>

<label for="city">City:</label>

<input type="text" id="city" name="city" required>

<label for="state">State:</label>

<input type="text" id="state" name="state" required>

<label for="zipcode">Zip Code:</label>

<input type="text" id="zipcode" name="zipcode" pattern="[0-9]{5}" required>

<small>Format: 12345</small>

<button type="submit">Submit</button>

</form>

</body>

</html>

Q2

1. Model the following Customer Loan information as a documentdatabase. Consider Customer Loan information system where the customer can take many types of loans.

2. Assume appropriate attributes and collections as per the query requirements [3]

3. Insert at least 10 documents in each collection. [3]

-- Create Customer table

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255) NOT NULL,

address VARCHAR(255) NOT NULL,

city VARCHAR(255) NOT NULL

);

-- Create Loan table

CREATE TABLE Loan (

loan\_id INT PRIMARY KEY,

customer\_id INT,

loan\_type VARCHAR(255) NOT NULL,

loan\_amount DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

-- Insert sample data into Customer table

INSERT INTO Customer (customer\_id, customer\_name, address, city) VALUES

(1, 'Mr. Patil', '123 Main St', 'Pimpri'),

(2, 'Mrs. Deshmukh', '456 Oak St', 'Mumbai'),

(3, 'Ms. Sharma', '789 Maple St', 'Pimpri'),

-- Add more customers as needed;

-- Insert sample data into Loan table

INSERT INTO Loan (loan\_id, customer\_id, loan\_type, loan\_amount) VALUES

(101, 1, 'Home Loan', 150000),

(102, 2, 'Car Loan', 50000),

(103, 3, 'Education Loan', 80000),

-- Add more loans as needed;

4. Answer the following Queries.

a. List all customers whose name starts with 'D' character [3]

SELECT \*

FROM Customer

WHERE customer\_name LIKE 'D%';

b.List the names of customer in descending order who has taken a loan from Pimpri city. [3]

SELECT c.customer\_name

FROM Customer c

JOIN Loan l ON c.customer\_id = l.customer\_id

WHERE c.city = 'Pimpri'

ORDER BY c.customer\_name DESC;

c.Display customer details having maximum loan amount. [4]

SELECT c.\*, l.loan\_amount

FROM Customer c

JOIN Loan l ON c.customer\_id = l.customer\_id

WHERE l.loan\_amount = (SELECT MAX(loan\_amount) FROM Loan);

d.Update the address of customer whose name is “Mr. Patil” and loan\_amt is greater than 100000.

UPDATE Customer

SET address = 'New Address'

WHERE customer\_name = 'Mr. Patil'

AND (SELECT loan\_amount FROM Loan WHERE customer\_id = (SELECT customer\_id FROM Customer WHERE customer\_name = 'Mr. Patil')) > 100000;

**S10**

Q1) Create a web page that shows use of transition properties, transition delay and duration effect.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Transition Effects Example</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

margin: 50px;

}

button {

padding: 15px 30px;

font-size: 16px;

background-color: #3498db;

color: #fff;

border: none;

cursor: pointer;

transition: background-color 0.3s ease, font-size 0.5s ease-in-out;

}

button:hover {

background-color: #2ecc71;

font-size: 20px;

transition-delay: 0.2s; /\* Adds a delay before starting the transition \*/

}

</style>

</head>

<body>

<h2>Transition Effects Example</h2>

<button>Hover me</button>

</body>

</html>

Q2

1.Model the following Online shopping information as a document database. Consider online shopping where the customer can get different products from different brands. Customers can rate the brands and products

2. Assume appropriate attributes and collections as per the query requirements [3]

3. Insert at least 5 documents in each collection. [3]

-- Create Product table

CREATE TABLE Product (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255) NOT NULL,

brand\_name VARCHAR(255) NOT NULL,

warranty\_period INT NOT NULL,

rating DECIMAL(3, 2) NOT NULL

);

-- Create Customer table

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255) NOT NULL,

city VARCHAR(255) NOT NULL

);

-- Create Purchase table (to represent the many-to-many relationship between Customer and Product)

CREATE TABLE Purchase (

purchase\_id INT PRIMARY KEY,

customer\_id INT,

product\_id INT,

purchase\_date DATE NOT NULL,

bill\_amount DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (product\_id) REFERENCES Product(product\_id)

);

-- Insert sample data into Product table

INSERT INTO Product (product\_id, product\_name, brand\_name, warranty\_period, rating) VALUES

(1, 'Laptop', 'BrandA', 1, 4.5),

(2, 'Smartphone', 'BrandB', 2, 3.8),

(3, 'Headphones', 'BrandC', 1, 4.0),

(4, 'TV', 'BrandA', 2, 4.2),

(5, 'Refrigerator', 'BrandB', 1, 4.8);

-- Insert sample data into Customer table

INSERT INTO Customer (customer\_id, customer\_name, city) VALUES

(101, 'John Doe', 'New York'),

(102, 'Jane Smith', 'Los Angeles'),

(103, 'Bob Johnson', 'Chicago'),

(104, 'Alice Brown', 'Houston'),

(105, 'Charlie Green', 'Miami');

-- Insert sample data into Purchase table

INSERT INTO Purchase (purchase\_id, customer\_id, product\_id, purchase\_date, bill\_amount) VALUES

(1001, 101, 1, '2023-08-15', 1200),

(1002, 102, 3, '2023-08-15', 80),

(1003, 103, 2, '2023-08-15', 500),

(1004, 104, 5, '2023-08-15', 1000),

(1005, 105, 4, '2023-08-15', 800);

4. Answer the following Queries.

a. List the names of product whose warranty period is one year [3 ]

SELECT product\_name

FROM Product

WHERE warranty\_period = 1;

b. List the customers has done purchase on “15/08/2023”. [3 ]

SELECT customer\_name

FROM Customer c

JOIN Purchase p ON c.customer\_id = p.customer\_id

WHERE p.purchase\_date = '2023-08-15';

c. Display the names of products with brand which have highest rating. [4]

SELECT p.product\_name, p.brand\_name

FROM Product p

WHERE rating = (SELECT MAX(rating) FROM Product);

d. Display customers who stay in …… city and billamt >50000

SELECT \*

FROM Customer c

JOIN Purchase p ON c.customer\_id = p.customer\_id

WHERE c.city = 'New York' AND p.bill\_amount > 50000;