WINTER – 2022 EXAMINATION

Subject Name: Database Management System **Model Answer** **Subject Code:** 22319

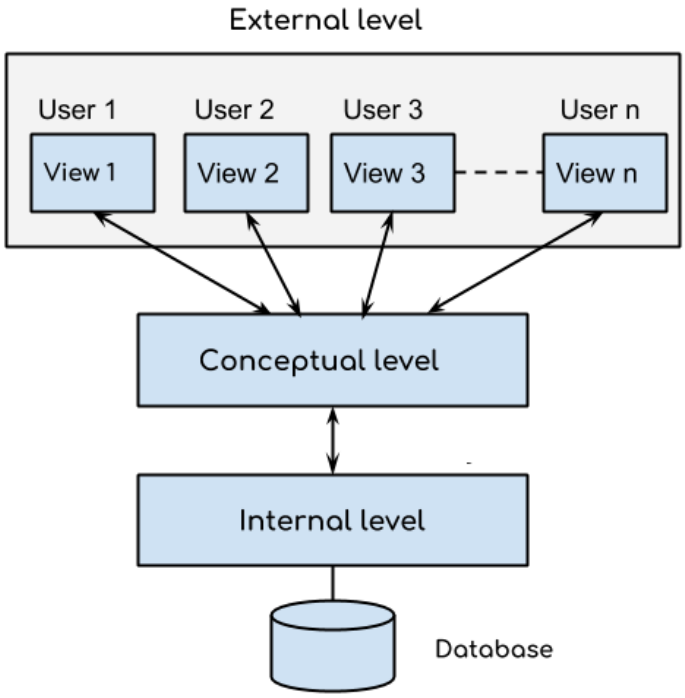
Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**Define** **i) Data Abstraction** **ii) Instance.**	**2 M**
	Ans	**i) Data Abstraction**: Hiding complexity of data structures from end user through different levels is known as data abstraction. Many end users are not computer trained so it is needed to hide complex data structures from them. It has 3 levels : a. Physical level b. logical level c. view level **ii) Instance:** The data stored in database at a particular moment of time is called instance of database. Example: Let's say a table teacher in our database whose name is School, suppose the table has 50 records so the instance of the database has 50 records for now and tomorrow we are going to add another fifty records so tomorrow the instance have total 100 records. This is called an instance.	Each correct definition: 1M

	b)	**State any two advantages of DBMS.**	**2 M**
	Ans	• Reduction in Data redundancy • Data consistency and integrity • Data security • Privacy • Easy access of data • Easy recovery • Flexibility	Any two correct advantages each 1M
	c)	**Define Normalization. Enlist its types.**	**2 M**
	Ans	Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. Types of normalization are: • First normal form(1NF) • Second normal form(2NF) • Third normal form(3NF) • Boyce & Codd normal form (BCNF) • Fourth normal form(4NF)	1 M for correct definition, 1 M for correct types
	d)	**Write syntax for creating and Renaming a table.**	**2 M**
	Ans	Syntax of the CREATE TABLE statement is as follows: CREATE TABLE table_name (column1 datatype (size), column2 datatype(size), column3 datatype(size), ...); Syntax of RENAME TABLE statement is as follows: RENAME old_table _name To new_table_name ;	Correct Syntax of Create Table: 1M, Correct Syntax of Rename Table: 1M
	e)	**Enlist arithmetic and logical SQL operators.**	**2 M**
	Ans	**SQL Arithmetic Operators:** Addition Operator (+)	SQL arithmetic operators: **1M,** SQL logical

		Subtraction Operator (-) Multiplication Operator (+) Division Operator (-) Modulus Operator (+) **SQL Logical Operators:** ALL operator AND operator OR operator BETWEEN operator IN operator NOT operator ANY operator LIKE operator	operators: **1M**
	f)	**Write syntax for creating and dropping views.**	**2 M**
	Ans	**CREATE VIEW Syntax:** create view view_name As select column1, column2… from table_name where condition ; **DROP VIEW Syntax:** DROP VIEW view_name;	Correct syntax for create view: 1M, Correct syntax for drop view: 1M
	g)	**State two advantages of PL/SQL**	**2 M**
	Ans	Advantages of PL/SQL are: • Work can be divided into smaller modules so that it can be manageable and also enhances the readability of the code. • It promotes reusability. • It is secure, as the code is in the database and hides the internal database details from the user. • It improves performance against running SQL queries multiple times.	Any two correct advantages each 1M

