

Slip 1

1)



Edit with WPS Office

2)

Property collection

```
[  
  {  
    "_id": ObjectId("prop1"),  
    "name": "Villa in Pune",  
    "area": "Pune",  
    "rate": 1200000,  
    "size": 2500,  
    "owner_id": ObjectId("owner1")  
  },  
  {  
    "_id": ObjectId("prop2"),  
    "name": "Apartment in Mumbai",  
    "area": "Mumbai",  
    "rate": 900000,  
    "size": 1500,  
    "owner_id": ObjectId("owner2")  
  },  
  {  
    "_id": ObjectId("prop3"),  
    "name": "Beach House in Goa",  
    "area": "Goa",  
    "rate": 2500000,  
    "size": 3500,  
    "owner_id": ObjectId("owner3")  
  },  
  {  
    "_id": ObjectId("prop4"),
```



Edit with WPS Office

```
"name": "Farmhouse in Nashik",
"area": "Nashik",
"rate": 800000,
"size": 4000,
"owner_id": ObjectId("owner4")
},
{
  "_id": ObjectId("prop5"),
  "name": "Bungalow in Bangalore",
  "area": "Bangalore",
  "rate": 1500000,
  "size": 2800,
  "owner_id": ObjectId("owner5")
}
]
```

Owner collection

```
[
  {
    "_id": ObjectId("owner1"),
    "name": "Mr. Patil",
    "contact": "mr.patil@email.com",
    "properties": [ObjectId("prop1")]
  },
  {
    "_id": ObjectId("owner2"),
    "name": "Mr. Sharma",
    "contact": "mr.sharma@email.com",
    "properties": [ObjectId("prop2")]
  }
]
```



Edit with WPS Office

```

},
{
  "_id": ObjectId("owner3"),
  "name": "Mr. Mehta",
  "contact": "mr.mehta@email.com",
  "properties": [ObjectId("prop3")]
},
{
  "_id": ObjectId("owner4"),
  "name": "Mr. Yadav",
  "contact": "mr.yadav@email.com",
  "properties": [ObjectId("prop4")]
},
{
  "_id": ObjectId("owner5"),
  "name": "Mr. Rao",
  "contact": "mr.rao@email.com",
  "properties": [ObjectId("prop5")]
}
]

```

3)

a)

```

db.properties.aggregate([
{
  $group: {
    _id: "$area",    // Group by area
    properties: { $push: { name: "$name", rate: "$rate", size: "$size" } }
  }
}
]

```



```
},  
{  
  $sort: { "_id": 1 } // Sort by area name (alphabetical)  
}  
])
```

b)

```
db.properties.aggregate([  
  {  
    $match: { "owner_id": ObjectId("owner1") } // Find properties owned by Mr. Patil  
  },  
  {  
    $sort: { "rate": 1 } // Sort by rate in ascending order  
  },  
  {  
    $limit: 1 // Take only the first property (with minimum rate)  
  }  
])
```

c)

```
db.properties.aggregate([  
  {  
    $match: { "area": "Nashik" } // Find properties in Nashik  
  },  
  {  
    $lookup: {  
      from: "owners", // Join with owners collection  
      localField: "owner_id",  
      foreignField: "_id",  
    }  
  }  
])
```



```

        as: "owner_details"
    }
},
{
    $unwind: "$owner_details" // Flatten the owner details
},
{
    $project: {
        "owner_details.name": 1,
        "owner_details.contact": 1,
        "name": 1,
        "area": 1
    }
}
])

```

```

d)
db.properties.find({
    "rate": { $lt: 1000000 }
},{
    "area": 1, // Only return the area field
    "_id": 0
})

```



Slip 2

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Bootstrap Example</title>
```

```
  <!-- Bootstrap CSS -->
```

```
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
```

```
<body>
```

```
  <!-- Container -->
```

```
  <div class="container">
```

```
    <!-- Row -->
```

```
    <div class="row">
```

```
      <!-- Column 1 -->
```

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

```
        <div class="p-3 border bg-light">Column 1</div>
```

```
      </div>
```

```
      <!-- Column 2 -->
```

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

```
        <div class="p-3 border bg-light">Column 2</div>
```

```
      </div>
```

```
      <!-- Column 3 -->
```

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

```
        <div class="p-3 border bg-light">Column 3</div>
```



Edit with WPS Office

```
    </div>
  </div>
</div>

<!-- Bootstrap JS and dependencies -->
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.min.js"></script>
</body>
</html>
```





2)

Newspaper collection

```
[
  {
    "_id": ObjectId("newspaper1"),
    "name": "Times of India",
    "language": "English",
    "sale": 500000,
    "publisher_id": ObjectId("publisher1"),
    "cities": ["Mumbai", "Pune", "Nashik"]
  },
  {
    "_id": ObjectId("newspaper2"),
    "name": "Lokmat",
    "language": "Marathi",
    "sale": 350000,
    "publisher_id": ObjectId("publisher2"),
    "cities": ["Pune", "Nagpur", "Mumbai"]
  },
  {
    "_id": ObjectId("newspaper3"),
    "name": "Maharashtra Times",
    "language": "Marathi",
    "sale": 200000,
    "publisher_id": ObjectId("publisher1"),
    "cities": ["Mumbai", "Pune", "Nashik"]
  },
  {
    "_id": ObjectId("newspaper4"),
```



Edit with WPS Office

```
"name": "Hindustan Times",
"language": "English",
"sale": 450000,
"publisher_id": ObjectId("publisher3"),
"cities": ["Delhi", "Mumbai"]
},
{
  "_id": ObjectId("newspaper5"),
  "name": "Sakal",
  "language": "Marathi",
  "sale": 150000,
  "publisher_id": ObjectId("publisher2"),
  "cities": ["Pune", "Nashik"]
}
]
```

Publisher collection

```
[
  {
    "_id": ObjectId("publisher1"),
    "name": "Times Group",
    "state": "Maharashtra",
    "newspapers": [ObjectId("newspaper1"), ObjectId("newspaper3")]
  },
  {
    "_id": ObjectId("publisher2"),
    "name": "Lokmat Group",
    "state": "Maharashtra",
    "newspapers": [ObjectId("newspaper2"), ObjectId("newspaper5")]
  }
]
```



```
},  
{  
  "_id": ObjectId("publisher3"),  
  "name": "Hindustan Media",  
  "state": "Delhi",  
  "newspapers": [ObjectId("newspaper4")]  
}  
]
```

Cities collection

```
[  
  {  
    "_id": ObjectId("city1"),  
    "name": "Nashik",  
    "state": "Maharashtra",  
    "newspapers": [ObjectId("newspaper1"), ObjectId("newspaper3"), ObjectId("newspaper5")]  
  },  
  {  
    "_id": ObjectId("city2"),  
    "name": "Pune",  
    "state": "Maharashtra",  
    "newspapers": [ObjectId("newspaper1"), ObjectId("newspaper2"), ObjectId("newspaper3"),  
ObjectId("newspaper5")]  
  },  
  {  
    "_id": ObjectId("city3"),  
    "name": "Mumbai",  
    "state": "Maharashtra",  
    "newspapers": [ObjectId("newspaper1"), ObjectId("newspaper2"), ObjectId("newspaper3"),
```



```

ObjectId("newspaper4")]
},
{
  "_id": ObjectId("city4"),
  "name": "Nagpur",
  "state": "Maharashtra",
  "newspapers": [ObjectId("newspaper2")]
},
{
  "_id": ObjectId("city5"),
  "name": "Delhi",
  "state": "Delhi",
  "newspapers": [ObjectId("newspaper4")]
}
]

```

3)

A)

```

db.cities.aggregate([
  { $match: { name: "Nashik" } },
  { $lookup: {
    from: "newspapers",
    localField: "newspapers",
    foreignField: "_id",
    as: "newspapers_details"
  }
},
  { $project: { "newspapers_details.name": 1, _id: 0 } }
])

```



B)

```
db.newspapers.find({ "language": "Marathi" }, { "name": 1, "_id": 0 })
```

C)

```
db.publishers.count({ "state": "Gujrat" })
```

D)

```
db.newspapers.aggregate([  
  {  
    $match: {  
      "cities": { $in: ["Mumbai", "Pune", "Nashik"] }  
    }  
  },  
  {  
    $sort: { "sale": -1 } // Sort by sale in descending order  
  },  
  { $limit: 1 } // Show only the top newspaper by sale  
])
```



Slip 3

1)

```
<!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-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
```

```
<body>
```

```
  <div class="container my-5">
```

```
    <h1 class="text-center mb-4">Image Thumbnails</h1>
```

```
    <!-- Row to display thumbnails -->
```

```
    <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 g-4">
```

```
      <!-- Thumbnail 1 -->
```

```
      <div class="col">
```

```
        <div class="card">
```

```
          
```

```
          <div class="card-body">
```

```
            <p class="card-text text-center">Image 1</p>
```

```
          </div>
```

```
        </div>
```

```
      </div>
```



Edit with WPS Office

<!-- Thumbnail 2 -->

<div class="col">

<div class="card">



<div class="card-body">

<p class="card-text text-center">Image 2</p>

</div>

</div>

</div>

<!-- Thumbnail 3 -->

<div class="col">

<div class="card">



<div class="card-body">

<p class="card-text text-center">Image 3</p>

</div>

</div>

</div>

<!-- Thumbnail 4 -->

<div class="col">

<div class="card">



<div class="card-body">

<p class="card-text text-center">Image 4</p>

</div>

</div>

</div>



Edit with WPS Office

```
<!-- Thumbnail 5 -->
<div class="col">
  <div class="card">
    
    <div class="card-body">
      <p class="card-text text-center">Image 5</p>
    </div>
  </div>
</div>

<!-- Thumbnail 6 -->
<div class="col">
  <div class="card">
    
    <div class="card-body">
      <p class="card-text text-center">Image 6</p>
    </div>
  </div>
</div>

<!-- Bootstrap JS and dependencies -->
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.min.js"></script>
</body>
</html>
```





2)

Employee collection

```
[
  { "_id": 1, "name": "Alice", "salary": 60000, "department_id": 101, "designation": "Manager" },
  { "_id": 2, "name": "Bob", "salary": 45000, "department_id": 102, "designation": "Sales Executive" },
  { "_id": 3, "name": "Charlie", "salary": 80000, "department_id": 101, "designation": "Lead Engineer" },
  { "_id": 4, "name": "Diana", "salary": 55000, "department_id": 103, "designation": "HR Specialist" },
  { "_id": 5, "name": "Eve", "salary": 75000, "department_id": 102, "designation": "Sales Manager" }
]
```

Department collection

```
[
  { "_id": 101, "name": "Engineering", "num_employees": 2 },
  { "_id": 102, "name": "Sales", "num_employees": 2 },
  { "_id": 103, "name": "HR", "num_employees": 1 }
]
```

3)

A)

```
db.Employees.find().sort({ salary: -1 }).limit(1);
```

B)

```
db.Departments.find().sort({ num_employees: -1 }).limit(1);
```

C)

```
db.Departments.aggregate([
  {
    $lookup: {
```



Edit with WPS Office

```
    from: "Employees",
    localField: "_id",
    foreignField: "department_id",
    as: "employees"
  }
},
{
  $project: {
    name: 1,
    employees: { name: 1, salary: 1, designation: 1 }
  }
}
]);
```

D)

```
db.Employees.find({
  $and: [
    { salary: { $gt: 50000 } },
    { department_id: 102 }
  ]
});
```



Slip 4

1)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Table</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-5">
    <h2 class="mb-4">Basic Styled Table</h2>
    <table class="table">
      <thead>
        <tr>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email ID</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>John</td>
          <td>Doe</td>
          <td>john.doe@example.com</td>
```



Edit with WPS Office

```
</tr>
<tr>
  <td>Jane</td>
  <td>Smith</td>
  <td>jane.smith@example.com</td>
</tr>
<tr>
  <td>Mark</td>
  <td>Lee</td>
  <td>mark.lee@example.com</td>
</tr>
</tbody>
</table>
</div>
```

```
<!-- Bootstrap JS (optional) -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```



2)

Hospital collection

```
[
  { "_id": 1, "name": "City Hospital", "city": "Nashik", "specializations": ["Pediatric", "Orthopedic"],
    "rating": 4.5 },
  { "_id": 2, "name": "Health Care Center", "city": "Nashik", "specializations": ["Gynaec"], "rating": 3.8
  },
  { "_id": 3, "name": "Wellness Hospital", "city": "Pune", "specializations": ["Orthopedic", "Pediatric"],
    "rating": 4.0 },
  { "_id": 4, "name": "Life Line Hospital", "city": "Nashik", "specializations": ["Orthopedic"], "rating":
    4.2 },
  { "_id": 5, "name": "Healing Touch", "city": "Mumbai", "specializations": ["Gynaec", "Pediatric"],
    "rating": 3.9 }
]
```

Doctore collection

