Gujarat University
Choice Based Credit System (CBCS)
Syllabus for B. Sc. Semester IV(Computer Science)
COM 204: System Development Tools – 2 (Theory)

Hours: 4 /week **Credits: 4**

Unit: 1

1. Advanced Object Oriented Programming
   1.1. Events and Delegates
   1.2. Interfaces and Inheritance
   1.3. Namespaces
   1.4. Class hierarchy design consideration for extensibility
   1.5. Object Serialization

2. Exception Handling
   2.1. Error in programming
   2.2. Exception Handling Overview
   2.3. Structures Exception Handling
   2.4. Programmer-Defined Exception Class
   2.5. On Error Statement
   2.6. Debugging

Unit: 2

3. Multithreading
   3.1. Introduction to Thread
   3.2. Life Cycle of a Thread
   3.3. Creating Multithreaded Application
   3.4. Thread Priorities and Thread Scheduling
   3.5. Thread synchronization

4. Networking
   4.1. Introduction
   4.2. What is Socket?
   4.3. Client/Server Interaction via Stream – Socket Connections
   4.4. Datagram

Unit: 3

5. Extensible Markup Language (XML)
   5.1. Introduction
   5.2. XML Documents
   5.3. XML Namespaces
   5.4. Document Type Definitions (DTP$_s$) , Schemas and Validation

6. Introduction to Web Programming.
    6.1. Introduction to HTML
    6.2. Overview of ASP.NET
    6.3. Building a Web Application
    6.4. Building Forms with Web Controls
    6.5. Validating User Input
    6.6. Session Tracking

Unit: 4

7. ASP.NET Database Programming
    7.1. Understanding Data Binding
    7.2. Working with Data Grids

8. Introduction to Wireless Applications with ASP.NET
    8.1. ASP.NET and Services
    8.2. Introduction
    8.3. SOAP and Web Services
    8.4. Publishing and Consumming Web Services

Reference Books:-

1. ASP.NET Bible
   By MridulaParihar, et al. – IDG Books India.
2. ASP.NET Black Book
   By Kogent Solutions Inc. – Dreamtech Press
3. Building XML Web Services for ASP.NET
   By Bill Evjen – Wiley Dreamtech.

Gujarat University
Choice Based Credit System (CBCS)
Syllabus for B. Sc. Semester IV(Computer Science)
COM 205: Relational Database Management System (Theory)

Hours: 4 /week **Credits: 4**

Unit – 1

1. Introduction to DBMS
   1.1 What is database?
   1.2 Requirement of database system.
   1.3 Data models and data independence.
   1.4 DDL, DML
   1.5 Database Manager, Database Administrator.

2. Entity Relationship Models
   2.1 Entities and Entity sets
   2.2 Relationship and relationship sets
   2.3 Mapping constrains
   2.4 Primary keys
   2.5 Entity Relationship diagram and reducing it to tables.
   2.6 Generation and Specialization.
   2.7 Aggregation

3. Relational Model
   3.1 Structure of relational database.
   3.2 Relation algebra.

Unit – 2

4. Introduction to other models
   4.1 Network Model
   4.2 Hierarchical Model

5. Relational Database Design
   5.1 Functional Dependencies
   5.2 Referential Integrity
   5.3 Need for Normalization
   5.4 Normal forms
   5.5 Data Dictionary
   5.6 Tables, Table spaces & Data files, Views.

6. Crash Recovery
   6.1 Failure Classification
   6.2 Transactions

- 6.2.1 Transaction Concept, State.
- 6.2.2 Implementation of ACID rules.
- 6.2.3 Concurrent Executions.
- 6.2.4 Serializability, Recoverability.

6.3 Incremental log with differed and immediate updates.

6.4 Checkpoints

6.5 Buffer Management

6.6 Shadow Paging.

6.7 Concurrency Control.

- 6.7.1 Lock Based Protocols.
- 6.7.2 Timestamp-based Protocols.
- 6.7.3 Validation based protocols.
- 6.7.4 Deadlock Handling.

Unit – 3

7. PL/SQL

7.1 Overview of SQL

7.2 Various types of data, conventions and terminology

7.3 Retrival of information from tables. :

Making a query, SELECT command, column reordering, Use
of relational operators, use of Boolean operators, operations like IN, BETWEEN, LIKE, NULL, Not etc., Aggregate functions, COUNT, GROUP By clause, HAVING clause.

7.4 Formatting Query output:

String and expressions, ordering output by fields, multiple columns, Aggregate Group, Column number, ORDER BY, with NULL.

