Answer all Questions   **Time: 02 Hours**

Question 01

(a) Create the following table in *SQL* and answer the *queries* given below:

Student No	Class	Name	Game1	Grade1	Game2	Grade2
10	7	Sammer	Cricket	B	Swimming	A
11	8	Sujit	Tennis	A	Scating	C
12	7	Kamal	Swimming	B	Football	B
13	7	Venna	Tennis	C	Tennis	A
14	9	Archana	Basketball	A	Cricket	A
15	10	Arpit	Cricket	A	Athletics	C

(i) Display the names of the students who are getting 'C' in either Game1 or Game2 or both.

(ii) Display the number of students getting grade 'A' in Cricket.

(iii) Display the names of the students who have same game for both Game1 and Game2.

(iv) Display the games taken up by the students, whose name starts with 'A'.

(v) Add a new column named 'Marks'.

(vi) Assign a value 200 for Marks for all those who are getting grade 'B' or grade 'A' in both Game1 and Game2.

(vii) Arrange the whole table in the alphabetical order of Name.

(b) Create the fallowing table in *SQL* and answer the *queries* given below:

Student No	Class	Name	Game	Grade1	SUPW	Grade2
10	7	Sammer	Cricket	B	photography	A
11	8	Sujit	Tennis	A	gardening	C
12	7	Kamal	Swimming	B	photography	B
13	7	Venna	Tennis	C	Cooking	A
14	9	Archana	Basketball	A	Literature	A
15	10	Arpit	Cricket	A	gardening	C

(i)   Display the names of the students who have grade 'C' in either Game or *SUPW*.

(II)   Dislplay the number of students getting grade 'B' in Cricket.

(III)   Dislplay the Different games offered in the school.

(iv)   Dislplay the *SUPW* taken up by the students, whose names starts with 'A'.

(v)   Add a new column named 'scores'.

(vi)   Assign a value 100 for scores for all those who are getting grade 'B' or above in Game.

(vii)   Arrange the whole table in the descending order of Class.

Question 02

Create the fallowing two tables EM, DEPT. Add records to the tables and answer the fallowing queries using the basic *SQL SELECT* statements.
Execute these queries on *SQL 7.0*

EMP (Employee Table)

COL NAME	TYPE	SIZE	NULL	DESCRIPTION
EMPNO	CHAR	6	no	Employee number, unique
FIRSTNAME	VARCHAR	12	no	First name
MIDINIT	CHAR	1	no	Middle initial
LASTNAME	VARCHAR	15	no	Last name
WORKDEPT	CHAR	3		Employee's dept number
PHONENO	CHAR	4		Employee's telephone number
HIREDATE	DATE			Date hired
JOB	CHAR	8		Job held by employee
EDLEVEL	NUMBER	2		No. of years of formal educ.
SEX	CHAR	1		M=male, F=female
BIRTHDATE	DATE			Date of birth
SALARY	NUMBER	(8,2)		Annual salary
BONUS	NUMBER	(8,2)		Annual bonus
COMM	NUMBER	(8,2)		Annual commission

DEPT (Department Table)

COL NAME	TYPE	SIZE	NULL	DESCRIPTION
DEPTNO	CHAR	3	no	Department number, unique
DEPTNAME	VARCHAR	36	no	Department name
MGRNO	CHAR	6		Dept manager's employee no.
ADMRDEPT	CHAR	3	no	ID of administrative dept
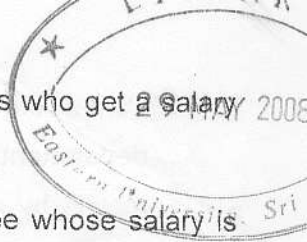
i) Display the last name, work department and salary of all employees who get a salary of $40,000 or more per year.

ii) Display the last name, first name and birth date of every employee whose salary is less than $40,000 per year. Sort the result table by last name and first name.

iii) Show the information of all departments whose administrative department has 'A00' as department number.

iv) Show the employee number, last name and work department of employees whose work department number is between 'D11' and 'E21' (inclusive).

v) Produce a list of all employees whose work department number begins with an 'E'. Show the employee number, first name, and last name.

vi) Produce a list of employees who work in department numbers 'B01', 'C01', 'D11' and 'E21', showing last name, department number and monthly salary. List output in ascending order of department number and descending order of monthly salary within each department.

vii) Display an output table containing the information of department number 'E01' as well as the information of all departments that have 'E01' as the administrative department.

viii) Display last name and first name of all employees who work in the same department as an employee with the lastname of ADAMSON. Write two SQL statements for this query, the first using a join and the second a subselect.

ix) Display the sum of all company salaries along with the company average salary, the minimum salary and the maximum salary.

x) Show the average salary for each work department (display department number and average salary).

xi) List the average salary for each job where the average salary is greater than $55000.00.

xii) Produce a list showing the department number, the average salary, and the number of employees for each department. Do not include employees having a job of 'PRES'. Exclude departments for which less than four employee salaries are being averaged. Sequence the list by descending values of average salary.

xiii) Produce a list of all employees in departments B01, C01, and E21. Include a literal 'MANAGER' for managers and 'NON-MANAGER' for employees who are not managers. Display employee number, last name, and first name. Sequence the list by lastname and firstname. (Managers are identified by the value 'MANAGER' in the job column of the EMP table.)

xiv) Show the average salary for men and the average salary for women for each department. Identify the department by both department number and name. Order the results by ascending department numbers, and descending average salaries within each department.

xv) Display information of departments that have names containing the string 'SYSTEM'.

Question 03

Consider the following relational schema. An employee can work in more than one department; the *pct-time* field of the Works relation shows the percentage of time that a given employee works in a given department.

Emp(*eid*: integer, *ename*: string, *age*: integer, *salary*: real)
Works(*eid*: integer, *did*: string, *pct-time*: integer)
Dept(*did*: string, *budget*: real, *managerid*: integer)

Create the above three tables Emp,Works,Dept . Add records to the tables and answer the fallowing queries :
Execute the queries in SQL.

1. Display the names and ages of each employee who works in both the Hardware department and the Software department.

2. Display the name of each employee whose salary exceeds the budget of all of the departments that he or she works in.

3. Display the *managerids* of managers who manage only departments with budgets greater than $1,000,000.

4. Display the *enames* of managers who manage the departments with the largest budget.

5. If a manager manages more than one department, he or she *controls* the sum of all the budgets for those departments. Display the *managerid* of managers who control more than $5,000,000.

6. Display the *managerid* of managers who control the largest amount.