

Sr. No	Title
	<b>MongoDB</b>
1	<p>MongoDB Queries: Design and Develop MongoDB Queries using CRUD operations.</p> <p><b>Create collection “student” and</b></p> <ol style="list-style-type: none"> <li>1. Insert entry of 10 or more students.</li> <li>2. select all documents from the collection "student" which satisfying the following condition -               <ol style="list-style-type: none"> <li>a. gender of student is male and</li> <li>b. class of the student is TE A or</li> <li>c. grd_point of the student is greater than equal to 31 with comparison operator.</li> </ol> </li> <li>3. Select such a documents from the collection "student” who belongs to “Bangalore” and whose age is less than equal to 19.</li> <li>4. Update course to “MCA” from collection “student” whose first name is “ Bhakti”</li> <li>5. Delete the document from collection student where age is “18” and gender is “ male”.</li> <li>6. Delete the document from collection student where last name is Jadhav and belongs to class TE B.</li> <li>7. Make statement with nor and other arithmetic operator.</li> <li>8. Use save method</li> </ol>
2	<p>MongoDB Queries: Design and Develop MongoDB Queries using CRUD operations.</p> <p><b>Create collection “student” and</b></p> <ol style="list-style-type: none"> <li>1. <b>Insert entry of 10 or more students</b></li> <li>2. <b>select all documents from the collection "student" which satisfying the following condition -</b> <ol style="list-style-type: none"> <li>a. gender of student is female or</li> <li>b. Class of the student is TE B and</li> <li>c. grd_point of the student is greater than equal to 31 with comparison operator.</li> </ol> </li> <li>3. <b>Update age to 20 from collection “student” whose first name is “ Riyansh”</b></li> <li>4. Update or save the city to nagpur and State to MH whose last name is “<b>soniminde</b>”</li> <li>5. Students have cancelled admission from the college who is belongs to state “<b>KA</b>”.</li> <li>6. Delete the document from collection student where First name is Evanshika and age is less than 18.</li> <li>7. Make statement with nor and other arithmetic operator.</li> </ol>
3	<p>MongoDB - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.</p>

	<p>Create collection student{ Rollno ,Name, Class, Div, Subject,Marks, Address}and enter 6 entries or more. And perform the following:</p> <ol style="list-style-type: none"> <li>1. Find average of total marks in TOC.</li> <li>2. Find the number of students division wise.</li> <li>3. Find students of div B who scored min marks in DBMS.</li> <li>4. Find total sum of marks in SPOS of students staying pune</li> <li>5. Find the division wise count of students whose DBMS marks.</li> <li>6. Find the count of each city.</li> <li>7. Create the simple and compound Indexing on column.</li> </ol>
4	<p>MongoDB - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.</p> <p>Create collection student{ Rollno ,Name, Class, Div, Subject, Marks, Address} and enter 6 entries or more. And perform the following:</p> <ol style="list-style-type: none"> <li>1. Find the maximum marks of student in DS who stay in the same city</li> <li>2. Calculates the average of given marks.</li> <li>3. Inserts the any value to an array in the resulting document.</li> <li>4. Create a simple and compound index on name and class.</li> <li>5. Delete index of name.</li> <li>6. Implement aggregation pipeline using pipeline operations(example : \$sort, \$ match ,\$group etc in single statement)</li> </ol>
5	<p>Write mapreduce function on <b>students info</b> collection to find the number of students who scored more than 70 marks in DBMS as “ Highscores”,less than 70 marks but greater than 40 marks as “Average scores” and less than 40 marks as “Failed.</p> <p>And also perform any other mapreduce fuction according to you.</p>
6	<p>Create a collection City (cityname, area, population). Write a MapReduce function to find the number of cities having populationless than 10 Lakhs and cities with poulation greater than 10 Lakhs and categorize them into categories HIGHLY POPULATED AND LOW POPULATED.</p>
	<p>7. Create Database SITS using MongoDB Create following Collections Teachers(Tname,dno,dname,experience,salary,date_of_joining )</p>

