Samajh gaya — tumhe **total 45 questions** chahiye, jisme:

* **Subquery** → 15 questions (5 Beginner, 5 Intermediate, 5 Expert)
* **Joins** → 15 questions (5 Beginner, 5 Intermediate, 5 Expert)
* **Index** → 15 questions (5 Beginner, 5 Intermediate, 5 Expert)

Main tumhare pehle wale **EmployeeDetails** type ke dataset ko base maan ke banaunga, taki practice realistic lage.  
Lo, main ekdum category-wise de raha hoon:

**1. SUBQUERY QUESTIONS**

**Beginner (5)**

1. Get the employees whose Department matches the Department of EmployeeID = 102.
2. Find all employees whose salary is higher than the average salary of all employees.
3. Retrieve employees who work in a project whose name is 'ProjectX'.
4. Get all employees who joined after the earliest joining date in the table.
5. Find employees whose ProjectID matches any ProjectID assigned to the 'IT' department.

**Intermediate (5)**

1. Get employees whose salary is more than the salary of EmployeeID = 105.
2. Retrieve employees working in a department where the number of employees is more than 2.
3. List employees whose DOJ is later than the DOJ of all employees in the 'HR' department.
4. Find employees working on a project which has more than 3 employees assigned to it.
5. Get employees whose department is located in the same city as the department of EmployeeID = 110 (assume Department table with locations).

**Expert (5)**

1. Find employees whose DOJ is among the top 3 most recent joining dates.
2. Retrieve employees whose salary is above the average salary of their department.
3. Get employees working in a project where no employees from 'Marketing' department are assigned.
4. List employees whose ProjectID appears in more than one department.
5. Get employees whose DOJ is earlier than the DOJ of at least one employee in 'IT' but later than all in 'HR'.

**2. JOINS QUESTIONS**

**Beginner (5)**

1. Display EmployeeName along with DepartmentName by joining Employee and Department tables.
2. List EmployeeName and ProjectName for all employees assigned to projects.
3. Show EmployeeName and Department location by joining Employee, Department tables.
4. Get all employees and their project names (include employees with no project assigned).
5. Show ProjectName and the count of employees assigned to each project.

**Intermediate (5)**

1. List employees who are working in the same department as 'Amit Sharma'.
2. Display all employees along with their department name and project name (3-table join).
3. Find employees who are working on projects that ended before '2024-01-01'.
4. List employees and their project names, but only those in departments located in 'Mumbai'.
5. Get all project names and department names, even if no employees are assigned.

**Expert (5)**

1. Find employees who share the same ProjectID but are in different departments.
2. List all departments and their employees, including departments with zero employees.
3. Show employees who are assigned to multiple projects (assume EmployeeProjects table).
4. Get employees and their project names where the project has employees from more than one department.
5. Display all employees who have never been assigned any project (LEFT JOIN usage).

**3. INDEX QUESTIONS**

*(Note: Index questions will be about identifying where indexes can be applied or queries where index usage is beneficial.)*

**Beginner (5)**

1. Create an index on the EmployeeName column of Employee table.
2. Create a composite index on Department and ProjectID.
3. Create an index on DOJ to speed up date range queries.
4. Drop an index created on EmployeeName.
5. Show how to view all indexes created on a table.

**Intermediate (5)**

1. Write a query to retrieve all employees who joined between '2023-01-01' and '2023-12-31' using an index on DOJ.
2. Use an index to speed up searching employees by ProjectID.
3. Create a UNIQUE index on ProjectID in Project table.
4. Create an index to improve the performance of queries filtering by Department and DOJ.
5. Drop a composite index on Department and DOJ.

**Expert (5)**

1. Create a filtered index for employees in the 'IT' department.
2. Write a query to find employees in 'HR' whose DOJ is in 2023, optimized using an index.
3. Create a clustered index on EmployeeID (if supported by DB).
4. Create a non-clustered index on EmployeeName and demonstrate its effect with an execution plan.
5. Design an index strategy to improve performance for frequent joins between Employee and Project tables.