```
[
  { "_id": 101, "name": "Dr. Deshmukh", "specializations": ["Orthopedic"], "hospital_ids": [1, 4] },
  { "_id": 102, "name": "Dr. Mehta", "specializations": ["Pediatric"], "hospital_ids": [1, 3] },
  { "_id": 103, "name": "Dr. Sharma", "specializations": ["Gynaec"], "hospital_ids": [2, 5] }
]
```

Reviews collection

```
[
  { "_id": 201, "hospital_id": 1, "reviewer_name": "John", "review": "Great service!", "rating": 5 },
  { "_id": 202, "hospital_id": 2, "reviewer_name": "Alice", "review": "Decent experience", "rating": 3.5
  },
  { "_id": 203, "hospital_id": 4, "reviewer_name": "Robert", "review": "Excellent staff", "rating": 4.5 }
]
```



Edit with WPS Office

3

A)

```
db.Hospitals.find({ specializations: "Orthopedic" }, { name: 1, _id: 0 });
```

B)

```
db.Hospitals.find({ city: "Nashik" }, { name: 1, _id: 0 });
```

C)

```
db.Hospitals.find({ _id: { $in: [1, 4] } }, { name: 1, _id: 0 });
```

D)

```
db.Hospitals.find({ rating: { $gte: 4 } }, { name: 1, _id: 0 });
```



Slip 5

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>List of Persons</title>
```

```
  <!-- Bootstrap CSS -->
```

```
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
  <style>
```

```
    /* Custom styles for the table */
```

```
    .table-custom {
```

```
      border: 2px solid #000;
```

```
      border-radius: 10px;
```

```
    }
```

```
    .table-custom th, .table-custom td {
```

```
      border: 1px solid #ddd;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div class="container my-5">
```



Edit with WPS Office

```
<h1 class="text-center mb-4">List of Persons</h1>
```

```
<!-- Table with borders and border-radius -->
```

```
<table class="table table-custom">
```

```
  <thead class="table-light">
```

```
    <tr>
```

```
      <th scope="col">Srno</th>
```

```
      <th scope="col">Person Name</th>
```

```
      <th scope="col">Age</th>
```

```
      <th scope="col">Country</th>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <!-- Empty records -->
```

```
    <tr>
```

```
      <td>1</td>
```

```
      <td></td>
```

```
      <td></td>
```

```
      <td></td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>2</td>
```

```
      <td></td>
```

```
      <td></td>
```

```
      <td></td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>3</td>
```

```
      <td></td>
```





```
        <td></td>
        <td></td>
    </tr>
</tbody>
</table>
</div>
```

```
<!-- Bootstrap JS and dependencies -->
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.min.js"></script>
</body>
</html>
```



2)

Project collection

```
[
  { "_id": 1, "name": "Alpha", "type": "Software", "duration_in_months": 6, "employee_ids": [101, 102] },
  { "_id": 2, "name": "Beta", "type": "Construction", "duration_in_months": 4, "employee_ids": [103, 104] },
  { "_id": 3, "name": "Gamma", "type": "Software", "duration_in_months": 2, "employee_ids": [101, 105] },
  { "_id": 4, "name": "Delta", "type": "Marketing", "duration_in_months": 5, "employee_ids": [102, 103] },
  { "_id": 5, "name": "Epsilon", "type": "Research", "duration_in_months": 3, "employee_ids": [104, 105] }
]
```

Employee collection

```
[
  { "_id": 101, "name": "Mr. Patil", "designation": "Developer", "project_ids": [1, 3] },
  { "_id": 102, "name": "Ms. Sharma", "designation": "Manager", "project_ids": [1, 4] },
  { "_id": 103, "name": "Mr. Khan", "designation": "Engineer", "project_ids": [2, 4] },
  { "_id": 104, "name": "Ms. Desai", "designation": "Architect", "project_ids": [2, 5] },
  { "_id": 105, "name": "Mr. Mehta", "designation": "Analyst", "project_ids": [3, 5] }
]
```

3)

A)

```
db.Projects.find({ type: "Software" }, { name: 1, _id: 0 });
```

B)



Edit with WPS Office

```
db.Projects.find({ duration_in_months: { $gt: 3 } }, { name: 1, _id: 0 });
```

C)

```
db.Projects.findOne({ name: "Alpha" }).employee_ids.length;
```

D)

```
db.Projects.find({ _id: { $in: [1, 3] } }, { name: 1, _id: 0 });
```



Slip 6

1)

1) <!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: 0;

padding: 0;

}

header {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

display: flex;

align-items: center;

}

header img {

height: 50px;

margin-right: 20px;

}

nav {

margin-top: 10px;

}

nav a {



```
margin: 0 10px;
text-decoration: none;
color: white;
}
nav a:hover {
    text-decoration: underline;
}
.container {
    padding: 20px;
}
.table-container {
    margin-top: 20px;
}
table {
    width: 100%;
    border-collapse: collapse;
}
table, th, td {
    border: 1px solid #ddd;
}
th, td {
    padding: 10px;
    text-align: left;
}
th {
    background-color: #f4f4f4;
}
.images {
    display: flex;
```



```
    gap: 20px;
    margin-top: 20px;
}
.images img {
    width: 150px;
    height: auto;
    border: 1px solid #ddd;
    border-radius: 5px;
}
</style>
</head>
<body>

<header>
    
    <h1>Welcome to Our Website</h1>
    <nav>
        <a href="#about">About</a>
        <a href="#services">Services</a>
        <a href="#contact">Contact</a>
    </nav>
</header>

<div class="container">
    <section id="about">
        <h2>About Us</h2>
        <p>This is a sample web page demonstrating how to include various elements such as a logo,
text, pictures, links, and a table.</p>
    </section>
```



```
<section id="images">
  <h2>Our Gallery</h2>
  <div class="images">
    
    
    
  </div>
</section>
```

```
<section id="table">
  <h2>Our Team</h2>
  <div class="table-container">
    <table>
      <thead>
        <tr>
          <th>Name</th>
          <th>Role</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>John Doe</td>
          <td>Manager</td>
          <td>john.doe@example.com</td>
        </tr>
        <tr>
          <td>Jane Smith</td>
```



```
<td>Developer</td>
<td>jane.smith@example.com</td>
</tr>
<tr>
<td>Mark Lee</td>
<td>Designer</td>
<td>mark.lee@example.com</td>
</tr>
</tbody>
</table>
</div>
</section>
</div>

</body>
</html>
```





2)

Customer collection

```
[  
  { "_id": 1, "name": "John Doe", "age": 35, "policies_taken": [101, 102] },  
  { "_id": 2, "name": "Jane Smith", "age": 28, "policies_taken": [103] },  
  { "_id": 3, "name": "Robert Brown", "age": 45, "policies_taken": [104, 105] },  
  { "_id": 4, "name": "Emily Davis", "age": 32, "policies_taken": [101] },  
  { "_id": 5, "name": "Mark Wilson", "age": 50, "policies_taken": [102, 105] }  
]
```

Policies collection

```
[  
  { "_id": 101, "name": "Komal Jeevan", "policy_type": "Monthly", "premium_amount": 500,  
    "company_name": "LIC" },  
  { "_id": 102, "name": "Jeevan Anand", "policy_type": "Half Yearly", "premium_amount": 1200,  
    "company_name": "LIC" },  
  { "_id": 103, "name": "Health Plus", "policy_type": "Monthly", "premium_amount": 800,  
    "company_name": "ICICI" },  
  { "_id": 104, "name": "Life Shield", "policy_type": "Yearly", "premium_amount": 3000,  
    "company_name": "HDFC" },  
  { "_id": 105, "name": "Jeevan Saral", "policy_type": "Half Yearly", "premium_amount": 1500,  
    "company_name": "LIC" }  
]
```

3)

A)

```
db.Customers.find({ policies_taken: 101 });
```



Edit with WPS Office

B)

```
db.Policies.aggregate([
  { $group: { _id: null, average_premium: { $avg: "$premium_amount" } } }
]);
```

C)

```
db.Policies.updateMany(
  { policy_type: "Monthly" },
  { $mul: { premium_amount: 1.05 } }
);
```

D)

```
db.Policies.aggregate([
  { $match: { policy_type: "Half Yearly" } },
  { $lookup: { from: "Customers", localField: "_id", foreignField: "policies_taken", as: "customers" } },
  { $project: { customer_count: { $size: "$customers" } } }
]);
```



Slip 7

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>3D Text with Hover Effect</title>
```

```
<style>
```

```
body {
```

```
display: flex;
```

```
justify-content: center;
```

```
align-items: center;
```

```
height: 100vh;
```

```
margin: 0;
```

```
background: #f0f0f0;
```

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

```
}
```

```
.text {
```

```
font-size: 50px;
```

```
font-weight: bold;
```

```
color: #333;
```

```
text-transform: uppercase;
```

```
transition: transform 0.3s, text-shadow 0.3s, color 0.3s;
```

```
}
```

```
.text:hover {
```



Edit with WPS Office

```
transform: translateY(-10px);
color: #4CAF50;
text-shadow: 2px 2px 5px #aaa, -2px -2px 5px #aaa;
}
</style>
</head>
<body>
  <div class="text">Hover Me</div>
</body>
</html>
```



2)

Customer collection

```
[
  { "_id": 1, "first_name": "Sam", "last_name": "Smith", "account_ids": [101, 102], "branch":
"Downtown", "account_open_date": "2020-01-01" },
  { "_id": 2, "first_name": "Sophia", "last_name": "Johnson", "account_ids": [103], "branch": "Central",
"account_open_date": "2021-05-10" },
  { "_id": 3, "first_name": "Ryan", "last_name": "Williams", "account_ids": [104], "branch":
"Downtown", "account_open_date": "2020-01-01" },
  { "_id": 4, "first_name": "Sara", "last_name": "Brown", "account_ids": [105], "branch": "East",
"account_open_date": "2019-12-15" },
  { "_id": 5, "first_name": "Michael", "last_name": "Davis", "account_ids": [106], "branch":
"Downtown", "account_open_date": "2020-01-01" }
]
```

Accounts collection

```
[
  { "_id": 101, "account_type": "Saving", "balance": 5000, "branch": "Downtown" },
  { "_id": 102, "account_type": "Loan", "balance": 10000, "branch": "Downtown" },
  { "_id": 103, "account_type": "Saving", "balance": 2000, "branch": "Central" },
  { "_id": 104, "account_type": "Loan", "balance": 15000, "branch": "Downtown" },
  { "_id": 105, "account_type": "Saving", "balance": 7000, "branch": "East" },
  { "_id": 106, "account_type": "Loan", "balance": 8000, "branch": "Downtown" }
]
```



Edit with WPS Office

Transaction collection

```
[
  { "_id": 201, "account_id": 101, "transaction_type": "Deposit", "amount": 500, "date": "2022-11-20"
},
  { "_id": 202, "account_id": 102, "transaction_type": "Withdrawal", "amount": 1000, "date": "2022-10-15" },
  { "_id": 203, "account_id": 103, "transaction_type": "Deposit", "amount": 200, "date": "2022-09-10"
},
  { "_id": 204, "account_id": 104, "transaction_type": "Deposit", "amount": 500, "date": "2022-08-25"
},
  { "_id": 205, "account_id": 105, "transaction_type": "Withdrawal", "amount": 300, "date": "2022-07-05" }
]
```

3)

A)

```
db.Customers.find({ first_name: { $regex: "^S" } }, { first_name: 1, last_name: 1, _id: 0 });
```

B)

```
db.Customers.find({ account_open_date: "2020-01-01", branch: "Downtown" }, { first_name: 1, last_name: 1, _id: 0 });
```

C)

```
db.Customers.aggregate([
  { $lookup: { from: "Accounts", localField: "account_ids", foreignField: "_id", as: "accounts" } },
  { $match: { "accounts.account_type": "Saving" } },
  { $project: { first_name: 1, last_name: 1, _id: 0 } }
]);
```

D)



Edit with WPS Office

```
db.Accounts.countDocuments({ account_type: "Loan", branch: "Downtown" });
```

slip 8

```
1) <!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>Bootstrap Buttons</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-5">
```

```
<h2>Bootstrap Button Styles</h2>
```

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

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

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

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

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

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

```
<button type="button" class="btn btn-light">Light</button>
```

```
<button type="button" class="btn btn-dark">Dark</button>
```

```
</div>
```

```
<!-- Bootstrap JS -->
```

```
<script
```



Edit with WPS Office

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

2)

Item collection

