|  |  |
| --- | --- |
|  | University of Science and Technology ChittagongDepartment of CSE, FSET |
|  | |
| Course Title: Advanced Database Management System Lab  Course Code: CSE 468 | |
|  | |
| Course Instructor: Nowshin Tasnim, Jr Lecturer, Department of CSE, USTC. | |
|  | |
|  | |
| 1. Write a SQL program to solve the given problem following integrity constraint.   WorksOn Database:  emp (eno, ename, bdate, title, salary, dno)  proj (pno, pname, budget, dno)  dept (dno, dname, mgreno)  workson (eno, pno, resp, hours)  Questions:   1. Write an SQL query that returns the project number and name for projects with a budget greater than $100,000. 2. Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is 'Manager'. 3. Write an SQL query that returns the employees (number and name only) who have a title of 'EE' or 'SA' and make more than $35,000. 4. Write an SQL query that returns the employees (name only) in department 'D1' ordered by decreasing salary. 5. Write an SQL query that returns the departments (all fields) ordered by ascending department name. 6. Write an SQL query that returns the employee name, department name, and employee title. 7. Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10. 8. Write an SQL query that returns the project name, department name, and budget for all projects with a budget < $50,000. 9. Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary. 10. 10) Write an SQL query that returns the employee name, project name, employee title, and hours for all works on records. | |
| 1. Trigger    1. Write a SQL program which automatically preserves the updated salary of an employee in a separate table.    2. Write a SQL program which automatically preserves the deleted record of a Main table in a Backup table.    3. Write a SQL program which automatically calculates the total marks and the percentage marks when student’s English, Physics, Chemistry and Mathematics marks are inserted. | |
| 1. Convert this table to MongoDB NoSQL Format.  |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Age** | **City** | **Occupation** | | John | 25 | New York | Engineer | | Emily | 30 | Los Angeles | Lawyer | | Michael | 40 | Chicago | Doctor | | Sarah | 35 | Houston | Teacher | | David | 28 | San Francisco | Software Engineer | | Jessica | 45 | Miami | Nurse | | Ryan | 32 | Seattle | Architect | | Amanda | 29 | Boston | Marketing Specialist | | Daniel | 33 | Denver | Chef | | Olivia | 27 | Atlanta | Graphic Designer | | Matthew | 38 | Dallas | Business Analyst | | Ashley | 31 | Phoenix | Accountant | | Jacob | 36 | Philadelphia | Consultant | | Elizabeth | 26 | San Diego | Journalist | | William | 42 | Austin | Entrepreneur |  1. Find all documents. 2. Find documents where Age is greater than 30. 3. Find documents where City is "New York". 4. Find documents where Occupation is "Engineer". 5. Find documents where Name starts with "J". 6. Find documents where Age is between 30 and 40. 7. Find documents where Occupation is not "Teacher". 8. Find documents where City is in a list of cities. 9. Find the document with the highest Age. 10. Find the average Age of all documents. 11. Find documents where the Name contains a specific substring. 12. Find the number of documents in each city. 13. Find documents where Age is less than or equal to 30 and Occupation is "Engineer". | |
| 1. Insert the following collection “Orders” in database “Bookstore” and perform the problems based on aggregation.   db.Orders.insertMany([ { \_id: 1, customer: { name: "Alice", country: "USA" }, items: [ { title: "MongoDB Basics", category: "Database", price: 25, qty: 2 }, { title: "Node.js in Action", category: "Web", price: 30, qty: 1 } ], order\_date: ISODate("2024-03-10") }, { \_id: 2, customer: { name: "Bob", country: "UK" }, items: [ { title: "Python 101", category: "Programming", price: 20, qty: 1 } ], order\_date: ISODate("2024-03-11") }, { \_id: 3, customer: { name: "Charlie", country: "USA" }, items: [ { title: "MongoDB Basics", category: "Database", price: 25, qty: 1 }, { title: "JavaScript Guide", category: "Web", price: 28, qty: 3 } ], order\_date: ISODate("2024-03-13") } ])   1. Find the most loyal customer. 2. Category-wise earnings. 3. Who loves "MongoDB Basics" the most? 4. Monthly Sales Summary. | |

N.B: The students should complete the experiments within their lab session. The report of the experiments should be submitted within deadline. The content of the report should be as follows:

a) Title of the Experiment

b) Objective

c) Source Code

d) Input and Output

e) Analyses of the code / Conclusion