	<p>Students(<sup>7</sup>name,roll_no,class)</p> <ol style="list-style-type: none"> <li>1. Display<sup>7</sup> the department wise average salary</li> <li>2. Display<sup>7</sup> the no. Of employees working in each department</li> <li>3. Display the department wise total salary of departments having total salary greater than or equals to 50000/-</li> <li>4. Write the queries using the different operators like max, min. Etc.</li> <li>5. Create unique index on any field for above given collections</li> <li>6. Create compound index on any fields for above given collections</li> <li>7. Show all the indexes created in the database SITS.</li> <li>8. Delete index.</li> </ol>
	<b>SQL statements</b>
8	<p>1Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.</p> <ol style="list-style-type: none"> <li>1. Create tables <b>Emp</b> and <b>dept</b> with constraints(Primary key, Foreign key etc)</li> <li>2. create views on any two tables using conditions.(with types of views)</li> <li>3. Insert, Update and delete values from views</li> <li>3. Create index called EmployeeId for the deptment table. Entries should be in ascending order by department id and then by employee id within each department.</li> <li>4. Create sequence on Employee id.</li> <li>5. Delete index.</li> </ol>
9	<p>Create database propertydearalers  Branch(branchno, street, city, postcode)  Staff(Staffno, fname, lname, position, Sex, DOB, salary, branchno)  PropertyforRent(propertyNo, street, city, Postcode,Type,rooms,rent,staffno,branchno)</p> <ol style="list-style-type: none"> <li>1. Insert the data as shown in the above tables</li> <li>2. Update the lname of the staff with staffno. 106 to shinde</li> <li>3. Delete the branch with postcode 411041</li> <li>4. List the employee who have at least 2 A's in their name.</li> <li>5. List the maximum salary from each branch of the staff members from highest to lowest</li> <li>6. List all properties managed by a staff member having id S123</li> <li>7. Find the average rent of properties in a Pune</li> <li>8. Find the number of properties managed by each branch</li> </ol>
10	<p>Create database propertydearalers  Branch(branchno, street, city, postcode)</p>

	<p>Staff(Staffno, fname, lname, position, Sex, DOB, salary, branchno)  PropertyforRent(propertyNo, street, city, Postcode,Type, rooms,rent,staffno, branchno)</p> <ol style="list-style-type: none"> <li>1. Insert the data as shown in the above tables</li> <li>2. Update the street of branchNo 1001 to MG</li> <li>3. List the name of staff who have salary greater than average salary of all staff.</li> <li>4. Find the type and rooms of properties in pune</li> <li>5. Find the name of staffs who work as salesman or saleswomen</li> <li>6. Find the no. of properties which are of type flat.</li> <li>7. List staff members born before 2004.</li> <li>8. Find the total number of staff members in each branch.</li> </ol>
11	<p>Create a db called company consist of the following tables.  1.Emp (eno,ename, job,hiredate,salary,commission,deptno,)  2.dept(deptno,deptname,location)  eno is primary key in emp  deptno is primary key in dept</p> <p>Solve Queries by SQL</p> <ol style="list-style-type: none"> <li>1. List the maximum salary paid to salesman</li> <li>2. List name of emp whose name start with 'I'</li> <li>3. List details of emp who have joined before '30-sept-81'</li> <li>4. List the emp details in the descending order of their basic salary</li> <li>5. List of no. of emp &amp; avg salary for emp in the dept no '20'</li> <li>6. List the avg salary, minimum salary of the emp hiredate wise for dept no '10'.</li> <li>7. List emp name and its department</li> <li>8. List total salary paid to each department</li> <li>9. List details of employee working in 'Dev' department</li> <li>10. Update salary of all employees in deptno 10 by 5 %</li> </ol>
12	<p>Create the following tables. And Solve following queries by SQL</p> <ol style="list-style-type: none"> <li>1. Deposit (actno,cname,bname,amount,adate)</li> <li>2. Branch (bname,city)</li> <li>3. Customers (cname, city)</li> <li>4. Borrow (loanno, cname, bname, amount) Add primary key and foreign key wherever applicable.</li> </ol> <p>Insert data into the above created tables.</p> <ol style="list-style-type: none"> <li>a. Display names of all branches located in city Bombay.</li> <li>b. Display account no. and amount of depositors.</li> <li>c. Update the city of customers Anil from Pune to Mumbai</li> <li>d. Find the number of depositors in the bank</li> <li>e. Calculate Min,Max amount of customers.</li> <li>f. Create an index on deposit table.</li> <li>g. Create View on Borrow table.</li> </ol>
13	<p>Create the following tables:  Orders(Order_no, cust, product, Quantity, amount, Disc)</p>