7.5 Querying multiple tables :

Joining tables through Referential Integrity, Equijoins and other kinds of joins of more than two tables, joining a table to itself.

Unit – 4

7.6 Subqueries :

DISTINCT with subqueries, Predicates with subqueries, Aggregate Functions in subqueries, Correlated subqueries, Correlating tables to itself, Correlated subqueries in HAVING, Correlated subqueries and joins EXISTS operator, using EXISTS with Correlated sub queries, combining EXISTS and joins, special operator ANY or SOME, ALL, UNION classes.

7.7 Entering Deleting and Changing Field Values :

DML Update command, UPDATE with multiple columns, Updating to NULL values, INSERT command using sub queries with UPDATE commands.

7.8 CREATE TABLE Command:

Indexing Altering a table, Dropping a table, Constraining a Table, Declaring Constraints, PRIMARY KEY constraint, Foreign and Parent keys, Multicolumn Foreign keys, FOREIGN KEY constraint, Foreign key restriction.

7.9 CREATE VIEW Command:

Updating views, Group views and Joins, Views and subqueries, Changing values through views, Grant command, using ALL and PUBLIC arguments, GRANT OPTION.

Recommended Reference Books:

1. HenryKroth&Silbershats, Database System Concept.
2. C.J. Date, Introduction to Database Design, Addition Wesley, Nasora.
3. Martin Gruber, Understanding SQL, BPB pub., New Delhi.
4. Ivan Bayross, SQL, PL/SQL the Programming Language of ORACLE, BPB pub., New Delhi.

Gujarat University
Choice Based Credit System (CBCS)
Syllabus for B. Sc. Semester IV (Computer Science)
COM 206: Practicals (Based on ASP.NET, SQL and PL/SQL)

Hours: 6 /week **Credits: 2.5**

❖ <u>ASP.NET Practical List</u>

1. Create and ASP.net web page using different validation controls.
2. Create and ASP.net web page that lists the customer from customers database table in a storable Data Grid with paging option. The Data Grid should display three columns, one for the customer’s ids, one for the customer’s names and one for the customer’sphone numbers. The user should be able to sort the Data Grid by customer ID.
3. Create simple web services.
4. Write a web services which return prime number of any given number.

❖ PL/SQL

- Using Variables, Constants & Data types
- User-Defined RECORD and TABLE data types.
- Assigning Database Values to variables
  SELECT………………INTO…………CURSORS.
- Using Flow Control Statements. The IF….THEN Statement, The LOOP statement WHILE loops, the GOTO statement.
- Error handling built in PL/SQL Exceptions, User – Defined Exceptions, Unhandled Exceptions.
- PL/SQL Programs
  Anonymous PL/SQL Blocks, procedures, Functions, Packages, Triggers.

❖ SQL Practical List

Create following Three Tables.

1. **Salesman**

SNUM	SNAME	CITY	COMMITION
1001	PIYUSH	LONDON	12%
1002	NIRAJ	SURAT	13%
1003	MITI	LONDON	11%
1004	RAJESH	BARODA	15%
1005	ANAND	NEW DELHI	10%
1006	RAM	PATAN	10%
1007	LAXMAN	BOMBAY	09%

SNUM : A Unique numer assign to each salesman
SNAME : The name of salesman
CITY : The location of salesman
COMMITION :The salesman commition on order

2. **Customer**

CNUM	CNAME	CITY	RATING	SNUM
2001	HARDIK	LONDON	100	1001
2002	GITA	ROME	200	1003
2003	LAXIT	SURAT	200	1002
2004	GOVIND	BOMBAY	300	1002
2005	CHANDRESH	LONDON	100	1001
2006	CHAMPAK	SURAT	300	1007
2007	PRATIK	ROME	100	1004

CNUM : A Unique number assign to each customer.
CNAME : The name of customer.
CITY : The location of customer.
RATING : A level of preference indicator given to this customer.
SNUM : A salesman number assign to this customer.

3. Order

ONUM	AMOUNT	ODATE	CNUM	SNUM
3001	18.69	10/03/99	2007	1007
3002	767.19	10/03/99	2001	1001
3003	1900.10	10/03/99	2007	1004
3004	5160.45	10/03/99	2003	1002
3005	1098.25	10/04/99	2007	1007
3006	1713.12	10/04/99	2002	1003
3007	75.75	10/05/99	2004	1002
3008	4723.00	10/05/99	2006	1001
3009	1309.95	10/05/99	2004	1002
3010	9898.87	10/06/99	2006	1001

ONUM : A Unique number assign to each Order.
AMOUNT : Amount of order in Rs.
ODATE : The date of order.
CNUM : The number of customer making the order.
SNUM : The number of salesman credited with the sale.