2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Explain three level architecture of Database system.**	**4 M**
	Ans	 Fig: Three Level Architecture of DBMS This architecture has three levels: 1. External level 2. Conceptual level 3. Internal level **1. External level** It is also called view level because several users can view their desired data from this level which is internally fetched from database with the help of conceptual and internal level mapping. The user doesn't need to know the database schema details such as data structure; table definition etc. user is only concerned about data which is what returned back to the view level after it has been fetched from database which is present at the internal level. External level is the top level of the three level DBMS architecture. **2. Conceptual level** It is also called logical level. The whole design of the database such as relationship among data, schema of data etc. are described in this level. Database constraints and security are also implemented in this level of architecture.	Explanation=3M Diagram=1M

		This level is maintained by DBA (database administrator). **3. Internal level** This level is also known as physical level. This level describes how the data is stored in the storage devices. This level is also responsible for allocating space to the data. This is the lowest level of the architecture.	
	b)	**Write SQL queries for following:** **i) Create table student with following attributes using suitable data types. Roll no., as primary key, name, marks as not null and city.** **ii) Add column Date of Birth in above student table.** **iii) Increase the size of attribute name by 10 in above student table.** **iv) Change name of Student table to stud.**	**4 M**
	Ans	i) CREATE TABLE Student (Rollno int PRIMARY KEY, name varchar(30) NOT NULL, marks int NOT NULL, city varchar(20)); ii) ALTER TABLE student ADD DateofBirth varchar(20); iii) ALTER TABLE student Modify name varchar(40); iv) RENAME Student to Stud;	Correct Query 1M Each
	c)	**Write and Explain the syntax for creating and dropping indexes with an example.**	**4 M**
	Ans	**CREATE INDEX** The CREATE INDEX command is used to create indexes in tables. It allows duplicate values. Indexes are used to retrieve data from the database very fast. The users cannot see the indexes; they are just used to speed up searches/queries. **Syntax:** CREATE INDEX index_name	Correct explanation of CREATE INDEX Syntax with example: 2M,

		ON table_name (column1, column2, ...); **Example:** The following SQL creates an index named id_firstname on the FirstName column in the Student table: CREATE INDEX id_firstname ON Student (FirstName); **DROP INDEX** The DROP INDEX statement is used to delete an index in a table. **Syntax:** DROP INDEX index_name ON table_name; **Example:** DROP INDEX id_firstname ON Student;	Correct explanation of DROP INDEX Syntax with example: 2M
	d)	**Write a PL/SQL code to print reverse of a number.**	**4 M**

	Ans	**PL/SQL code to print reverse of a number:**
		
		declare
		n number;
		i number;
		rev number:=0;
		r number;
		
		begin
		n:=&n;
		while n>0
		loop
		r:=mod(n,10);
		rev:=(rev*10)+r;
		n:=trunc(n/10);
		end loop;
		
		dbms_output.put_line('reverse is '||rev);
		end;
			Correct Logic: 2M, Correct Code: 2M,
			
			Any other correct logic shall be considered

3.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write down any four Dr. E.F Codd's rules.**	**4 M**
	Ans	**Dr. E. F. Codd's Rules:** Rule 1: The information Rule: all data viewed to users.	Any 4: 1 M each

		Rule 2: Guaranteed Access Rule: all user get access to database Rule 3: Systematic treatment of null values: null value should be empty Rule 4: Dynamic online Catalog: record all transactions in database Rule 5: Data Sub language rule: use only one language Rule 6: View updating rule: table and view updated simultaneously Rule 7: High level insert, delete and update: multiple insert delete update Rule 8: Physical data independence: hardware change Rule 9: Logical data independence: structure change Rule 10: Integrity independence: store correct data Rule 11: Distribution independence: distributed database Rule 12: No subversion rule: no version of language used.	
	b)	**State the use of group by and order by clauses.**	**4 M**
	Ans	**Group by Clause:** Group by clause is used to collect the data as multiple records and group them to produce the result. **Syntax:** SELECT column_name, function(column_name) FROM table_name WHERE condition GROUP BY column_name; **Ex:** select avg(sal) from emp_details **group by** deptno; **Order by Clause:** To view the data in sorted order, the order by clause is used. By default, the data is sorted in ascending order. **Syntax:** SELECT expressions FROM tables [WHERE conditions] ORDER BY expression [ASC | DESC]; **Ex:** select deptno from emp_details **order by** deptno; **Ex:** select deptno from emp_details order by deptno desc; (for descending order)	2 M=group by 2 M=order by
	c)	**Explain the steps of cursor implementation with syntax and example.**	**4 M**
	Ans	**Cursor Implementation:** Steps to create Cursor: **a. Declaring cursor:** Cursor is declared in the declaration section. Syntax: cursor <cursor_name>is<select query>; Ex: cursor **a** is **select ename from emp_details where empno=3;** **b. Opening cursor:** After declaring the cursor, the cursor needs to open. Syntax: open <cursor_name>;	2 M=cursor implementation 1M=syntax 1M=example