	<p>Customers(Cust_No, Company, Cust_Rep, Credit_Limit) Sales_Representative (Rep_no,Name, Re_office,Quota, sales)</p> <p><b>Note: Orders (cust –foreign key for cust_no from Customer )</b></p> <p><b>Customers</b>(Cust_Rep foreign key for Rep_no from Sales_Representative )</p> <p>Write a query for following:</p> <ol style="list-style-type: none"> <li>1. List for each customer: customer name, their rep’s name, their rep’s office number.</li> <li>2. List orders over \$25,000 including the name of the salesperson who took the order and the name of the customer who placed it.</li> <li>3. Find the products which have been sold to TCS.</li> <li>4. Find company which has been offered maximum discount.</li> <li>5. Find the sales representatives who work in the same office.</li> <li>6. Find the credit limit of company and the discount it has received</li> </ol>
14	<p>Create the following tables: Orders(Order_no, cust, product, Quantity, amount, Disc) Customers(Cust_No, Company, Cust_Rep, Credit_Limit) Sales_Representative (Rep_no,Name, Re_office,Quota, sales)</p> <p><b>Note: Orders (cust –foreign key for cust_no from Customer )</b></p> <p><b>Customers</b>(Cust_Rep foreign key for Rep_no from Sales_Representative )</p> <p>Write a query for following:</p> <ol style="list-style-type: none"> <li>1. List for each customer : customer name, credit limit, rep name serving the customer and rep sales.</li> <li>2. List all orders showing number and amount, and name and credit limit of customer.</li> <li>3. Find the product wise sale amount of products which exceeds \$12000.</li> <li>4. Find the names of amount, names of customers and names of representatives who have been involved in the sale of software.</li> <li>5. Find the credit limit of company and the discount it has received</li> <li>6. Find the sales representatives who work in the same office.</li> </ol>
15	<p>Create the following tables.</p> <ol style="list-style-type: none"> <li>1) PUBLISHER( PID , PNAME ,ADDRESS ,STATE ,PHONE ,EMAILID );</li> <li>2) BOOK( ISBN ,BOOK_TITLE , CATEGORY , PRICE , COPYRIGHT_DATE , YEAR ,PAGE_COUNT</li> </ol>

	<p>,PID );</p> <p>3) AUTHOR(AID,ANAME,STATE,CITY ,ZIP,PHONE,URL )</p> <p>Solve following queries by SQL</p> <ol style="list-style-type: none"> <li>1. Retrieve city, phone, url of author whose name is 'CHETAN BHAGAT'.</li> <li>2. Retrieve book title, price, author name and url for publishers 'MEHTA'.</li> <li>3. In a PUBLISHER relation change the phone number of 'MEHTA' to 123456</li> <li>4. Calculate and display the average, maximum, minimum price of each publisher.</li> <li>5. Delete details of all books having a page count less than 100.</li> <li>6. Retrieve details of all authors residing in city Pune and whose name begins with character 'C'.</li> <li>7. Retrieve details of authors residing in same city as 'Korth'.</li> <li>8. Find books with a copyright date before 2010:</li> </ol>
16	<p>Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is <math>\leq 1500</math> and marks <math>\geq 990</math> then student will be placed in <b>distinction</b> category if marks scored are between 989 and 900 category is <b>first class</b>, if marks 899 and 825 category is <b>Higher Second Class</b></p>
17	<p>Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory.</p> <p>Suggested Problem statement: Consider Tables:</p> <ol style="list-style-type: none"> <li>1. Borrower(Roll_no, Name, DateofIssue, NameofBook, Status)</li> <li>2. Fine(Roll_no,Date,Amt)</li> </ol> <ul style="list-style-type: none"> <li>• Accept Roll_no and NameofBook from user.</li> <li>• Check the number of days (from date of issue).</li> <li>• If days are between 15 to 30 then fine amount will be Rs 5per day.</li> <li>• If no. of days <math>&gt; 30</math>, per day fine will be Rs 50 per day and for days less than 30, Rs. 5 per day.</li> <li>• After submitting the book, status will change from I to R.</li> <li>• If condition of fine is true, then details will be stored into fine table.</li> <li>• Also handles the exception by named exception handler or user define exception handler</li> </ul>
18	<p>Named PL/SQL Block: PL/SQL Stored Procedure and Stored Function.</p> <p>Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scoredby students in examination is <math>\leq 1500</math> and marks <math>\geq 990</math> then student will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 and 825 category is Higher Second Class. Write a PL/SQL block to use procedure created with above requirement. Stud_Marks(name, total_marks) Result(Roll,Name, Class)</p>
19	<p>Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor) Write a PL/SQL block of</p>

	code using parameterized Cursor that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped
20	<p>Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).</p> <p>Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table.</p>
21	<p>Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).</p> <p>Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table.</p>
23	<p>SQL Queries - all types of Join, Sub-Query and View: (Note: also examiner can ask it without join)</p> <p>Create the tables Employee(EmpID, Ename, Salary, Contactno, City, DeptID) and Department (DeptID, Dname, Location)</p> <ol style="list-style-type: none"> <li>1. Display Employee Name and Department name from given entity.</li> <li>2. Retrieve list of employees along with their department names and locations.(Using inner join or natural join)</li> <li>3. Display all employee names, location and their department name which are matching with employee table.</li> <li>4. Display all department names, salary, location and department names of employee.(Right Joins)</li> <li>5. Display list of all employees and departments (Full joins)</li> <li>6. Display all possible combinations of employees and departments (cross joins)</li> <li>7. Retrieve parts of employees who shares the same department(Self Join)</li> <li>8. Create simple and complex views</li> <li>9. Update simple view</li> <li>10. Delete the view</li> </ol>
24	Database Connectivity: Write a program to implement MySQL/Oracle database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)
25	Database Connectivity: Write a program to implement MongoDB database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)