Solve following request with the help of SQL query.

1. Produce the order no, a mount and date of all orders.
2. Give all the information about all the customers with salesman number 1001
3. Display the information in the sequence of city, sname, snum, and Commission.
4. List of rating followed by the name of each customer in Surat.
5. List of all orders for more than Rs. 1000.
6. List out names and cities of all salesmen in London with commission above 10%
7. List all customers excluding those with rating <=100 or they are located in Rome.
8. List all order for more than Rs.1000 except the orders of snum, 1006 of 10/03/99.
9. List all orders taken on 10th march or april 1999.
10. List all customers whose names begins with a letter ‘C’
11. List all customers whose names begins with letter ‘A’ to ‘G’
12. List all order with zero or NULL amount.
13. Find out the largest orders of salesman 1002 and 1007.
14. Count all orders of 10-Mar-99.
15. Calculate the total amount ordered.
16. Calculate the average amount ordered.
17. Count the no. of salesmen currently having order.
18. Find the largest order taken by each salesman on each date.
19. Find the largest order taken by each salasman on 10/03/1999.
20. Count the no. of different non NULL cities in the customer table.
21. Find out each customer’s smallest order.
22. Find out the customer in alphabetical order whose name begins with ‘G’

23. Count the no. of salesmen registering orders for each day.
24. List all salesmen with their % of commission.
25. Display the no. of order for each day in the following format. FOR dd-mon-yy,there are orders.
26. Assume each salesperson has a 12% commission. Write a query on the order table that will produce the order number, salesman no and amount of commission for that order.
27. Find the highest rating in each city.
28. List all customers in descending order of rating.
29. Calculate the total of orders for each day.
30. Show the name of all customers with their salesman's name.
31. List all customers and salesmen who shared a same city.
32. List all orders with the names of their customer and salesman.
33. List all orders by the customers not located in the same city as their salesman.
34. List all customers serviced by salesman with commission above 12%.
35. Calculate the amount of the salesman commission on each order by customer with rating above 100.
36. List all customers located in cities where salesman Niraj has customers.
37. List all salesmen who are living in the same rating with our duplication.
38. Produce the name and city of all the customers with the same city as Hardik.
39. Extract all orders of Miti.
40. Extract all orders of Baroda's salesmen.
41. Find all orders of the salesmen who services 'Hardik'.
42. List all orders that are greater than the average of April 10, 1999.
43. Find all orders attributed to salesmen in 'London'.
44. List the commission of all salesmen serving customers in 'London'.
45. Find all customers whose cnum is 1000 above than the snum of Niraj.
46. Count the no. of customers with the rating above than the average of 'Surat'.
47. List all orders of the customer 'Chandresh'.
48. Produce the name and rating of all customers who have above average orders.
49. Find all customers with orders on 10th March, 1999 using correlate sub query.
50. List the name and number of all salesmen who has more than zero customers.
51. Calculate the total amount ordered on each day eliminating the days where the total amount was not at least Rs. 2000 above the maximum amount of that day.
52. Using correlated sub query find the name and number of all customer with rating equal to maximum for their city.
53. Select the name and number of all salesmen who have customers their cities.
54. Find all salesmen who have customers with rating > 300.
55. List all salesmen with customers located in their cities.
56. Find all salesmen for whom there are customers that follow them alphabetical order.
57. Find all customers having rating greater than any customer in 'Rme'.
58. List all orders that has amount greater than at least one of the orders from 10th June, 1999.
59. Find all orders with amounts smaller than any amount for a customer in 'Rome'.
60. Find all the customers who have greater rating than every customer in 'Rome'.
61. Select all customers whose rating doesn,t match with any rating customer of 'Surat'.