```
[
  { "_id": 1, "name": "Notebook", "category": "Stationery", "tags": ["writing", "paper"], "dimensions":
    { "height": 10, "width": 8, "length": 1 }, "status": "A" },
  { "_id": 2, "name": "Planner", "category": "Stationery", "tags": ["calendar", "organization"],
    "dimensions": { "height": 12, "width": 8, "length": 1 }, "status": "B" },
  { "_id": 3, "name": "Marker", "category": "Supplies", "tags": ["writing", "ink", "board"], "dimensions":
    { "height": 6, "width": 1, "length": 1 }, "status": "A" },
  { "_id": 4, "name": "Chair", "category": "Furniture", "tags": ["office", "seating"], "dimensions":
    { "height": 40, "width": 20, "length": 20 }, "status": "B" },
  { "_id": 5, "name": "Table", "category": "Furniture", "tags": ["office", "desk"], "dimensions": { "height":
    30, "width": 50, "length": 80 }, "status": "A" }
]
```

Warehouse collection

```
[
  { "_id": 101, "name": "Central Warehouse", "location": "Downtown", "inventory": [ { "item_id": 1,
    "quantity": 500 }, { "item_id": 2, "quantity": 15 } ] },
  { "_id": 102, "name": "East Warehouse", "location": "Eastside", "inventory": [ { "item_id": 3,
    "quantity": 50 }, { "item_id": 4, "quantity": 30 } ] },
  { "_id": 103, "name": "West Warehouse", "location": "Westside", "inventory": [ { "item_id": 5,
    "quantity": 80 }, { "item_id": 2, "quantity": 10 } ] },
  { "_id": 104, "name": "South Warehouse", "location": "Southside", "inventory": [ { "item_id": 1,
    "quantity": 200 }, { "item_id": 3, "quantity": 300 } ] },
  { "_id": 105, "name": "North Warehouse", "location": "Northside", "inventory": [ { "item_id": 4,
    "quantity": 100 }, { "item_id": 5, "quantity": 60 } ] }
]
```



Edit with WPS Office



3)A)

```
db.Warehouses.aggregate([
  { $unwind: "$inventory" },
  { $match: { "inventory.quantity": { $gt: 300 } } },
  { $lookup: { from: "Items", localField: "inventory.item_id", foreignField: "_id", as: "item_details" } },
  { $project: { "item_details.name": 1, "inventory.quantity": 1, _id: 0 } }
]);
```

B)

```
db.Items.find({ $expr: { $lt: [{ $size: "$tags" }, 5] } }, { name: 1, tags: 1, _id: 0 });
```

C)

```
db.Warehouses.aggregate([
  { $unwind: "$inventory" },
  { $lookup: { from: "Items", localField: "inventory.item_id", foreignField: "_id", as: "item_details" } },
  { $unwind: "$item_details" },
  { $match: {
    $or: [
      { "item_details.status": "B" },
      { $and: [ { "inventory.quantity": { $lt: 50 } }, { "item_details.dimensions.height": { $gt: 8 } } ] }
    ]
  } },
  { $project: { "item_details.name": 1, _id: 0 } }
]);
```



D)

```
db.Warehouses.aggregate([
  { $unwind: "$inventory" },
  { $lookup: { from: "Items", localField: "inventory.item_id", foreignField: "_id", as: "item_details" } },
  { $unwind: "$item_details" },
  { $match: { "item_details.name": "Planner", "inventory.quantity": { $lt: 20 } } },
  { $project: { name: 1, location: 1, "item_details.name": 1, "inventory.quantity": 1, _id: 0 } }
]);
```



Slip 6

1)

```
<!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;
```

```
background-color: #f4f4f9;
```

```
padding: 20px;
```

```
}
```

```
.container {
```

```
max-width: 600px;
```

```
margin: 0 auto;
```

```
background-color: #ffffff;
```

```
padding: 20px;
```

```
border-radius: 8px;
```

```
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
```

```
}
```

```
h2 {
```



Edit with WPS Office

```
text-align: center;  
color: #333;  
}
```

```
label {  
  font-weight: bold;  
}
```

```
input, select, textarea {  
  width: 100%;  
  padding: 8px;  
  margin-bottom: 15px;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
}
```

```
.form-group {  
  margin-bottom: 20px;  
}
```

```
.radio-group, .checkbox-group {  
  margin-top: 10px;  
}
```

```
button {  
  width: 100%;  
  padding: 10px;  
  background-color: #4CAF50;  
  color: white;
```



```
border: none;
border-radius: 4px;
cursor: pointer;
}

button:hover {
background-color: #45a049;
}

</style>
</head>
<body>
<div class="container">
<h2>College Admission Registration</h2>
<form action="/submit-registration" method="POST">

<!-- Full Name -->
<div class="form-group">
<label for="full-name">Full Name:</label>
<input type="text" id="full-name" name="full-name" required placeholder="Enter your full name">
</div>

<!-- Email -->
<div class="form-group">
<label for="email">Email Address:</label>
<input type="email" id="email" name="email" required placeholder="Enter your email address">
</div>
```



```
<!-- Phone Number -->
<div class="form-group">
  <label for="phone">Phone Number:</label>
  <input type="tel" id="phone" name="phone" required placeholder="Enter your phone number">
</div>
```

```
<!-- Date of Birth -->
<div class="form-group">
  <label for="dob">Date of Birth:</label>
  <input type="date" id="dob" name="dob" required>
</div>
```

```
<!-- Gender -->
<div class="form-group radio-group">
  <label>Gender:</label>
  <input type="radio" id="male" name="gender" value="Male" required>
  <label for="male">Male</label>
  <input type="radio" id="female" name="gender" value="Female" required>
  <label for="female">Female</label>
</div>
```

```
<!-- Program Choice -->
<div class="form-group">
  <label for="program">Program of Study:</label>
  <select id="program" name="program" required>
    <option value="">Select Program</option>
    <option value="undergraduate">Undergraduate</option>
```



```
<option value="postgraduate">Postgraduate</option>
<option value="diploma">Diploma</option>
</select>
</div>

<!-- Search for Course -->
<div class="form-group">
  <label for="course-search">Search for Courses:</label>
  <input type="search" id="course-search" name="course-search" placeholder="Search
courses...">
</div>

<!-- Additional Information -->
<div class="form-group">
  <label for="comments">Additional Information:</label>
  <textarea id="comments" name="comments" rows="4" placeholder="Any additional
comments"></textarea>
</div>

<!-- Terms & Conditions -->
<div class="form-group checkbox-group">
  <input type="checkbox" id="terms" name="terms" required>
  <label for="terms">I agree to the <a href="#">terms and conditions</a></label>
</div>

<!-- Submit Button -->
<button type="submit">Submit Registration</button>
</form>
</div>
```



</body>

</html>

2)

Customer collection

[

{

"\_id": 1,

"name": "Mr. Patil",

"address": "123 Main Street, Pune",

"contact\_number": "1234567890",

"email": "patil@example.com",

"loans": [

{ "loan\_id": 1, "loan\_type": "Home Loan", "loan\_amount": 500000, "loan\_date": "2022-05-10",  
"city": "Pune", "status": "approved" },

{ "loan\_id": 2, "loan\_type": "Personal Loan", "loan\_amount": 200000, "loan\_date": "2023-01-15",  
"city": "Pune", "status": "approved" }

]

},

{

"\_id": 2,

"name": "Mr. Deshmukh",

"address": "456 Elm Street, Pimpri",

"contact\_number": "9876543210",



Edit with WPS Office



```
"email": "deshmukh@example.com",
"loans": [
  { "loan_id": 3, "loan_type": "Car Loan", "loan_amount": 300000, "loan_date": "2023-03-20", "city":
"Pimpri", "status": "approved" }
]
},
{
  "_id": 3,
  "name": "Ms. Shalini",
  "address": "789 Oak Avenue, Nashik",
  "contact_number": "5551234567",
  "email": "shalini@example.com",
  "loans": [
    { "loan_id": 4, "loan_type": "Education Loan", "loan_amount": 150000, "loan_date": "2021-10-10",
"city": "Nashik", "status": "approved" }
  ]
},
{
  "_id": 4,
  "name": "Mr. Dhawan",
  "address": "101 Pine Street, Pimpri",
  "contact_number": "6543210987",
  "email": "dhawan@example.com",
  "loans": [
    { "loan_id": 5, "loan_type": "Home Loan", "loan_amount": 800000, "loan_date": "2022-06-15",
"city": "Pimpri", "status": "approved" }
  ]
},
{
  "_id": 5,
```



```
"name": "Mr. Desai",
"address": "202 Birch Road, Pune",
"contact_number": "1122334455",
"email": "desai@example.com",
"loans": [
  { "loan_id": 6, "loan_type": "Personal Loan", "loan_amount": 120000, "loan_date": "2023-04-10",
"city": "Pune", "status": "approved" }
],
{
  "_id": 6,
  "name": "Ms. Devika",
  "address": "303 Cedar Lane, Pimpri",
  "contact_number": "2233445566",
  "email": "devika@example.com",
  "loans": [
    { "loan_id": 7, "loan_type": "Education Loan", "loan_amount": 100000, "loan_date": "2022-09-12",
"city": "Pimpri", "status": "approved" }
  ],
  {
    "_id": 7,
    "name": "Mr. Rao",
    "address": "404 Maple Drive, Nashik",
    "contact_number": "3344556677",
    "email": "rao@example.com",
    "loans": [
      { "loan_id": 8, "loan_type": "Home Loan", "loan_amount": 400000, "loan_date": "2023-02-18",
"city": "Nashik", "status": "approved" }
    ]
  }
```



```
},
{
  "_id": 8,
  "name": "Mr. Dinesh",
  "address": "505 Oakwood Lane, Pune",
  "contact_number": "4455667788",
  "email": "dinesh@example.com",
  "loans": [
    { "loan_id": 9, "loan_type": "Car Loan", "loan_amount": 250000, "loan_date": "2021-11-22", "city":
"Pune", "status": "approved" }
  ]
},
{
  "_id": 9,
  "name": "Ms. Darshita",
  "address": "606 Ashwood Street, Pimpri",
  "contact_number": "5566778899",
  "email": "darshita@example.com",
  "loans": [
    { "loan_id": 10, "loan_type": "Personal Loan", "loan_amount": 50000, "loan_date": "2023-05-05",
"city": "Pimpri", "status": "approved" }
  ]
},
{
  "_id": 10,
  "name": "Mr. Deepak",
  "address": "707 Birch Lane, Pune",
  "contact_number": "6677889900",
  "email": "deepak@example.com",
```



```
"loans": [  
  { "loan_id": 11, "loan_type": "Education Loan", "loan_amount": 180000, "loan_date": "2022-08-18", "city": "Pune", "status": "approved" }  
]  
}
```

3)

A)

```
db.Customers.find({ "name": { $regex: "^D", $options: "i" } }, { name: 1, _id: 0 });
```

B)

```
db.Customers.aggregate([  
  { $unwind: "$loans" },  
  { $match: { "loans.city": "Pimpri" } },  
  { $sort: { "name": -1 } },  
  { $project: { name: 1, _id: 0 } }  
]);
```

C)

```
db.Customers.aggregate([  
  { $unwind: "$loans" },  
  { $sort: { "loans.loan_amount": -1 } },  
  { $limit: 1 },  
  { $lookup: { from: "Loans", localField: "loans.loan_id", foreignField: "_id", as: "loan_details" } },  
  { $project: { name: 1, "loan_details.loan_type": 1, "loan_details.loan_amount": 1, "loan_details.city": 1, _id: 0 } }  
]);
```





```
.container {  
    text-align: center;  
}  
  
/* Button Styling */  
.transition-button {  
    background-color: #4CAF50;  
    color: white;  
    border: none;  
    padding: 20px 40px;  
    font-size: 18px;  
    cursor: pointer;  
    border-radius: 8px;  
    transition: all 0.5s ease-in-out; /* Transition for all properties */  
}  
  
/* Transition Effects */  
.transition-button:hover {  
    background-color: #45a049;  
    transform: scale(1.2); /* Make the button grow */  
}  
  
/* Add delay and duration for smooth transition */  
.transition-button:focus {  
    outline: none;  
    transition-delay: 0.3s; /* Delay before transition starts */  
}  
  
</style>
```



```
</head>
<body>
  <div class="container">
    <h1>CSS Transition Example</h1>
    <button class="transition-button">Hover Over Me</button>
  </div>
</body>
</html>
```

2)

Customer collection

