PG/IIS/COS/205/25(Pr.)

M.Sc. 2nd Semester Examination, 2025

COMPUTER SCIENCE

(Practical)

PAPER – COS-205

Full Marks : 50

Time : 4 hours

Answer all questions

The figures in the right hand margin indicate marks

Candidates are required to give their answers in their own words as far as practicable

COS-205(M_1)

(*DBMS Lab*)

Answer any one question selected by lottery

20 × 1

1. Create the table described below with the constraints/attributes specified :

Table name : STAFF_XX (XX → Last two digits of your class roll number)
Descriptions : Used to store staff information.

Column Name	Data Type	Size	Constraints/Attributes
StaffID	Number	4	Primary key, values between 5000 and 5999
StaffName	Varchar2	25	Not null, Name must be in upper case
DeptCode	Number	2	
Position	Varchar2	20	Not null
Supervisor	Number	4	Foreign key references StaffID of STAFF_XX, values between 5000 and 5999
JoinDate	Date		Not null
Pay	Number	6	Default 10000

Perform the following queries :

(*a*) Display all the distinct positions.

(*Continued*)

(*b*) Show the names of staff members who joined in the year 2022.

(*c*) Find the staff member with the lowest salary.

2. Relation Schema :

Course(CourseID, CourseName, Credits, Department)
Instructor(InstructorID, InstructorName, Qualification)
Teaches(InstructorID, CourseID, Year)

Questions :

(*a*) Create the above database using SQL.

(*b*) List all courses and their corresponding departments.

(*c*) Find the course(s) with the highest credit.

(*d*) Display all instructors who taught courses in the current year.

(*e*) List names of instructors teaching more than 3 courses.

3. Relation Schema :

Vehicle(VehicleID, Model, Manufacturer, YearOfMake)
Customer(CustID, CustName, ContactNo)
Purchases(CustID, VehicleID, PurchaseDate)

Questions :

(*a*) Create the above database using SQL.

(*b*) Show the total number of vehicles purchased by each customer.

(*c*) List all vehicle models made after 2015.

(*d*) Display customer names starting with the letter 'A'.

(*e*) List customers who have purchased all vehicles manufactured by "Honda".

(Continued)

4. Relation Schema :

Invoice(InvoiceID, ClientID, InvoiceDate, Amount)
Client(ClientID, ClientName, City)
InvoiceItem(InvoiceID, ProductID, Quantity, Rate)

Questions :

(*a*) Create the above database using SQL.

(*b*) Show clients who received invoices in the last 15 days.

(*c*) List client names with their total invoice amounts in descending order.

(*d*) Add a constraint to make sure client names always start with an uppercase letter.

(*e*) List all clients with their invoice details, sorted by invoice date.

5. Relation Schema :

Developer(DevID, DevName, Age, Location)
Application(AppID, AppName, LaunchDate)
WorksOn(DevID, AppID, Responsibility)

Questions :

(*a*) Create the above database using SQL.

(*b*) Show developers and the applications they work on.

(*c*) Find developers assigned to more than 2 applications.

(*d*) Find developers working on an application launched in the same city they reside in.

(*e*) List all applications along with the count of developers working on each.

6. Relation Schema :

Retailer(RetailerName, ItemID)
Inventory(ItemID, ItemName, Category, InStock)
Buyer(BuyerName, Area, City)
PurchaseOrder(OrderID, BuyerName, ItemID, Quantity)

Questions :

(*a*) **Create the above database using SQL.**

(*b*) **Count the number of buyers who made purchases.**

(*c*) **Find buyers who purchased but never supplied any item.**

(*d*) **Identify buyers who are also retailers, including their city.**

(*e*) **List buyers sorted by quantity of items purchased.**

7. Relation Schema :

Item(ItemID, ItemName, Category, UnitPrice, StockQty)
Depot(DepotId, Region)
StockStatus(DeptID, ItemID, QtyAvailable)

Questions :

(*a*) Create the above database using SQL.

(*b*) Display regions with the total quantity of items available

(*c*) List items in category 'Furniture' with stock above 50.

(*d*) Identify items with quantity less than reorder level (assume it).

(*e*) Show all items added within the last 3 months sorted by name.

8. Write a PL/SQL program that accepts an employee's basic salary as input and uses a CASE statement to compute the following :

House Rent Allowance (HRA) :

- If salary < 10000 → HRA = 20% of basic
- If salary between 10000 and 25000 → HRA = 25% of basic
- If salary > 25000 → HRA = 30% of basic

Dearness Allowance (DA) :

- If salary < 10000 → DA = 80% of basic
- If salary between 10000 and 25000 → DA = 90% of basic
- If salary > 25000 → DA = 95% of basic

Display :

(*a*) Basic Salary
(*b*) HRA
(*c*) DA
(*d*) Gross Salary = Basic + HRA + DA

If the input salary is negative or zero, show an appropriate error message.

9. Write a PL/SQL program that accepts a number from the user. If the number is prime, display "Prime number", otherwise display "Not a prime number". Also, handle the case when the input is less than 1.

Viva – 3
PNB – 2

COS-205(M_2)

(*Compiler Lab*)

Answer any one from the following : 20×1

1. Write a Lex/Yacc program to count the number of vowels and consonants in a given sentence.

2. Write a Lex/Yacc program to count the number of integers and floating-point numbers in a given input.

3. Write a Lex/Yacc program to count the number of variables or identifier in a given program.

4. Write a Lex/Yacc program to count the number of lines, words characters in a given string.

5. Write a Lex/Yacc program to validate email addresses and print whether they are valid or not according to general format rules.

6. Write a Lex/Yacc program to prepend line numbers to each line of a given file.

7. Write a Lex/Yacc program to delete the comments in a given file.

8. Write a Lex/Yacc program to delete blank lines and then count the number of lines of a given file.

9. Write a Lex/Yacc program to check whether a given arithmetic statement is valid or not.

10. Write a Lex/Yacc program to implement a simple desk calculator.

Viva – 3
PNB – 2