62. List all customers whose ratings are equal to or greater than ANY ‘Niraj’.
63. Find out which salesmen produce largest and smallest orders on each date.
64. Create a union of two queries that shows the names, cities and ratings of all customers. Those with rating of >=200 should display ‘HIGH RATING’ and those with < 200 should display ‘LOW RATIN’
65. Insert a row into salesmen table with the values snum is 100 salesmen name is Rakesh, city is unknown and commission is 14%.
66. Insert a row in to customer table with values London, Pratik a 2005 for the columns city, name and number.
67. Create another table London staff having same structure as salesmen table.
68. Insert all the rows of salesmen table with city London in the London staff table.
69. Create another table Day totals with two attributes date and total and insert rows into this table from order table.
70. Create a duplicate of the salesmen table with a name Multicust. Now delet all the rows from the salesmen table.
71. Get back all the rows of salesmen table from its duplicate table.
72. Remove all orders from customer chandresh from the orders table.
73. Set the ratings of all the customers of Piyush to 400
74. Increase the rating of all customers in Rome by 100
75. Salesmen Miti has resigned. Reassign her number to a new salesmen Gopal whose city is Bombay and commission is 10%
76. Double the commission of all salesmen of London.
77. Set ratings for all customers in London to NULL.
78. Suppose we have a table called sales Manager with the same definition as Salesmen table. Company decides to promote salesmen having total order more than 5000 to sales Manager.Fill up the sales Managaer table.
79. Assume that we have a table called smcity. Store the information of all salesmen with the customers in their home cities into smcity.
80. Create a table Bonus that contains date wise maximum amount of order for all salesmen.
81. Create a table Multicust containing the salesmen with more than one customer.
82. New Delhi office has closed. Remove all customers assigned to salesmen in New Delhi.
83. Delete all salesmen who have at least one customer with a rating of 100 from salesmen table.
84. Delete the salesmen who produce the lowest order for each day.
85. Find the smallest order for each day. Reduce the commission of all salesmen by 2% who produce this order.
86. Delete all customers with no current orders.
87. Write a command to find out the orders by date.
88. Write a command to add the item-name column to the order table.
89. Create a copy of your order table. Drop the original order table.
90. Write a command to create the order table so that all onum values as well as all combinations of cnum and snum are different from one another and so that NULL values are excluded from the date field.
91. Write a command to create the salesmen table so that default commission is 10% with no NULL permitted, snum is the primary key and all names contain alphabets only.

92. Give the commands to create our sample tables (salesmen, customer, orders) with all the necessary constraints like PRIMARY KEY , NOT NULL UNIQUE, FOREING KEY.
93. Create a view called Big orders which stores all orders larger than Rs. 4000.
94. Create a view Rate count that gives the count of no. of customers at each rating.
95. Create a view that shows all the customers who have the highest ratings.
96. Create a view that shows all the number of salesmen in each city.
97. Create a view that shows all the number of salesmen in each city.
98. Create a view that shows the average and total orders for each
    Salesmen after his name and number.
99. Create a view that shows all the salesmen with multiple customers
100. Create a view to keep track of the total no of customers ordering, no of salesmen taking orders, the no of orders, the average amount ordered, and the total amount ordered for each day.
101. Create a view show name that shows for each order the order no, amount, salesmen name and the customer name.
102. List all orders of salesmen ‘Rajesh’ using show name view along with his commission.
103. Create a view Max sales to store the name and number of salesmen, along with the date, who have the highest order on any given date.
104. Using above view, find out the name and number of salesmen who have the highest order at least two times. Store the result in another view.
105. Create a view same city that shows the no and name and city of the customers along with the city of the salesmen serving them.
106. Create a view commission of salesmen table to include only snum and commission field so that through. This view someone can enter or change the commission but only to values between 10% and 20%.
107. Assume that the CURDATE is a constant representing current date. Give a command to create orders table with CURDATE as a default date.
108. List all salesmen in London who had at least one customer located there as well.
109. List all salesmen in London who didn’t have any customer there.

PL/SQL Programs

1. Write a PL/SQL block to find the given numbers is ODD or EVEN.
2. Write a PL/SQL block to find the factorial of given number.
3. Write a PL/SQL block to find the maximum number from given three numbers.
4. Write a PL/SQL block to find the sum of first 100 natural Nos.
5. Write a PL/SQL block to display the pyramid using any character.
6. Make a block to insert the row in table. (CREATE student TABLE with all possible fields.)
7. Write a PL/SQL block to display the information of given student on following table.
   (using variable) Stud(sno,sname,address,city)
8. Write a PL/SQL block to display the information of given student on following table.
   (using record type variable)Student(sno,sname,address,city)

9. Create a block which delete the records of student whose total marks is less than 200 out of 500. Deletion is done after confirmation from user.

10 Retrieve the details of the employees on the basis of the department name. Write exception handler if no row are retrieves; otherwise display the employee getting the maximum salary.

11 Give an increment of 15% to those who have joined in any year but the month of current date.

12 Write PL/SQL block that find the area of patient ‘Sector 15’. If the patient count < 5 in the area the exception is raise.