```
[
  {
    "_id": 1,
    "name": "John Doe",
    "address": "123 Main Street, Pune",
    "email": "johndoe@example.com",
    "contact_number": "1234567890",
    "purchases": [
      { "purchase_id": 1, "product_id": 1, "purchase_date": "2023-08-15", "quantity": 1, "total_amount":
1000 }
    ]
  },
  {
    "_id": 2,
    "name": "Jane Smith",
    "address": "456 Elm Street, Mumbai",
    "email": "janesmith@example.com",
    "contact_number": "9876543210",
```



Edit with WPS Office

```
"purchases": [  
  { "purchase_id": 2, "product_id": 2, "purchase_date": "2023-08-16", "quantity": 2, "total_amount":  
3000 }  
]  
},  
{  
  "_id": 3,  
  "name": "Alice Johnson",  
  "address": "789 Oak Avenue, Pune",  
  "email": "alicej@example.com",  
  "contact_number": "5551234567",  
  "purchases": [  
    { "purchase_id": 3, "product_id": 3, "purchase_date": "2023-08-15", "quantity": 1, "total_amount":  
1500 }  
  ]  
},  
{  
  "_id": 4,  
  "name": "Robert Brown",  
  "address": "101 Pine Street, Nashik",  
  "email": "robertb@example.com",  
  "contact_number": "6543210987",  
  "purchases": [  
    { "purchase_id": 4, "product_id": 4, "purchase_date": "2023-08-17", "quantity": 1, "total_amount":  
2000 }  
  ]  
},  
{  
  "_id": 5,  
  "name": "Emma White",
```





```
"address": "202 Birch Road, Pune",
"email": "emmaw@example.com",
"contact_number": "1122334455",
"purchases": [
  { "purchase_id": 5, "product_id": 5, "purchase_date": "2023-08-15", "quantity": 2, "total_amount":
4000 }
]
}
```

3)

A)

```
db.Products.find({ "warranty_period": "1 year" }, { name: 1, _id: 0 });
```

B)

```
db.Purchases.aggregate([
  { $match: { "purchase_date": "2023-08-15" } },
  { $lookup: { from: "Customers", localField: "customer_id", foreignField: "_id", as: "customer_info"
} },
  { $unwind: "$customer_info" },
  { $project: { "customer_info.name": 1, "customer_info.email": 1, _id: 0 } }
]);
```

C)

```
db.Products.aggregate([
  { $lookup: { from: "Brands", localField: "brand", foreignField: "name", as: "brand_info" } },
  { $unwind: "$brand_info" },
  { $sort: { "brand_info.rating": -1 } },
  { $limit: 1 },
```



```
{ $project: { "name": 1, "brand_info.name": 1, _id: 0 } }  
]);
```

D)

```
db.Purchases.aggregate([  
  { $lookup: { from: "Customers", localField: "customer_id", foreignField: "_id", as: "customer_info"  
  } },  
  { $unwind: "$customer_info" },  
  { $match: { "customer_info.address": /Pune/i, "total_amount": { $gt: 50000 } } },  
  { $project: { "customer_info.name": 1, "customer_info.address": 1, _id: 0 } }  
]);
```

Slip 11

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Company Information</title>
```

```
</head>
```

```
<body>
```

```
  <!-- Create a frameset with three frames -->
```

```
  <frameset rows="20%, 30%, 50%">
```

```
    <!-- First frame for the company name -->
```

```
    <frame src="about:blank" name="headerFrame">
```

```
    <!-- Second frame for the department list with hyperlinks -->
```

```
    <frame src="about:blank" name="departmentFrame">
```



Edit with WPS Office

```

<!-- Third frame for displaying department information -->
<frame src="about:blank" name="infoFrame">
</frameset>

<script>
    // Set the content for the first frame (Company Name)
    document.getElementsByName("headerFrame")[0].document.body.innerHTML =
"<h1>Company XYZ</h1>";

    // Set the content for the second frame (Department List)
    document.getElementsByName("departmentFrame")[0].document.body.innerHTML = `
        <h2>Departments</h2>
        <ul>
            <li><a href="javascript:void(0)" onclick="showDepartmentInfo('HR')">Human
Resources</a></li>
            <li><a href="javascript:void(0)" onclick="showDepartmentInfo('IT')">Information
Technology</a></li>
            <li><a href="javascript:void(0)" onclick="showDepartmentInfo('Sales')">Sales</a></li>
            <li><a href="javascript:void(0)"
onclick="showDepartmentInfo('Marketing')">Marketing</a></li>
        </ul>
    `;

    // Function to display department information in the third frame
    function showDepartmentInfo(department) {
        let departmentInfo = "";
        if (department === "HR") {
            departmentInfo = "<h3>Human Resources</h3><p>The HR department is responsible
for managing employee relations, benefits, and recruitment.</p>";

```



```
    } else if (department === "IT") {  
        departmentInfo = "<h3>Information Technology</h3><p>The IT department manages  
all computer systems, networks, and software within the company.</p>";  
    } else if (department === "Sales") {  
        departmentInfo = "<h3>Sales</h3><p>The Sales department is responsible for  
generating revenue by selling the company's products and services.</p>";  
    } else if (department === "Marketing") {  
        departmentInfo = "<h3>Marketing</h3><p>The Marketing department focuses on  
promoting and advertising the company's products and services to potential customers.</p>";  
    }  
    document.getElementsByName("infoFrame")[0].document.body.innerHTML =  
departmentInfo;  
}  
</script>  
  
</body>  
</html>
```



2)

[

{

"\_id": 1,

"product\_name": "Smartphone",

"category": "Electronics",

"price": 15000,

"quantity\_in\_stock": 50

},

{

"\_id": 2,

"product\_name": "Laptop",

"category": "Electronics",

"price": 35000,

"quantity\_in\_stock": 30

},

{



```
"_id": 3,
"product_name": "Washing Machine",
"category": "Home Appliances",
"price": 20000,
"quantity_in_stock": 20
},
{
  "_id": 4,
  "product_name": "Refrigerator",
  "category": "Home Appliances",
  "price": 25000,
  "quantity_in_stock": 15
},
{
  "_id": 5,
  "product_name": "Air Conditioner",
  "category": "Home Appliances",
  "price": 30000,
  "quantity_in_stock": 10
}
]
```

Invoice collection

```
[
{
  "_id": 1,
  "order_id": 1,
  "invoice_date": "2023-08-01",
  "amount": 50000,
```



Edit with WPS Office

```

    "status": "Paid"
  },
  {
    "_id": 2,
    "order_id": 4,
    "invoice_date": "2023-08-07",
    "amount": 60000,
    "status": "Paid"
  },
  {
    "_id": 3,
    "order_id": 5,
    "invoice_date": "2023-08-10",
    "amount": 35000,
    "status": "Paid"
  }
]

```

4)

a)

```
db.Products.find({}, { "product_name": 1, "category": 1, "price": 1, "_id": 0 });
```

b)

```
db.Orders.find({ "total_value": { $gt: 20000 } });
```

c)

```

db.Orders.aggregate([
  { $match: { "order_status": { $ne: "Processed" } } },
  { $lookup: {

```



```
    from: "Invoices",
    localField: "_id",
    foreignField: "order_id",
    as: "invoice_info"
  }},
  { $match: { "invoice_info": { $size: 0 } } }
]);
```

d)

```
db.Orders.aggregate([
  { $match: { "customer_id": 1 } },
  { $lookup: {
    from: "Invoices",
    localField: "_id",
    foreignField: "order_id",
    as: "invoice_info"
  }},
  { $lookup: {
    from: "Customers",
    localField: "customer_id",
    foreignField: "_id",
    as: "customer_info"
  }},
  { $unwind: "$customer_info" },
  { $project: {
    "order_id": 1,
    "products": 1,
    "total_value": 1,
    "order_date": 1,
```





```
"invoice_info": 1,  
"customer_info.name": 1,  
"_id": 0  
}  
]);
```

Slip 12

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>Customer Registration - Departmental Store</title>
```

```
<style>
```

```
body {
```

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

```
margin: 20px;
```

```
background-color: #f4f4f4;
```

```
}
```

```
form {
```

```
background-color: white;
```

```
padding: 20px;
```



Edit with WPS Office

```
border-radius: 8px;
width: 400px;
margin: 0 auto;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}
h2 {
    text-align: center;
}
.form-group {
    margin-bottom: 15px;
}
label {
    font-weight: bold;
}
input[type="text"],
input[type="tel"],
textarea,
select {
    width: 100%;
    padding: 8px;
    margin-top: 5px;
    border-radius: 4px;
    border: 1px solid #ccc;
}
.form-group input[type="radio"],
.form-group input[type="checkbox"] {
    margin-right: 10px;
}
.form-group button {
```



```
background-color: #4CAF50;
color: white;
border: none;
padding: 10px 20px;
border-radius: 5px;
cursor: pointer;
}
.form-group button[type="reset"] {
background-color: #f44336;
}
.form-group button:hover {
opacity: 0.8;
}
</style>
</head>
<body>
```

<h2>Customer Registration Form</h2>

```
<form action="#" method="post" id="registrationForm">
  <!-- Customer Name -->
  <div class="form-group">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>
  </div>

  <!-- Contact Number -->
  <div class="form-group">
    <label for="contact">Contact Number:</label>
```



```
<input type="tel" id="contact" name="contact" required>
</div>
```

```
<!-- Gender -->
```

```
<div class="form-group">
  <label>Gender:</label>
  <input type="radio" id="male" name="gender" value="Male" required>
  <label for="male">Male</label>
  <input type="radio" id="female" name="gender" value="Female" required>
  <label for="female">Female</label>
  <input type="radio" id="other" name="gender" value="Other" required>
  <label for="other">Other</label>
</div>
```

```
<!-- Preferred Days -->
```

```
<div class="form-group">
  <label for="preferred_days">Preferred Days for Purchasing:</label>
  <select id="preferred_days" name="preferred_days[]" multiple required>
    <option value="Monday">Monday</option>
    <option value="Tuesday">Tuesday</option>
    <option value="Wednesday">Wednesday</option>
    <option value="Thursday">Thursday</option>
    <option value="Friday">Friday</option>
    <option value="Saturday">Saturday</option>
    <option value="Sunday">Sunday</option>
  </select>
</div>
```

```
<!-- Favorite Item -->
```



```
<div class="form-group">
  <label for="favorite_item">Favorite Item:</label>
  <select id="favorite_item" name="favorite_item" required>
    <option value="Groceries">Groceries</option>
    <option value="Clothing">Clothing</option>
    <option value="Electronics">Electronics</option>
    <option value="Furniture">Furniture</option>
    <option value="Toys">Toys</option>
  </select>
</div>

<!-- Suggestions -->
<div class="form-group">
  <label for="suggestions">Suggestions:</label>
  <textarea id="suggestions" name="suggestions" rows="4" placeholder="Enter any
suggestions here..."></textarea>
</div>

<!-- Submit and Reset Buttons -->
<div class="form-group">
  <button type="submit">Submit</button>
  <button type="reset">Reset</button>
</div>
</form>

</body>
</html>
```



2)

Movies collection

```
[
  {
    "_id": "movie1",
    "name": "Movie A",
    "budget": 5000000,
    "release_year": 2020,
    "actors": [{"actor_name": "Akshay Kumar", "role": "Lead Actor"}, {"actor_name": "Katrina Kaif",
"role": "Lead Actress"}],
    "producers": [{"producer_name": "Producer 1"}, {"producer_name": "Producer 2"}]
  },
  {
    "_id": "movie2",
    "name": "Movie B",
    "budget": 4000000,
    "release_year": 2021,
```



```
"actors": [{"actor_name": "Akshay Kumar", "role": "Supporting Actor"}, {"actor_name": "Katrina Kaif", "role": "Lead Actress"}],
```

```
"producers": [{"producer_name": "Producer 2"}, {"producer_name": "Producer 3"}]
```

```
},
```

```
{
```

```
"_id": "movie3",
```

```
"name": "Movie C",
```

```
"budget": 6000000,
```

```
"release_year": 2021,
```

```
"actors": [{"actor_name": "Akshay Kumar", "role": "Lead Actor"}, {"actor_name": "Priyanka Chopra", "role": "Supporting Actress"}],
```

```
"producers": [{"producer_name": "Producer 1"}, {"producer_name": "Producer 4"}]
```

```
},
```

```
{
```

```
"_id": "movie4",
```

```
"name": "Movie D",
```

```
"budget": 3000000,
```

```
"release_year": 2019,
```

```
"actors": [{"actor_name": "Ranbir Kapoor", "role": "Lead Actor"}, {"actor_name": "Deepika Padukone", "role": "Lead Actress"}],
```

