USN

SET-1

CELEBRATING 30 YEARS
CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A++ GRADE BY NAAC

Internal Assessment Test 2 – December 2022

Sub:	Database Management System					Sub Code:	18CS53	Branch:	ISE
Date:	2/12/2022	Duration:	90 min's	Max Marks:	50	Sem/Sec:	V A, B & C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1a)	Write the relational algebra expression for the following: Consider the following two tables: Employee(Fname, Minit, Lname, ssn, bdate, address, salary, super_ssn, dno) Department(Dname, Dnumber, Mgr_ssn, Mgr_Start_date) Retrieve the Social Security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5 (use Union). Select ssn from employee where super_ssn=5 Union select ssn from department where Dnumber=5;	5	1	L3
1b)	Write the relational algebra expression for the following (use tables mentioned in 1 a): Retrieve the name of the manager of each department. Π Ename (σ super_ssn(Employee.dno=Department.Dnumber) (Employee X Department)	5	1	L2
2 a)	Discuss about nested subquery with a suitable example. A query within another is known as nested query or sometimes referred as subquery. Here, the outer query will execute dependents upon the inner query's output. Eg; Select Ename from Employee where Eno in (Select e_no from department where d_no=5);	5	2	L1
2 b)	Illustrate SQL query with ***'group by'*** and ***'having'*** clauses. Explain the restrictions on using these clauses. 'Group by clause' used to group the data items based on the one or more attributes and having clause can be used to reduce the number output generated by the 'group by' clause. Eg., Select p_name from product Group by p_name having count(*) > 100;	5	2	L2

3	Consider the following tables: Loan(lno, bname, amount) Borrower(lno, cname) Account(ano, bname, amount) Depositor(ano, cname) Write the nested subquery SQL statements using the following: ‘*in’* and ‘*not in’* *‘exist’ and ‘not exist’* *‘some’, ‘all’, ‘not all’, ‘any’* Select lno from loan where lno in (select lno from borrower); Select lno from loan where lno not in(select lno from borrower); Select lno from loan where exist (select lno from borrower where cname= ‘XYZ’); Select lno from loan where not exist (select lno from borrower where cname= ‘XYZ’); Select amount from loan where amount > any (select amount from loan where bname= ‘XYZ’); Select amount from loan where amount > some (select amount from loan where bname= ‘XYZ’); Select amount from loan where amount > all (select amount from loan where bname= ‘XYZ’); Select amount from loan where amount > not all (select amount from loan where bname= ‘XYZ’);	10	2	L3
4	Explain the different domain constraint that can be applicable during table creation in SQL with a suitable example. Domain constraint is a constraint which is used to keep the domain attribute in the consistent state. Create table abcd (emp int not null unique, check (emp > 100), name varchar(10)); **Data type:** is a constrain which describes what type of data should store. **Varchar**(10): here the 10 describes that the number of maximum character can be stored under the domain attribute. **Not null:** the not null constraint used to describe that the domain value should be empty or null (it is a mandatory field insists us to definitely store the value)/ **Unique:** This constraint used to store the unique value under the domain. There will be no duplicate data under that domain attribute.	10	2	L2

	Check: Another domain constraint knows as explicit domain constraint used to check the range of value before going to store the values under the domain attribute. **Primary key:** Key constraints or integrity used to retrieve data or tuple uniquely from the table. **Foreign key:** This key is known as referential integrity constraint to establish the relationship between the two tables.			

5	Consider the following schema for OrderDatabase
	SALESMAN (Salesman_id, Name, City, Commission)
	CUSTOMER (Customer_id, Cust_Name, City, Grade,Salesman_id)
	ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)
	Write SQL queries for the following:
	1. Count the customers with grades above Bangalore's average.
	SELECT GRADE, COUNT (DISTINCT CUSTOMER_ID)
	FROMCUSTOMER1
	GROUP BY GRADE
	HAVING GRADE > (SELECT AVG(GRADE)
	FROM CUSTOMER1
	WHERE CITY='BANGALORE');
	2. Find the name and numbers of all salesmen who had more than one customer.
	SELECT SALESMAN_ID, NAME
	FROM SALESMAN A
	WHERE 1 < (SELECT COUNT (*)
	FROM CUSTOMER1
	WHERE SALESMAN_ID=A.SALESMAN_ID);
	3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)
	SELECT SALESMAN.SALESMAN_ID, NAME, CUST_NAME, COMMISSION
	FROM SALESMAN, CUSTOMER1
	WHERE SALESMAN.CITY = CUSTOMER1.CITY
	UNION
	SELECT SALESMAN_ID, NAME, 'NO MATCH', COMMISSION
	FROM SALESMAN
	WHERE NOT CITY = ANY
	(SELECT CITY
	FROM CUSTOMER1)
	ORDER BY 2 DESC;
		10	2	L3

	4. Create a view that finds the salesman who has the customer with the highest order of a day.
	
	CREATE VIEW ELITSALESMAN AS
	SELECT B.ORD_DATE, A.SALESMAN_ID, A.NAME
	FROM SALESMAN A, ORDERS B WHERE A.SALESMAN_ID = B.SALESMAN_ID
	AND B.PURCHASE_AMT=(SELECT MAX (PURCHASE_AMT)
	FROM ORDERS C
	WHERE C.ORD_DATE = B.ORD_DATE);
	
	5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.
	
	Use ON DELETE CASCADE at the end of foreign key definitions while creating child table
	orders and then execute the following:
	Use ON DELETE SET NULL at the end of foreign key definitions while creating child table
	customers and then executes the following:
	DELETE FROM SALESMAN
	WHERE SALESMAN_ID=1000;			

6	Consider the ‘***Company’*** schema(database) with two tables: Employee(**eid**, name, dob, salary) Salary(**invoice_id,** eid, basic, da, hra, total) Illustrate the following using SQL statements: 1) Change the table name Alter table Employee rename Employee into Empl; 2) Change the attribute name Alter table Employee change column abc xyz varchar(10); 3) Add new attribute Alter table Employee add cname varchar(10); 4) Delete an existing attribute on the table Alter table Employee drop column cname; 5) Modify the data type of an attribute Alter table Employee change abc to xyz int; 6) Add primary key after table creation. Alter table Employee add primary key(abc); 7) Add foreign key after table creation.	10	1	L3

	Alter table employee add foreign key(abc) references xyz(abc) on delete cascade on update cascade; 8) Write the SQL query to illustrate ***‘default’*** and ***‘auto_increment’***. Write about ‘on delete cascade’ and ‘on update cascade’. Create table abc(rno int AUTO_INCREAMENT, name varchar(10) default ‘Today’);			

Faculty Signature　　　　CCI Signature　　　　HOD Signature