		Ex: open **a**;
		c. Fetching a record from cursor:
		Once the cursor is declared and opened, **we need to get records or rows from the cursor**. These records are accessed using the FETCH statement.
		Syntax: fetch <cursor_name> into <variable_list>;
		Ex: fetch **a** into **name**;
		d. Closing cursor:
		Once the cursor is opened and processing is over, we need to close it.
		Syntax: close <cursor_name>;
		Ex: close **a**;
		
		Example:
		
		declare
		name emp_details.ename%type;
		cursor a is select ename from emp_details where empno=3;//***cursor declaration***
		begin
		open a;//***opening the cursor***
		loop
		fetch a into name;//***fetching the rows from cursor***
		update emp_details set comm=3000 where empno=3;
		exit when a%notfound;
		dbms_output.put_line('record updated');
		end loop;
		close a;//***closing the cursor***
		
		end;
			2 M

	d)	**Explain ACID properties of transaction.**	**4 M**
	Ans	**ACID Properties of Transaction:** 1. **A**tomicity 2. **C**onsistency 3. **I**solation 4. **D**urability 1. Atomicity: Atomicity means all the operations included in the single transaction gets executed at a time or none. 2. Consistency: Consistency means update or edits the same data stored at different locations. 3. Isolation: Isolation means all the transactions gets executed independent of each other. 4. Durability: Durability means data can be saved in database permanently until user change it.	1M for each properties

4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Describe any four responsibilities of Database Administrator.**	**4 M**
	Ans	**Responsibilities of Database Administrator (DBA):** 1. Schema Definition: Database or schema can be designed or defined by DBA. 2. Creating storage structure: DBA allocate or decide the space to store the database. 3. Create grant access methods: Different access methods to access the database can be granted by DBA to the users. 4. Schema modification: The database or schema which is already defined can be modified by DBA as per the requirements. 5. Granting authorization: To access the different databases, DBA can grant the authorization to authorized users only. 6. Performance tuning: The problems/errors arise in database accessing; can be resolved by DBA to increase the performance. 7. Regular maintenance: DBA can monitor the transactions in database and maintain the database error free by doing the regular maintenance.	Any 4: 1 M each
	b)	**Explain Primary and Unique key constraint with syntax.**	**4 M**
	Ans	**<u>Primary key constraint:</u>** Primary key constraint applied to any column can't accept the duplicate and null values. This constraint can be applied at the time of table creation. <u>Syntax for Primary Key:</u> CREATE TABLE <Table_Name> (Column1 datatype, Column2 datatype,CONSTRAINT <Name> PRIMARY KEY (Column name)); <u>Example:</u> CREATE TABLE CUSTOMERS (ID INT NOT NULL, AGE INT NOT NULL, ADDRESS CHAR (25), PRIMARY KEY (ID)); **OR**	1 M=primary key use 1M=syntax for primary key 1 M=unique key use 1M=syntax for unique key

```
CREATE TABLE CUSTOMERS
(
 ID   INT            NOT NULL,
 NAME VARCHAR (20)     NOT NULL,
 AGE  INT            NOT NULL,
 ADDRESS  CHAR (25) ,
 SALARY   DECIMAL (18, 2),
 PRIMARY KEY (ID, NAME)
);
```

Unique key constraint:

Unique key constraint applied to any column can't accept the duplicate values (only accepts unique values).

This constraint can be applied at the time of table creation.

Syntax for Unique key:

```
CREATE TABLE <Table_Name>
(
Column1 datatype,
Column2 datatype,CONSTRAINT <Name> UNIQUE KEY (Column name)
);
```

Example:

```
CREATE TABLE students
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
City varchar (255),
UNIQUE (S_Id)
)
```

OR

```
CREATE TABLE students
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
City varchar (255),
CONSTRAINT uc_studentId UNIQUE (S_Id, LastName)
)
```

	c)	**Write and Explain the syntax for creating database trigger.**	**4 M**
	Ans	Database trigger: Triggers can be referred as stored procedures that are fired or executed when an	Explanation of syntax:2M Example of