```
"producers": [{"producer_name": "Producer 2"}]
```

```
},
```

```
{
```

```
"_id": "movie5",
```

```
"name": "Movie E",
```

```
"budget": 7000000,
```

```
"release_year": 2022,
```

```
"actors": [{"actor_name": "Akshay Kumar", "role": "Lead Actor"}, {"actor_name": "Alia Bhatt", "role": "Supporting Actress"}],
```

```
"producers": [{"producer_name": "Producer 3"}, {"producer_name": "Producer 4"}]
```



```
}  
]
```

Actors collection

```
[  
  {  
    "_id": "actor1",  
    "actor_name": "Akshay Kumar",  
    "movies": [  
      {"movie_name": "Movie A", "role": "Lead Actor"},  
      {"movie_name": "Movie B", "role": "Supporting Actor"},  
      {"movie_name": "Movie C", "role": "Lead Actor"},  
      {"movie_name": "Movie E", "role": "Lead Actor"}  
    ]  
  },  
  {  
    "_id": "actor2",  
    "actor_name": "Katrina Kaif",  
    "movies": [  
      {"movie_name": "Movie A", "role": "Lead Actress"},  
      {"movie_name": "Movie B", "role": "Lead Actress"}  
    ]  
  },  
  {  
    "_id": "actor3",  
    "actor_name": "Priyanka Chopra",  
    "movies": [  
      {"movie_name": "Movie C", "role": "Supporting Actress"}  
    ]  
  }  
]
```





```

},
{
  "_id": "actor4",
  "actor_name": "Ranbir Kapoor",
  "movies": [
    {"movie_name": "Movie D", "role": "Lead Actor"}
  ]
},
{
  "_id": "actor5",
  "actor_name": "Deepika Padukone",
  "movies": [
    {"movie_name": "Movie D", "role": "Lead Actress"}
  ]
}
]

```

Producer collection

```

[
  {
    "_id": "producer1",
    "producer_name": "Producer 1",
    "movies": [
      {"movie_name": "Movie A", "year": 2020},
      {"movie_name": "Movie C", "year": 2021}
    ]
  },
  {
    "_id": "producer2",

```



```
"producer_name": "Producer 2",
"movies": [
  {"movie_name": "Movie A", "year": 2020},
  {"movie_name": "Movie B", "year": 2021},
  {"movie_name": "Movie D", "year": 2019}
],
{
  "_id": "producer3",
  "producer_name": "Producer 3",
  "movies": [
    {"movie_name": "Movie B", "year": 2021},
    {"movie_name": "Movie E", "year": 2022}
  ],
  {
    "_id": "producer4",
    "producer_name": "Producer 4",
    "movies": [
      {"movie_name": "Movie C", "year": 2021},
      {"movie_name": "Movie E", "year": 2022}
    ],
    {
      "_id": "producer5",
      "producer_name": "Producer 5",
      "movies": [
        {"movie_name": "Movie E", "year": 2022}
      ]
    }
  ]
}
```



```
}  
]
```

4)

A)

```
db.movies.find().sort({ "budget": -1 }).limit(1)
```

b)

```
db.producers.aggregate([  
  { $unwind: "$movies" },  
  { $group: { _id: { producer_name: "$producer_name", year: "$movies.year" }, count: { $sum: 1 } }  
},  
  { $match: { count: { $gt: 1 } } }  
])
```

c)

```
db.actors.find({ "movies.movie_name": { $in: db.movies.find({ "actors.actor_name": "Akshay  
Kumar" }).map(movie => movie.name) } })
```

d)

```
db.movies.find({ "producers.1": { $exists: true } })
```



Slip 13

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Useful Web Page</title>
```

```
  <link rel="stylesheet" href="styles.css">
```

```
</head>
```

```
<body>
```

```
  <!-- Header Section -->
```

```
  <header>
```

```
    <div class="logo">
```

```
      <h1>My Awesome Website</h1>
```



Edit with WPS Office

```
</div>

<div class="tagline">

  <p>Your one-stop solution for everything!</p>

</div>

</header>
```

```
<!-- Navigation Section -->
```

```
<nav>

  <ul>

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

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

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

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

  </ul>

</nav>
```

```
<!-- Main Content Section -->
```

```
<main>

  <!-- Left Sidebar (Aside) -->

  <aside>

    <h2>Latest Updates</h2>

    <ul>

      <li><a href="#">New Product Launch</a></li>

      <li><a href="#">Special Offers</a></li>

      <li><a href="#">Blog Post: How to Use Our Services</a></li>

    </ul>

  </aside>
```

```
<!-- Main Section (Content) -->
```



Edit with WPS Office

```
<section>

  <h2>Welcome to Our Website!</h2>

  <p>This is a simple web page layout using HTML5 semantic tags. It contains a header,
navigation, sidebar, and main content area. Learn more about our services below.</p>


  <h3>Our Services</h3>

  <p>We offer a wide range of services tailored to your needs. Whether it's consulting,
development, or customer support, we are here to help!</p>

</section>

</main>


<!-- Footer Section -->

<footer>

  <p>&copy; 2024 My Awesome Website. All rights reserved.</p>

  <div class="social-media">

    <a href="#">Facebook</a> | <a href="#">Twitter</a> | <a href="#">Instagram</a>

  </div>

</footer>


</body>
</html>
```

Style.css

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}
```



Edit with WPS Office

```
header {  
    background-color: #333;  
    color: white;  
    padding: 10px 0;  
    text-align: center;  
}
```

```
header h1 {  
    margin: 0;  
}
```

```
nav ul {  
    list-style-type: none;  
    background-color: #444;  
    padding: 10px;  
    margin: 0;  
    text-align: center;  
}
```

```
nav ul li {  
    display: inline;  
    margin: 0 15px;  
}
```

```
nav ul li a {  
    color: white;  
    text-decoration: none;  
    font-weight: bold;
```



```
}
```

```
main {  
    display: flex;  
    justify-content: space-between;  
    padding: 20px;  
}
```

```
aside {  
    width: 25%;  
    background-color: #fff;  
    padding: 10px;  
    margin-right: 20px;  
}
```

```
section {  
    width: 70%;  
    background-color: #fff;  
    padding: 20px;  
}
```

```
footer {  
    background-color: #333;  
    color: white;  
    text-align: center;  
    padding: 10px;  
    position: fixed;  
    bottom: 0;  
    width: 100%;
```





```
}
```

```
footer .social-media a {  
    color: white;  
    text-decoration: none;  
    margin: 0 10px;  
}
```

2)

Student collection

```
{  
  "_id": 1,  
  "name": "John Doe",  
  "class": "FY",  
  "gender": "Male",  
  "email": "john.doe@example.com"  
},  
{  
  "_id": 2,  
  "name": "Jane Smith",  
  "class": "SY",  
  "gender": "Female",  
  "email": "jane.smith@example.com"  
},
```



```
{
  "_id": 3,
  "name": "Alice Brown",
  "class": "FY",
  "gender": "Female",
  "email": "alice.brown@example.com"
}
```

#### Competition collection

```
{
  "_id": 1,
  "name": "E-Rangoli",
  "category": "Art",
  "total_participants": 5,
  "winners": [
    {"rank": 1, "student_id": 1},
    {"rank": 2, "student_id": 3},
    {"rank": 3, "student_id": 2}
  ]
},
{
  "_id": 2,
  "name": "Programming",
  "category": "Technology",
  "total_participants": 8,
  "winners": [
    {"rank": 1, "student_id": 2},
    {"rank": 2, "student_id": 1},
    {"rank": 3, "student_id": 3}
  ]
}
```



```
]
}
```

Participants collection

```
{
  "_id": 1,
  "student_id": 1,
  "competition_id": 1,
  "status": "Completed"
},
{
  "_id": 2,
  "student_id": 2,
  "competition_id": 1,
  "status": "Completed"
},
{
  "_id": 3,
  "student_id": 3,
  "competition_id": 2,
  "status": "Completed"
}
```

a)

```
db.competitions.aggregate([
  {
    $group: {
      _id: null,
      average_participants: { $avg: "$total_participants" }
    }
  }
])
```



```
}  
}  
])
```

b)

```
db.competitions.find({ "name": "Programming" }, { "total_participants": 1 })
```

c)

```
db.competitions.aggregate([  
  {  
    $unwind: "$winners"  
  },  
  {  
    $sort: { "winners.rank": 1 }  
  },  
  {  
    $limit: 3  
  },  
  {  
    $lookup: {  
      from: "students",  
      localField: "winners.student_id",  
      foreignField: "_id",  
      as: "winner_details"  
    }  
  },  
  {  
    $project: {  
      "competition_name": 1,
```



```
    "winner_name": { $arrayElemAt: ["$winner_details.name", 0] },
    "rank": "$winners.rank"
  }
}
])
```

```
d)
db.participations.aggregate([
  {
    $match: { "competition_id": 1 }
  },
  {
    $lookup: {
      from: "students",
      localField: "student_id",
      foreignField: "_id",
      as: "student_details"
    }
  },
  {
    $unwind: "$student_details"
  },
  {
```



```
{  
  $match: { "student_details.class": "FY" }  
},  
{  
  $project: { "student_name": "$student_details.name", "class": "$student_details.class" }  
}  
])
```

Slip 14

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Travel Plan Booking Form</title>
```

```
  <style>
```

```
    body {
```

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

```
      margin: 20px;
```

```
      padding: 20px;
```

```
      background-color: #f4f4f4;
```

```
    }
```



Edit with WPS Office

```
form {  
    background-color: #fff;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
}
```

```
label {  
    display: block;  
    margin: 8px 0;  
    font-weight: bold;  
}
```

```
input[type="text"], input[type="tel"], input[type="email"], select, textarea {  
    width: 100%;  
    padding: 8px;  
    margin-bottom: 10px;  
    border-radius: 4px;  
    border: 1px solid #ccc;  
}
```

```
.checkbox-group {  
    margin: 10px 0;  
}
```

```
.checkbox-group label {  
    display: inline-block;  
    margin-right: 15px;  
}
```



```
.buttons {
  display: flex;
  justify-content: space-between;
}

.buttons button {
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
}

.buttons button[type="reset"] {
  background-color: #f1f1f1;
  border: 1px solid #ccc;
}

.buttons button[type="submit"] {
  background-color: #28a745;
  color: white;
  border: none;
}

.buttons button[type="reset"]:hover,
.buttons button[type="submit"]:hover {
  background-color: #218838;
}
</style>
</head>
```





<body>

<h1>Travel Plan Booking</h1>

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

<!-- Name -->

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

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

<!-- Address -->

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

<textarea id="address" name="address" rows="4" required></textarea>

<!-- Contact Number -->

<label for="contact">Contact No:</label>

<input type="tel" id="contact" name="contact" pattern="[0-9]{10}" placeholder="Enter 10 digit phone number" required>

<!-- Gender -->

<label>Gender:</label>

<input type="radio" id="male" name="gender" value="Male">

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="Female">

<label for="female">Female</label>

<!-- Preferred Season -->

<label for="season">Preferred Season:</label>

<div class="checkbox-group">

<input type="checkbox" id="summer" name="season" value="Summer">

<label for="summer">Summer</label>



Edit with WPS Office

```
<input type="checkbox" id="winter" name="season" value="Winter">
<label for="winter">Winter</label>
<input type="checkbox" id="monsoon" name="season" value="Monsoon">
<label for="monsoon">Monsoon</label>
</div>

<!-- Location Type -->
<label for="location">Location Type:</label>
<select id="location" name="location" required>
  <option value="" disabled selected>Select a location</option>
  <option value="Beach">Beach</option>
  <option value="Hill Station">Hill Station</option>
  <option value="Adventure">Adventure</option>
  <option value="Cultural">Cultural</option>
  <option value="Wildlife">Wildlife</option>
</select>

<!-- Buttons -->
<div class="buttons">
  <button type="submit">Submit</button>
  <button type="reset">Reset</button>
</div>
</form>

</body>
</html>
```



2)

#### **Cypher Queries to Create Nodes and Relationships:**

// Create Students

```
CREATE (s1:Student {name: 'John Doe', dob: '2001-04-15', income: 25000})
```

```
CREATE (s2:Student {name: 'Jane Smith', dob: '1999-09-22', income: 30000})
```

// Create Scholarships

```
CREATE (scholarship1:Scholarship {name: 'OBC Scholarship', category: 'OBC', income_limit: 50000, year: '2020-2021'})
```

```
CREATE (scholarship2:Scholarship {name: 'SC Scholarship', category: 'SC', income_limit: 40000, year: '2020-2021'})
```

// Create Family Members

```
CREATE (f1:FamilyMember {name: 'Suresh Kumar'})
```

```
CREATE (f2:FamilyMember {name: 'Anjali Patel'})
```