13 Write a PL/SQL block for preparing a Net salary of given employee on following table.
EMP(eno,ename,address,city)
SALARY(eno,basic,da,hra,it)
NET_SALARY(eno,total_allowance,total_deduction,netpay)
Notes : DA = 59% of basic, H.R.A. = 500, I.T. = 2% of basic
    a. Total_Allowance = basic + D.A. + H.R.A.
    b. Total_Deduction = I.T.
    c. Netpay = Total_Allowance – Total_Deduction

Procedure Programs

1. Write a procedure to check the given year is leap or not.
2. Write a procedure to display the following type of Multiplication Table as per given number.

```
5 * 1 = 5
5 * 2 = 10
 “    “ =  “
 “    “ =  “
5 * 10 = 50
```

3. Write a procedure to display this kind of output on screen.

```
1
2 3
3 4 5
4 5 6 7
5 6 7 8 9
. . . . . . . .  90 91
```

4. Write a procedure to convert given octal number to decimal number.
5. Write a procedure that take Employee Number and return all the information related to the employee. Display the following format using table – Emp(eno,ename,city,salary)

Employee Number	Employee Name	City	Salary

6. Create a procedure to generate an Invoice with the following information. Invoice number, customer name, qty, amount, total bill and date of invoice etc.
7. Create a procedure to convert the entire invoice from unpaid to paid for given customer no. For a given customer check his total pending amount if it equal to the amount if it equal to the amount given in the parameter. Convert the unpaid to pay the parameter also generate the error message.
8. Examine the following rule in each sub:
   - ❖ Passing marks in each subject = 40%
   - ❖ Fail in 1 sub = Grace
   - ❖ Fail in 2 sub = ATKT
   - ❖ Fail in >= 3 sub = Fail
   - ❖ If pass in all subject then
     - ✓ >=70% = Division
     - ✓ >=60 and >70 =”First”
     - ✓ <60 = “Second”

   Write a PL/SQL procedure to find out the status and update the field status.

Function Programs

1. Write a function to check whether the given number is prime or not.
2. Write a function to find the sum of digits of accepted no.
3. Write a function to display first 25 Fibonacci nos.
4. Write a function to display the reverse string of a given string.
5. Write a function that take Employee Number and return the salary on following table. Emp(eno,ename,city,salary)
6. Write a function to count the total number of student having ‘PASS’ on following table. Student(sno,sname,grade)
7. Write a function to assign the grade to all the student using following table.
   Stud(sno,sname,sub1,sub2,sub3,grade)
   Note :
   i. if percentage >= 70 then ‘Distinction’ else
   ii. if percentage >= 60 then ‘First’ else
   iii. if percentage >= 50 then ‘Second’ else
   iv. otherwise ‘Fail’.

Package Programs

1. Write a package that has a function that check the given string is palindrome or not.
2. Write a package that has a procedure to find 1 + 1/2 + 1/3 + …….. + 1/n.
3. Write a package that has a function to check given number is not negative and a procedure to convert the given number into word. For example 25 = Twenty Five.

4. Write a package that has two object on following table.
   I. The function is used to calculate the Net Salary.
   II. The procedure is used to display the Pay Slip in following format.
   Emp(eno,ename,basic)
   Salary(eno,da,hra,it,gross_sal,net_sal)
   Notes : D.A. = 59% of basic, H.R.A. = 500, I.T. = 2% of basic
   Gross_salary = Basic + D.A. + H.R.A. , Net_salary = Gross_salary – I.T.
5. Create a package,which contains following functions and procedure:
   - Function, which return the department having maximum employees.
   - Procedure, which gives the details of managers of each department.
   - Procedure, which prepare the report of employee overtime when employee id is passed.
6. Create one package which stores the procedures of insert, delete & display of the following table. Write one function which return the Names of employee whose birth date is between the two passing date.

   Table : employee master
   ===== ====================
   Empcd, Emp name, Emp address, Birth date, Salary

Trigger Programs

1. Write a trigger on insert to convert the name into capital letters.
2. Write a trigger to check the pin code is exactly six digits or not.
3. Write a trigger to check the mark is not zero or negative.
4. Write a trigger that check the student_id must be start with ‘M’.
5. Create trigger that is fired before the DML statement’s execution on the employee table. The trigger checks the day based on sysdate. If the day is Sunday, the trigger does not allow the DML statement’s execution and raises an exception. Write appropriate message in the exception section.
6. Write a trigger that is fired after an insert statement is executed for the student table. The trigger writes the new student’s ID, name and sysdate in a new table student_info.

****