		INSERT, UPDATE or DELETE statement is given against the associated table. Syntax: create trigger [trigger_name] [before | after] {insert | update | delete} on [table_name] [for each row] [trigger_body] Explanation of syntax: 1. create trigger [trigger_name]: Creates or replaces an existing trigger with the trigger_name. 2. [before | after]: This specifies when the trigger will be executed. 3. {insert | update | delete}: This specifies the DML operation. 4. on [table_name]: This specifies the name of the table associated with the trigger. 5. [for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected. 6. [trigger_body]: This provides the operation to be performed as trigger is fired Example: Given Student Report Database, in which student marks assessment is recorded. In such schema, create a trigger so that the total and percentage of specified marks is automatically inserted whenever a record is insert. Here, as trigger will invoke before record is inserted so, BEFORE Tag can be used. create trigger stud_marks before INSERT on Student for each row set Student.total = Student.subj1 + Student.subj2 + Student.subj3, Student.per = Student.total * 60 / 100;	Trigger=2M
	d)	**Write a trigger which invokes on deletion of record on emp table.**	**4 M**
	Ans	**Delete Trigger:** create or replace trigger trg1 before delete on emp_details declare begin	4 M Or any other correct logic for

		raise_application_error(-20000,'cannot delete the record'); end;	delete trigger
	e)	**Explain Database Recovery techniques in detail.**	**4 M**
	Ans	**Database Recovery Techniques:** Database recovery techniques are used to restore the original data in system from backup. Backward and forward recovery is two types of database recovery. **Recovery Techniques:** 1. Log based recovery. 2. Shadow paging recovery 3. Checkpoints 1. Log based recovery: It records sequence of log records, which includes all activities done by database users. It records the activities when user changes the database. In case of database failure, by referring the log records users can easily recover the data. 2. Shadow paging recovery: This technique is the alternative for log based recovery. In this technique, database is divided into pages that can be stored on the disk. The page table is used to maintain the record of location of pages. In case of database failure, page table is used to recover the parts of database. 3. Checkpoints: Checkpoint records all committed transactions into logs. When system fails, it check log to determine recovery action.	Any 2 techniques: 2 M each
5.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Draw the overall architecture of DBMS. Explain storage manager and query processor components.**	**6 M**
	Ans:	**Storage manager components:** 1. Buffer Manager The Buffer Manager allocates the space to the buffer to store data in it. 2. File Manager The task of the file manager is to manage the space allocation in disk for storing the information and also the data structures used for representing that information.	2M=Correct Diagram with naming 2M= storage manager components

3. Authorization and Integrity Manager

Allows only authorized users to access data and should be hidden from the public users. The Authorization and Integrity Manager verifies the authority of the user trying to access the data and it also checks the integrity constraints when the database is modified.

4. Transaction Manager

A transaction in DBMS is nothing but a very small unit of the program. The Transaction Manager manages all the transaction (program) execution.

2M=query processor

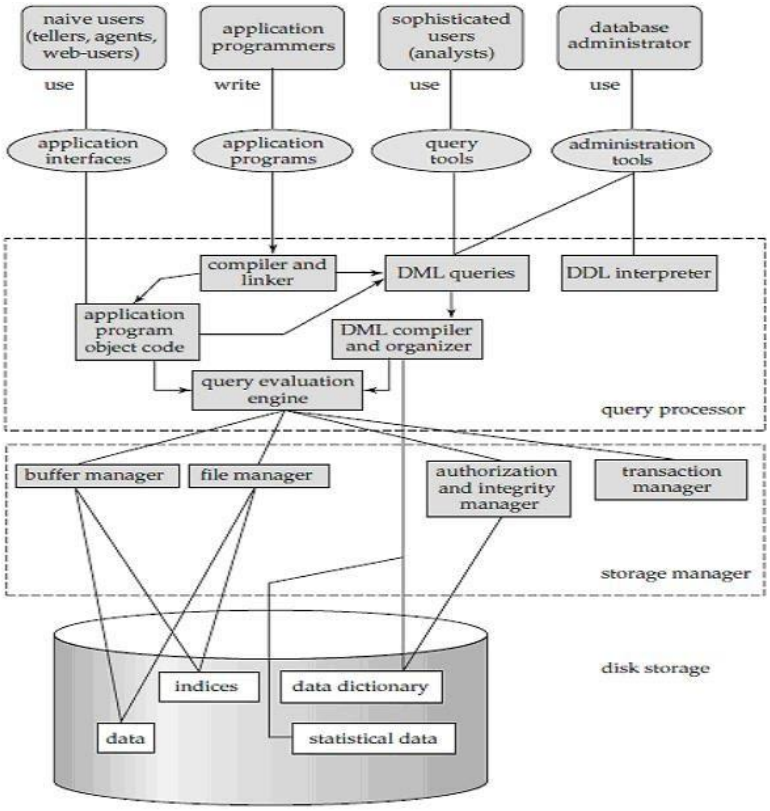


Fig: Overall architecture of DBBMS

Query Processor Components:
Query processor handles the query processing. It processes the query given by the user.