// Create Relationships

CREATE (s1)-[:APPLIES\_FOR]->(scholarship1)

CREATE (s2)-[:APPLIES\_FOR]->(scholarship2)

CREATE (s1)-[:BENEFITS\_FROM]->(scholarship1)

CREATE (s2)-[:BENEFITS\_FROM]->(scholarship2)

CREATE (s1)-[:RECOMMENDS]->(f1)

CREATE (s2)-[:RECOMMENDS]->(f2)

3)

a)

MATCH (s:Scholarship)

WHERE s.category = 'OBC'

RETURN s.name AS ScholarshipName

b)

MATCH (s:Student)-[:BENEFITS\_FROM]->(sch:Scholarship)

WHERE sch.year = '2020-2021'

RETURN COUNT(s) AS NumberOfStudentsBenefitted

c)

MATCH (s:Scholarship)

WHERE s.name = 'OBC Scholarship'

SET s.income\_limit = 55000

RETURN s.name AS ScholarshipName, s.income\_limit AS NewIncomeLimit

d)

MATCH (s:Student)-[:BENEFITS\_FROM]->(sch:Scholarship)

WITH sch, COUNT(s) AS numberOfBeneficiaries

ORDER BY numberOfBeneficiaries DESC



LIMIT 1

RETURN sch.name AS MostPopularScholarship, numberOfBeneficiaries

Slip 15

1)

2) Cypher Queries to Create Nodes and Relationships

// Create Movies

CREATE (movie1:Movie {title: 'Dilwale', release\_year: 2015, business: 400000000, awards: 5})

CREATE (movie2:Movie {title: 'My Name is Khan', release\_year: 2010, business: 300000000, awards: 7})

CREATE (movie3:Movie {title: 'Raees', release\_year: 2017, business: 500000000, awards: 3})

CREATE (movie4:Movie {title: 'Chennai Express', release\_year: 2013, business: 600000000, awards: 6})

// Create Actors

CREATE (actor1:Actor {name: 'Shahrukh Khan', dob: '1965-11-02', gender: 'Male'})

CREATE (actor2:Actor {name: 'Kajol', dob: '1974-08-05', gender: 'Female'})

CREATE (actor3:Actor {name: 'Mahira Khan', dob: '1984-12-21', gender: 'Female'})



Edit with WPS Office

```
CREATE (actor4:Actor {name: 'Deepika Padukone', dob: '1986-01-05', gender: 'Female'})
```

```
// Create Relationships
```

```
CREATE (actor1)-[:ACTED_IN]->(movie1)
```

```
CREATE (actor1)-[:ACTED_IN]->(movie2)
```

```
CREATE (actor1)-[:ACTED_IN]->(movie3)
```

```
CREATE (actor1)-[:ACTED_IN]->(movie4)
```

```
CREATE (actor2)-[:ACTED_IN]->(movie1)
```

```
CREATE (actor3)-[:ACTED_IN]->(movie2)
```

```
CREATE (actor4)-[:ACTED_IN]->(movie4)
```

3) a)

```
MATCH (m:Movie)
```

```
RETURN m.title AS Movie, m.business AS Business
```

```
ORDER BY m.business DESC
```

```
LIMIT 1
```

b)

```
MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)
```

```
RETURN m.title AS MovieTitle, m.release_year AS ReleaseYear, m.business AS Business,  
m.awards AS Awards, a.name AS ActorName
```

c)

```
MATCH (a:Actor {name: 'Shahrukh Khan'})-[:ACTED_IN]->(m:Movie)
```

```
RETURN m.title AS MovieTitle, m.release_year AS ReleaseYear, m.business AS Business
```



d)

MATCH (m:Movie)

WHERE m.awards > 2

RETURN m.title AS MovieTitle, m.awards AS Awards

Slip 16

1)

2) // Create FoodService Platforms

CREATE (swiggy:FoodService {name: 'Swiggy', location: 'India'})

CREATE (zomato:FoodService {name: 'Zomato', location: 'India'})

// Create Restaurants

CREATE (restaurant1:Restaurant {name: 'Pizza Hut', location: 'Mumbai', rating: 4.5})

CREATE (restaurant2:Restaurant {name: 'McDonald's', location: 'Delhi', rating: 3.9})

CREATE (restaurant3:Restaurant {name: 'Dominos', location: 'Bangalore', rating: 4.3})

// Create Customers

CREATE (customer1:Customer {name: 'Sam', contact\_number: '9876543210', email: 'sam@gmail.com'})



Edit with WPS Office

```
CREATE (customer2:Customer {name: 'Shivani', contact_number: '9887654321', email: 'shivani@yahoo.com'})
```

```
CREATE (customer3:Customer {name: 'Sonia', contact_number: '9998887777', email: 'sonia@hotmail.com'})
```

```
// Create Orders
```

```
CREATE (order1:Order {order_date: '2023-01-01', amount: 500})
```

```
CREATE (order2:Order {order_date: '2023-01-01', amount: 700})
```

```
CREATE (order3:Order {order_date: '2023-01-02', amount: 400})
```

```
// Create Ratings
```

```
CREATE (rating1:Rating {rating_value: 5})
```

```
CREATE (rating2:Rating {rating_value: 4})
```

```
CREATE (rating3:Rating {rating_value: 3})
```

```
// Create Recommendations
```

```
CREATE (recommendation1:Recommendation {recommendation_date: '2023-01-01'})
```

```
// Create Relationships
```

```
// Orders through Food Service
```

```
CREATE (customer1)-[:ORDERED_THROUGH]->(swiggy)
```

```
CREATE (customer2)-[:ORDERED_THROUGH]->(zomato)
```

```
CREATE (customer3)-[:ORDERED_THROUGH]->(swiggy)
```

```
// Placing Orders
```

```
CREATE (customer1)-[:PLACES_ORDER]->(order1)
```

```
CREATE (customer2)-[:PLACES_ORDER]->(order2)
```

```
CREATE (customer3)-[:PLACES_ORDER]->(order3)
```





// Restaurants Receiving Orders

CREATE (restaurant1)-[:RECEIVES\_ORDER]->(order1)

CREATE (restaurant2)-[:RECEIVES\_ORDER]->(order2)

CREATE (restaurant3)-[:RECEIVES\_ORDER]->(order3)

// Rating Food Service

CREATE (customer1)-[:GAVE\_RATING]->(rating1)

CREATE (customer2)-[:GAVE\_RATING]->(rating2)

CREATE (customer3)-[:GAVE\_RATING]->(rating3)

// Rating Restaurant

CREATE (customer1)-[:GAVE\_RATING]->(restaurant1)

CREATE (customer2)-[:GAVE\_RATING]->(restaurant2)

CREATE (customer3)-[:GAVE\_RATING]->(restaurant3)

// Recommendations

CREATE (customer1)-[:RECOMMENDED]->(customer2)

// Restaurants listed on Food Service

CREATE (restaurant1)-[:LISTED\_ON]->(swiggy)

CREATE (restaurant2)-[:LISTED\_ON]->(zomato)

CREATE (restaurant3)-[:LISTED\_ON]->(swiggy)

3)

a)

MATCH (c:Customer)-[:PLACES\_ORDER]->(o:Order)

WHERE o.order\_date = '2023-01-01'

RETURN COUNT(DISTINCT c) AS customers\_count



b)

```
MATCH (c:Customer)-[:ORDERED_THROUGH]->(f:FoodService {name: 'Swiggy'}), (c)-[:PLACES_ORDER]->(o:Order)
```

```
WHERE c.name STARTS WITH 'S'
```

```
RETURN c.name AS CustomerName
```

c)

```
MATCH (r:Restaurant)
```

```
WHERE r.rating >= 4
```

```
RETURN r.name AS HotelName, r.rating AS Rating
```

d)

```
MATCH (c:Customer)-[:RECOMMENDED]->(other:Customer)-[:ORDERED_THROUGH]->(f:FoodService)-[:LISTED_ON]->(r:Restaurant)
```

```
WHERE r.location = 'Mumbai' // You can replace 'Mumbai' with any area name
```

```
RETURN r.name AS HotelName, COUNT(*) AS Recommendations
```

```
ORDER BY Recommendations DESC
```

```
LIMIT 1
```

Slip 17

1)

2)

```
// Create FoodService Platforms
```

```
CREATE (swiggy:FoodService {name: 'Swiggy', location: 'India'})
```

```
CREATE (zomato:FoodService {name: 'Zomato', location: 'India'})
```

```
// Create Restaurants
```

```
CREATE (restaurant1:Restaurant {name: 'Pizza Hut', location: 'Mumbai', rating: 4.5})
```

```
CREATE (restaurant2:Restaurant {name: 'McDonald's', location: 'Delhi', rating: 3.9})
```

```
CREATE (restaurant3:Restaurant {name: 'Dominos', location: 'Bangalore', rating: 4.3})
```

```
// Create Customers
```



```
CREATE (customer1:Customer {name: 'Sam', contact_number: '9876543210', email: 'sam@gmail.com'})
```

```
CREATE (customer2:Customer {name: 'Shivani', contact_number: '9887654321', email: 'shivani@yahoo.com'})
```

```
CREATE (customer3:Customer {name: 'Sonia', contact_number: '9998887777', email: 'sonia@hotmail.com'})
```

```
// Create Orders
```

```
CREATE (order1:Order {order_date: '2023-01-01', amount: 500})
```

```
CREATE (order2:Order {order_date: '2023-01-01', amount: 700})
```

```
CREATE (order3:Order {order_date: '2023-01-02', amount: 400})
```

```
// Create Ratings
```

```
CREATE (rating1:Rating {rating_value: 5})
```

```
CREATE (rating2:Rating {rating_value: 4})
```

```
CREATE (rating3:Rating {rating_value: 3})
```

```
// Create Recommendations
```

```
CREATE (recommendation1:Recommendation {recommendation_date: '2023-01-01'})
```

```
// Create Relationships
```

```
// Orders through Food Service
```

```
CREATE (customer1)-[:ORDERED_THROUGH]->(swiggy)
```

```
CREATE (customer2)-[:ORDERED_THROUGH]->(zomato)
```

```
CREATE (customer3)-[:ORDERED_THROUGH]->(swiggy)
```

```
// Placing Orders
```

```
CREATE (customer1)-[:PLACES_ORDER]->(order1)
```

```
CREATE (customer2)-[:PLACES_ORDER]->(order2)
```

```
CREATE (customer3)-[:PLACES_ORDER]->(order3)
```



// Restaurants Receiving Orders

CREATE (restaurant1)-[:RECEIVES\_ORDER]->(order1)

CREATE (restaurant2)-[:RECEIVES\_ORDER]->(order2)

CREATE (restaurant3)-[:RECEIVES\_ORDER]->(order3)

// Rating Food Service

CREATE (customer1)-[:GAVE\_RATING]->(rating1)

CREATE (customer2)-[:GAVE\_RATING]->(rating2)

CREATE (customer3)-[:GAVE\_RATING]->(rating3)

// Rating Restaurant

CREATE (customer1)-[:GAVE\_RATING]->(restaurant1)

CREATE (customer2)-[:GAVE\_RATING]->(restaurant2)

CREATE (customer3)-[:GAVE\_RATING]->(restaurant3)

// Recommendations

CREATE (customer1)-[:RECOMMENDED]->(customer2)

// Restaurants listed on Food Service

CREATE (restaurant1)-[:LISTED\_ON]->(swiggy)

CREATE (restaurant2)-[:LISTED\_ON]->(zomato)

CREATE (restaurant3)-[:LISTED\_ON]->(swiggy)

3)

a)

MATCH (a:Author)-[:WROTE]->(b:Book)

WHERE b.genre = 'Comics'



```
RETURN a.name AS AuthorName
```

b)

```
MATCH (r:Reader)-[:READS]->(b:Book)-[:PUBLISHED_BY]->(p:Publisher {name: 'Sage'})
WHERE b.title = 'Harry Potter'
RETURN COUNT(r) AS reader_count
```

c)

```
MATCH (p:Publisher)
WHERE p.name STARTS WITH 'N'
RETURN p.name AS PublisherName
```

d)

```
MATCH (r:Reader)-[:GAVE_REVIEW]->(rev:Review)-[:REVIEWS]->(b:Book)
WHERE rev.rating >= 3 AND b.title = 'Harry Potter'
RETURN r.name AS ReaderName
```

Slip 18

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>2D Image Transformations</title>

<style>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;



Edit with WPS Office

```
    margin: 0;
    background-color: #f0f0f0;
}
.image-container {
    transition: transform 0.5s ease;
}
.image-container:hover {
    transform: rotate(45deg) scale(1.5) translate(50px, 50px);
}
img {
    max-width: 100%;
    height: auto;
    border: 2px solid #333;
    border-radius: 10px;
}
</style>
</head>
<body>
    <div class="image-container">
        
    </div>
</body>
</html>
```



2)

// Create doctors

CREATE (d1:Doctor {name: "Dr. Patel", specialization: "Orthopedic", area: "Kothrud"})

CREATE (d2:Doctor {name: "Dr. Mehta", specialization: "Pediatrics", area: "Seren Meadows"})

CREATE (d3:Doctor {name: "Dr. Desai", specialization: "Cancer Specialist", area: "Aundh"});

// Create hospitals

CREATE (h1:Hospital {name: "Apollo", location: "Baner"})

CREATE (h2:Hospital {name: "Ruby Hall", location: "Kothrud"});

// Create clinic

CREATE (c1:Clinic {name: "Healthy Kids Clinic", location: "Seren Meadows"});



```
// Create relationships
CREATE (d1)-[:VISITS]->(h2)
CREATE (d2)-[:VISITS]->(h1)
CREATE (d2)-[:VISITS]->(c1)
CREATE (p1:Person {name: "Anil"})-[:RECOMMENDS {rating: 5}]->(d2);
```

3)

a)

```
MATCH (d:Doctor)
WHERE d.specialization = "Orthopedic" AND d.area = "Kothrud"
RETURN d.name AS OrthopedicDoctors;
```

b)

```
MATCH (d:Doctor)
WHERE d.specialization = "Pediatrics"
RETURN d.name AS SpecializedDoctors;
```

c)

```
MATCH (p:Person)-[r:RECOMMENDS]->(d:Doctor)
WHERE d.specialization = "Pediatrics" AND d.area = "Seren Meadows"
RETURN d.name AS Doctor, COUNT(r) AS Recommendations
ORDER BY Recommendations DESC LIMIT 1;
```

d)

```
MATCH (d:Doctor)-[:VISITS]->(h:Hospital)
WITH d, COUNT(h) AS HospitalCount
WHERE HospitalCount > 2
RETURN d.name AS DoctorsVisitingMultipleHospitals;
```





Slip 19

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Progress Bar</title>
```

```
  <style>
```

```
    .progress-container {
```

```
      width: 100%;
```

```
      background-color: #f3f3f3;
```

```
      border: 1px solid #ccc;
```



Edit with WPS Office

```
border-radius: 5px;
overflow: hidden;
height: 30px;
}
.progress-bar {
width: 0;
height: 100%;
background-color: #4caf50;
text-align: center;
color: white;
line-height: 30px;
transition: width 1s ease;
}
.button {
margin-top: 20px;
padding: 10px 20px;
font-size: 16px;
background-color: #007bff;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
}
.button:hover {
background-color: #0056b3;
}
</style>
</head>
<body>
```



```

<div class="progress-container">
  <div class="progress-bar" id="progressBar">0%</div>
</div>

<button class="button" onclick="startDownload()">Start Download</button>

<script>
  let progress = 0;
  const colors = ["#4caf50", "#ff9800", "#f44336"]; // Green, Orange, Red
  let colorIndex = 0;

  function startDownload() {
    const progressBar = document.getElementById("progressBar");
    const interval = setInterval(() => {
      progress += 10;
      if (progress > 100) {
        clearInterval(interval);
        progressBar.textContent = "Download Complete";
        return;
      }
      progressBar.style.width = progress + "%";
      progressBar.textContent = progress + "%";

      if (progress % 30 === 0) {
        colorIndex = (colorIndex + 1) % colors.length;
        progressBar.style.backgroundColor = colors[colorIndex];
      }
    }, 1000);
  }
</script>

```



</body>

</html>

2)

// Create laptops

CREATE (l1:Laptop {name: "Inspiron 15", type: "Gaming", brand: "DELL", features: ["16GB RAM", "512GB SSD"]})

CREATE (l2:Laptop {name: "MacBook Air", type: "Ultrabook", brand: "Apple", features: ["8GB RAM", "256GB SSD"]})

CREATE (l3:Laptop {name: "ThinkPad X1", type: "Business", brand: "Lenovo", features: ["16GB RAM", "1TB SSD"]});

// Create manufacturers

CREATE (m1:Manufacturer {name: "DELL", location: "USA"})

CREATE (m2:Manufacturer {name: "Apple", location: "USA"})

CREATE (m3:Manufacturer {name: "Lenovo", location: "China"});



Edit with WPS Office

```
// Create customers
```

```
CREATE (c1:Customer {name: "John Doe", purchase_date: "26/01/2023"})
```

```
CREATE (c2:Customer {name: "Jane Smith", purchase_date: "15/02/2023"});
```

```
// Create relationships
```

```
CREATE (m1)-[:PRODUCES]->(l1)
```

```
CREATE (m2)-[:PRODUCES]->(l2)
```

```
CREATE (m3)-[:PRODUCES]->(l3)
```

```
CREATE (c1)-[:BUYS {purchase_date: "26/01/2023"}]->(l1)
```

```
CREATE (c2)-[:BUYS {purchase_date: "15/02/2023"}]->(l2)
```

```
CREATE (c1)-[:RATES {rating: 5}]->(l1)
```

```
CREATE (c2)-[:RECOMMENDS {comment: "Excellent performance"}]->(l1);
```

3)

a)

```
MATCH (l:Laptop {name: "Inspiron 15"})
```

```
RETURN l.features AS Characteristics;
```

b)

```
MATCH (c:Customer)-[:BUYS]->(l:Laptop {brand: "DELL"})
```

```
RETURN c.name AS Customers;
```

c)

```
MATCH (c:Customer)-[:BUYS {purchase_date: "26/01/2023"}]->(l:Laptop)
```

```
RETURN c.name AS Customers, l.name AS Laptop;
```



d)

MATCH (c:Customer)-[r:RECOMMENDS]->(l:Laptop)

RETURN l.name AS Laptop, COUNT(r) AS Recommendations

ORDER BY Recommendations DESC LIMIT 1;

Slip 20

1)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Download Progress</title>
```

```
  <style>
```

```
    #progress-container {
```

```
      width: 100%;
```

```
      background-color: #f3f3f3;
```

```
      border: 1px solid #ccc;
```



Edit with WPS Office

```

    margin-top: 20px;
}
#progress-bar {
    width: 0;
    height: 30px;
    background-color: #4caf50;
    text-align: center;
    line-height: 30px;
    color: white;
    font-weight: bold;
}
button {
    padding: 10px 20px;
    background-color: #007bff;
    color: white;
    border: none;
    cursor: pointer;
    font-size: 16px;
}
button:hover {
    background-color: #0056b3;
}
</style>
</head>
<body>
    <button id="start-btn">Start Download</button>
    <div id="progress-container">
        <div id="progress-bar">0%</div>
    </div>

```



```
<script>

const startButton = document.getElementById("start-btn");
const progressBar = document.getElementById("progress-bar");

startButton.addEventListener("click", () => {
  let progress = 0;
  const interval = setInterval(() => {
    progress += 5;
    progressBar.style.width = progress + "%";
    progressBar.textContent = progress + "%";
    if (progress >= 100) {
      clearInterval(interval);
      alert("Download completed");
    }
  }, 1000); // Increase by 5% every second
});
</script>
</body>
</html>
```





2)

a)

MATCH (p:Plant)

RETURN DISTINCT p.type AS PlantTypes;

b)

MATCH (p:Plant {type: "Flowering"})

WHERE p.rating >= 4

RETURN p.name AS PopularFloweringPlants, p.rating ORDER BY p.rating DESC;

c)

MATCH (c:Customer)-[r:PURCHASED]->(p:Plant)

WHERE r.quantity > 500 AND r.date >= date() - duration({days: 2})



```
RETURN p.name AS PlantName, r.quantity AS QuantitySold ORDER BY r.quantity DESC;
```

d)

```
MATCH (s:Supplier)-[:SUPPLIES]->(p:Plant {type: "Creeper"})
```

```
RETURN s.name AS SupplierName, COUNT(p) AS ProductCount
```

```
ORDER BY ProductCount DESC;
```

Slip 21

1)

```
<!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;
```

```
      background-color: #f4f4f9;
```

```
      margin: 0;
```



Edit with WPS Office

```
padding: 20px;  
}
```

```
.form-container {  
  max-width: 500px;  
  margin: auto;  
  background: #ffffff;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  
}
```

```
.form-container h2 {  
  text-align: center;  
  margin-bottom: 20px;  
  color: #333;  
}
```

```
.form-group {  
  margin-bottom: 15px;  
}
```

```
.form-group label {  
  display: block;  
  font-weight: bold;  
  margin-bottom: 5px;  
}
```

```
.form-group input,
```



```
.form-group select {  
  width: 100%;  
  padding: 10px;  
  font-size: 14px;  
  border: 1px solid #ddd;  
  border-radius: 5px;  
  box-sizing: border-box;  
}
```

```
.form-group input:focus {  
  border-color: #4caf50;  
  outline: none;  
}
```

```
.form-group .required {  
  color: red;  
  font-weight: bold;  
}
```

```
.form-group input[required]:not(:valid) {  
  border-color: #f44336;  
}
```

```
.form-group input[required]:not(:valid) + .error {  
  color: #f44336;  
  font-size: 12px;  
  margin-top: 5px;  
}
```



```
.form-group .error {  
  display: none;  
}
```

```
.form-group input:invalid + .error {  
  display: block;  
}
```

```
.btn-container {  
  text-align: center;  
}
```

```
.btn {  
  background: #4caf50;  
  color: white;  
  padding: 10px 15px;  
  font-size: 16px;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

```
.btn:hover {  
  background: #45a049;  
}
```

```
.btn:disabled {  
  background: #ddd;  
  cursor: not-allowed;
```



```

    }
</style>
</head>
<body>
  <div class="form-container">
    <h2>Student Registration Form</h2>
    <form>
      <div class="form-group">
        <label for="name">Name <span class="required">*</span></label>
        <input type="text" id="name" name="name" required />
        <span class="error">Name is required.</span>
      </div>

      <div class="form-group">
        <label for="email">Email <span class="required">*</span></label>
        <input type="email" id="email" name="email" required />
        <span class="error">Enter a valid email.</span>
      </div>

      <div class="form-group">
        <label for="contact">Contact No. <span class="required">*</span></label>
        <input type="tel" id="contact" name="contact" required pattern="[0-9]{10}" />
        <span class="error">Enter a valid 10-digit phone number.</span>
      </div>

      <div class="form-group">
        <label for="gender">Gender</label>
        <select id="gender" name="gender">
          <option value="">Select</option>

```



```
<option value="male">Male</option>
<option value="female">Female</option>
<option value="other">Other</option>
</select>
</div>

<div class="form-group">
  <label for="dob">Date of Birth</label>
  <input type="date" id="dob" name="dob" />
</div>

<div class="btn-container">
  <button type="submit" class="btn">Submit</button>
  <button type="reset" class="btn" style="background: #f44336;">Reset</button>
</div>
</form>
</div>
</body>
</html>
```



2)

// Create Brands

CREATE (:Brand {name: "Dr. Reddy"});

CREATE (:Brand {name: "Cipla"});

CREATE (:Brand {name: "SunPharma"});

// Create States

CREATE (:State {name: "Rajasthan"});

CREATE (:State {name: "Gujarat"});

CREATE (:State {name: "Maharashtra"});

// Create Medicines

CREATE (:Medicine {name: "Paracetamol", type: "Tablet", usagePercent: 92});





```
CREATE (:Medicine {name: "Cough Syrup", type: "Syrup", usagePercent: 80});  
CREATE (:Medicine {name: "Antibiotic Powder", type: "Powder", usagePercent: 55});
```

```
// Relationships
```

```
MATCH (b:Brand {name: "Cipla"}), (m:Medicine {name: "Paracetamol"})
```

```
CREATE (m)-[:MANUFACTURED_BY]->(b);
```

```
MATCH (b:Brand {name: "Dr. Reddy"}), (m:Medicine {name: "Cough Syrup"})
```

```
CREATE (m)-[:MANUFACTURED_BY]->(b);
```

```
MATCH (b:Brand {name: "SunPharma"}), (m:Medicine {name: "Antibiotic Powder"})
```

```
CREATE (m)-[:MANUFACTURED_BY]->(b);
```

```
// Usage Relationships
```

```
MATCH (m:Medicine {name: "Paracetamol"}), (s:State {name: "Rajasthan"})
```

```
CREATE (m)-[:USED_IN {usagePercent: 92}]->(s);
```

```
MATCH (m:Medicine {name: "Cough Syrup"}), (s:State {name: "Gujarat"})
```

```
CREATE (m)-[:USED_IN {usagePercent: 80}]->(s);
```

```
MATCH (m:Medicine {name: "Antibiotic Powder"}), (s:State {name: "Maharashtra"})
```