1. DDL Interpreter:

DDL expands to Data Definition Language. DDL Interpreter interprets the DDL statements such as schema definition statements like create, delete, etc.

2. DML Compiler:

DML expands to Data Manipulation Language. DML Compiler compiles (or translates) the DML statements such as select, update and delete statements into low-

		level instructions which is nothing but the machine-readable object code to make it executable. The DML compiler is also responsible for query optimization. 3. Query Evaluation Engine: The Query Evaluation Engine is also referred as "Query execution engine". Query Evaluation Engine first interprets the SQL commands to access the data from the database and then it returns the answer to that query.	
	b)	**Write the SQL queries for following EMP table. Emp (empno, deptno, ename, salary, designation, city.)** **i) Display average salary of all employees.** **ii) Display names of employees who stay in Mumbai or Pune.** **iii) Set the salary of employee 'Ramesh' to 50000.** **iv)Display names of employees whose salaries are less than 50000.** **v) Remove the Record of employees whose deptno is 10.** **vi) Remove the column deptno from EMP table.**	**6 M**
	Ans	i. select avg(salary) from emp; ii. select ename from emp where city='Mumbai' or city='Pune'; iii. update emp set salary=50000 where ename='Ramesh'; iv. select ename from emp where salary<50000; v. delete from emp where deptno=10; vi. alter table emp drop column deptno;	1 M each
	c)	**Write and Explain the syntax for creating, Altering and dropping the sequence.**	**6 M**
	Ans	**<u>Syntax for creating sequence:</u>** CREATE SEQUENCE sequence_name START WITH initial_value INCREMENT BY increment_value MINVALUE minimum value MAXVALUE maximum value CYCLE|NOCYCLE ; .where as:	2 M

		sequence_name: Name of the sequence. **initial_value:** starting value from where the sequence starts. Initial_value should be greater than or equal to minimum value and less than equal to maximum value. **increment_value:** Value by which sequence will increment itself. Increment_value can be positive or negative. **minimum_value:** Minimum value of the sequence. **maximum_value:** Maximum value of the sequence. **cycle:** When sequence reaches its set_limit it starts from beginning. **nocycle:** An exception will be thrown if sequence exceeds its max_value. Example: CREATE SEQUENCE sequence_1 start with 1 increment by 1 minvalue 0 maxvalue 100 cycle; **Alter sequence:** Syntax: alter sequence <sequence_name> maxvalue <number>; Alter sequence can change the maxvalue in the sequence created. **Dropping sequence:** Syntax: drop sequence <sequence_name>; To drop the sequence the DROP command is used.	2 M 2 M
6.		**Attempt any TWO of the following:**	**12 M**
	a)	**Write SQL queries for following. Consider table stud (roll no, name, subl, sub2, sub3)** **i) Display names of student who got minimum mark in subl.** **ii) Display names of students who got above 40 marks in sub2.** **iii) Display count of Students failed in sub2.** **iv) Display average marks of subl of all students.** **v) Display names of students whose name start with 'A' by arranging them in ascending order of subl marks.** **vi) Display student name whose name ends with h' and subject 2 marks are**	**6 M**

		between 60 to 75.	
	Ans	i. select name from stud where sub1= (select min(sub1) from stud); ii. select name from stud where sub2>40; iii. select count(*) from stud where sub2<40; iv. select avg(sub1) from stud; v. select name from stud where name like 'A%' order by sub1; vi. select name from stud where name like '%h' and sub2 between 60 and 75;	1 M each
	b)	**Write a PL/SQL code to check whether specified employee is present in Emp table or not. Accept empno from user. If employee does not exist display message using exception handling.**	**6 M**

	Ans	**PL/SQL Program:**
		declare
		no emp.empno%type;
		begin
		no:=&no;
		select empno into no from emp where empno=no;
		dbms_output.put_line('Empno is present: '||no);
		exception
		when no_data_found then
		dbms_output.put_line('Empno not present');
		end;
			2M =declaration
			2M=select statement
			2M=proper exception message

	c)	**Write SQL queries for following.** **1) Create user named 'user1' having Password '1234** **ii) Assign 'insert' and update' Privilege to 'userl".** **ii) Remove update Privilege assigned to the userl.** **iv) Assign the resource Pemission to userl.**	**6 M**
	Ans	i. create user user1 identified by 1234; ii. grant insert, update on system.emp to user1; iii. revoke update on system.emp from user1; iv. grant create session, unlimited tablespace, create table to user1;	**1 ½ M** each