```
CREATE (m)-[:USED_IN {usagePercent: 55}]->(s);
```

```
a)
```

```
MATCH (m:Medicine)
```

```
RETURN m.name AS MedicineNames;
```

```
b)
```

```
MATCH (m:Medicine)-[u:USED_IN]->(s:State {name: "Rajasthan"})
```



```
WHERE u.usagePercent >= 90  
RETURN m.name AS HighlyUsedMedicines;
```

c)

```
MATCH (m:Medicine {type: "Tablet"})-[u:USED_IN]->(s:State {name: "Gujarat"})  
WHERE u.usagePercent >= 90  
RETURN m.name AS HighlyUsedTablets;
```

d)

```
MATCH (m:Medicine {type: "Powder"})  
RETURN m.name AS PowderMedicines;
```

Slip 22

1)

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>3D Text with Effects</title>  
  <style>  
    body {  
      background-color: #f0f0f0;  
      font-family: Arial, sans-serif;  
      display: flex;
```



Edit with WPS Office

```
justify-content: center;
align-items: center;
height: 100vh;
margin: 0;
}
```

```
.text-container {
  text-align: center;
}
```

```
.threeD-text {
  font-size: 50px;
  font-weight: bold;
  color: #2d3e50;
  text-shadow: 3px 3px 5px rgba(0, 0, 0, 0.4), -3px -3px 5px rgba(0, 0, 0, 0.4), 6px 6px 10px
  rgba(0, 0, 0, 0.2);
  transform: perspective(500px) rotateX(10deg) rotateY(10deg);
  display: inline-block;
  white-space: nowrap;
  overflow: hidden;
}
```

```
.text-content {
  text-overflow: ellipsis;
  word-wrap: break-word;
  max-width: 300px;
  overflow: hidden;
  margin-top: 20px;
  font-size: 20px;
```



```

        color: #555;
    }

    /* Adding animation for 3D text */
    .threeD-text:hover {
        transform: perspective(500px) rotateX(15deg) rotateY(15deg);
        color: #ff6347;
        transition: transform 0.3s ease, color 0.3s ease;
    }
</style>
</head>
<body>
    <div class="text-container">
        <h1 class="threeD-text">3D Text Effect</h1>
        <p class="text-content">This is an example of a 3D text effect with applied text-shadow,
word-wrap, and text-overflow properties. The text also has a hover effect to make it more
interactive and dynamic. If the text exceeds the container, it will be truncated with an
ellipsis.</p>
    </div>
</body>
</html>

```



2)

// Create Car Models

CREATE (hondaCity:CarModel {name: 'Honda City', price: 1500000, type: 'Sedan'})

CREATE (skoda:CarModel {name: 'Skoda', price: 2000000, type: 'Sedan'})

CREATE (creta:CarModel {name: 'Creta', price: 1800000, type: 'SUV'})

CREATE (swift:CarModel {name: 'Swift', price: 700000, type: 'Hatchback'})

CREATE (ertiga:CarModel {name: 'Ertiga', price: 1100000, type: 'MPV'})

// Create Showroom Sections

CREATE (section1:ShowroomSection {section\_name: 'Sedan Section', location: 'A1'})

CREATE (section2:ShowroomSection {section\_name: 'SUV Section', location: 'B1'})

CREATE (section3:ShowroomSection {section\_name: 'Hatchback Section', location: 'C1'})

CREATE (section4:ShowroomSection {section\_name: 'MPV Section', location: 'D1'})

// Create Sales Staff



Edit with WPS Office

```
CREATE (staff1:SalesStaff {name: 'Mr. Narayan', position: 'Sales Manager', experience_level: 'High'})
```

```
CREATE (staff2:SalesStaff {name: 'Ms. Sita', position: 'Sales Executive', experience_level: 'Medium'})
```

```
// Create Relationships between Car Models and Showroom Sections
```

```
CREATE (hondaCity)-[:AVAILABLE_IN]->(section1)
```

```
CREATE (skoda)-[:AVAILABLE_IN]->(section1)
```

```
CREATE (creta)-[:AVAILABLE_IN]->(section2)
```

```
CREATE (swift)-[:AVAILABLE_IN]->(section3)
```

```
CREATE (ertiga)-[:AVAILABLE_IN]->(section4)
```

```
// Create Relationships between Sales Staff and Showroom Sections
```

```
CREATE (staff1)-[:HANDLED_BY]->(section1)
```

```
CREATE (staff1)-[:HANDLED_BY]->(section2)
```

```
CREATE (staff2)-[:HANDLED_BY]->(section3)
```

```
CREATE (staff2)-[:HANDLED_BY]->(section4)
```

```
// Create Customer
```

```
CREATE (customer1:Customer {name: 'John Doe', email: 'john@example.com', phone_number: '9876543210', city: 'Pune'})
```

```
CREATE (customer2:Customer {name: 'Alice Smith', email: 'alice@example.com', phone_number: '9876543200', city: 'Mumbai'})
```

```
// Create Enquiries
```

```
CREATE (enquiry1:Enquiry {enquiry_date: '2023-01-15', status: 'Resolved'})
```

```
CREATE (enquiry2:Enquiry {enquiry_date: '2023-02-20', status: 'Pending'})
```

```
// Create Purchases
```

```
CREATE (purchase1:Purchase {purchase_date: '2023-02-01', amount: 1800000})
```



```
CREATE (purchase2:Purchase {purchase_date: '2023-03-10', amount: 1500000})
```

```
// Create Relationships between Customers, Enquiries, and Purchases
```

```
CREATE (customer1)-[:MADE_ENQUIRY]->(enquiry1)
```

```
CREATE (customer2)-[:MADE_ENQUIRY]->(enquiry2)
```

```
CREATE (customer1)-[:MADE_PURCHASE]->(purchase1)
```

```
CREATE (customer2)-[:MADE_PURCHASE]->(purchase2)
```

```
// Relationship between Enquiry and Purchase
```

```
CREATE (enquiry1)-[:RESULTED_IN]->(purchase1)
```

a)

```
MATCH (c:CarModel)
```

```
RETURN c.name AS car_model, c.type AS car_type
```

b)

```
MATCH (staff:SalesStaff {name: 'Mr. Narayan'})-[:HANDLED_BY]->(section:ShowroomSection)
```

```
RETURN section.section_name AS section_name
```

c)

```
MATCH (customer:Customer)-[:MADE_ENQUIRY]->(enquiry:Enquiry)
```

```
WHERE NOT (customer)-[:MADE_PURCHASE]->(Purchase)
```

```
RETURN customer.name AS customer_name
```

d)



```
MATCH (customer:Customer)-[:MADE_PURCHASE]->(purchase:Purchase)-[:RESULTED_IN]
->(car:CarModel)

WITH car, count(purchase) AS sales_count

ORDER BY sales_count DESC

LIMIT 1

RETURN car.name AS car_model, sales_count
```

Slip 23

1)

2)

// Create Books

```
CREATE (textBook:Book {title: 'Advanced Programming', author: 'John Doe', type: 'Text', ISBN:
'12345', price: 500})
```

```
CREATE (refBook:Book {title: 'Database Systems', author: 'Jane Smith', type: 'Reference', ISBN:
'67890', price: 700})
```

```
CREATE (bibBook:Book {title: 'History of Science', author: 'Albert Einstein', type: 'Bibliography',
ISBN: '11223', price: 300})
```

// Create Students

```
CREATE (student1:Student {name: 'Alice', student_id: 'S001', email: 'alice@example.com',
contact_number: '9876543210'})
```

```
CREATE (student2:Student {name: 'Bob', student_id: 'S002', email: 'bob@example.com',
contact_number: '9876543200'})
```



Edit with WPS Office



```
CREATE (student3:Student {name: 'Charlie', student_id: 'S003', email: 'charlie@example.com',  
contact_number: '9876543199'})
```

```
// Create Purchase relationships
```

```
CREATE (student1)-[:BOUGHT]->(textBook)
```

```
CREATE (student1)-[:BOUGHT]->(refBook)
```

```
CREATE (student2)-[:BOUGHT]->(bibBook)
```

```
CREATE (student3)-[:BOUGHT]->(textBook)
```

```
CREATE (student3)-[:BOUGHT]->(refBook)
```

```
// Create Recommendations and Ratings
```

```
CREATE (rec1:Recommendation {rating: 5, comment: 'Great book!'})
```

```
CREATE (rec2:Recommendation {rating: 4, comment: 'Very informative'})
```

```
CREATE (student1)-[:RECOMMENDED]->(textBook)
```

```
CREATE (student1)-[:RATED]->(rec1)
```

```
CREATE (student2)-[:RECOMMENDED]->(refBook)
```

```
CREATE (student2)-[:RATED]->(rec2)
```

```
CREATE (student3)-[:RECOMMENDED]->(bibBook)
```

```
CREATE (student3)-[:RATED]->(rec2)
```

a)

```
MATCH (b:Book)
```

```
WHERE b.type = 'Text'
```

```
RETURN b.title AS Book_Title, b.author AS Author
```

b)



```
MATCH (s:Student)-[:BOUGHT]->(b:Book)
WHERE b.type IN ['Text', 'Reference']
WITH s, COLLECT(b.type) AS book_types
WHERE ALL(type IN ['Text', 'Reference'] WHERE type IN book_types)
RETURN s.name AS Student_Name
```

c)

```
MATCH (s:Student)-[:RECOMMENDED]->(b:Book)
WITH b.type AS book_type, COUNT(*) AS recommendation_count
ORDER BY recommendation_count DESC
LIMIT 1
RETURN book_type, recommendation_count
```

d)

```
MATCH (s:Student)-[:BOUGHT]->(b:Book)
WITH s, COLLECT(DISTINCT b.type) AS book_types
WHERE SIZE(book_types) > 1
RETURN s.name AS Student_Name
```



Slip 24

1)

2)

// Create Departments

CREATE (phy:Department {name: 'Physics', location: 'Building A'})

CREATE (geo:Department {name: 'Geography', location: 'Building B'})

CREATE (comp:Department {name: 'Computer Science', location: 'Building C'})

// Create Courses

CREATE (course1:Course {course\_name: 'Mechanics', course\_code: 'PHY101', duration: '3 months'})

CREATE (course2:Course {course\_name: 'Geography of Earth', course\_code: 'GEO102', duration: '4 months'})

CREATE (course3:Course {course\_name: 'Data Structures', course\_code: 'CS103', duration: '6 months'})



```
// Create Recommendations
```

```
CREATE (rec1:Recommendation {rating: 5, comment: 'Excellent course!'})
```

```
CREATE (rec2:Recommendation {rating: 4, comment: 'Good course with practical applications'})
```

```
CREATE (rec3:Recommendation {rating: 5, comment: 'Very informative and well structured'})
```

```
// Create Relationships
```

```
CREATE (phy)-[:CONDUCTS]->(course1)
```

```
CREATE (geo)-[:CONDUCTS]->(course2)
```

```
CREATE (comp)-[:CONDUCTS]->(course3)
```

```
CREATE (phy)-[:CONDUCTS]->(course2) // Physics department also conducts Geography course
```

```
CREATE (geo)-[:CONDUCTS]->(course3) // Geography department also conducts Data Structures course
```

```
// Create Recommendations for Courses
```

```
CREATE (course1)-[:HAS_RECOMMENDATION]->(rec1)
```

```
CREATE (course2)-[:HAS_RECOMMENDATION]->(rec2)
```

```
CREATE (course3)-[:HAS_RECOMMENDATION]->(rec3)
```

a)

```
MATCH (d:Department)
```

```
RETURN d.name AS Department_Name, d.location AS Location
```

b)

```
MATCH (d:Department)-[:CONDUCTS]->(c:Course)
```

```
WHERE d.name = 'Physics'
```

```
RETURN c.course_name AS Course_Name, c.course_code AS Course_Code
```



c)

```
MATCH (d:Department)-[:CONDUCTS]->(c:Course)-[:HAS_RECOMMENDATION]
->(r:Recommendation)
```

```
WHERE d.name = 'Geography'
```

```
WITH c, AVG(r.rating) AS avg_rating
```

```
ORDER BY avg_rating DESC
```

```
LIMIT 1
```

```
RETURN c.course_name AS Most_Recommended_Course, avg_rating
```

d)

```
MATCH (d1:Department)-[:CONDUCTS]->(c:Course)<-[:CONDUCTS]-(d2:Department)
```

```
WHERE d1.name = 'Mathematics' AND d2.name = 'Computer Science'
```

```
RETURN c.course_name AS Common_Course
```

