V2V EDTECH LLP

Online Coaching at an Affordable Price.

OUR SERVICES:

- Diploma in All Branches, All Subjects
- Degree in All Branches, All Subjects
- BSCIT / CS
- Professional Courses

+91 93260 50669

v2vedtech.com

V2V EdTech LLP

v2vedtech

Winter – 19 EXAMINATION

Subject Name: Database Management **Model Answer** **Subject Code: 22416**

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.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1.		**Attempt any Five of the following:**	12
	a	**Define terms: i)Attribute ii) Domain**	2M
	Ans	• Domain: All permissible values of attributes are called as a domain. • Attribute: is a property or characteristics of an entity set.	Domain 1M; Attribute 1 M
	b	**State the use of 'Like' Operator.**	2M
	Ans	The LIKE operator is used in a WHERE clause to search for a specified pattern in a column. There are two operator often used in conjunction with the LIKE operator: • %:The percent sign represents zero, one, or multiple characters • _: The underscore represents a single character.	For Like Explanation 1M For two operator of like1 M
	c	**Write syntax to create view.**	2M
	Ans	Create view <view name> as select <query> OR CREATE VIEW name ASSELECT column1, column2.....FROM table_nameWHERE [condition];	Correct syntax 2 M
	d	**List the types of Cursor.**	2M

	Ans	Cursor: A cursor is a temporary work area created in the system memory when a SQL statement is executed. **Types of Cursor:** 1.Implicit Cursor 2.Explicit Cursor	Listed both the cursor 2 M
	e	**Enlist different types of database users.**	**2M**
	Ans	Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database. 1. Application Programmers 2. Sophisticated Users 3. Specialized Users 4. Native Users/ Naïve Users	½ M of each user of database
	f	**State the properties of Transaction.**	**2M**
	Ans	The ACID properties of transaction. 1. Atomicity. 2. Consistency. 3. Isolation. 4. Durability	½ M of each property of database
	g	**State the use of sequence.**	**2M**
	Ans	A sequence refers to a database object that is capable of generating unique and sequential integer values. Syntax: Create sequence<seq_name> [increment by num][start with num] [maxvaluenum] [minvaluenum][cycle/no cycle][cache/no cache] OR • Sequence is a set of integers 1, 2, 3 … that are generated and supported by some database systems to produce unique values on demand. • A sequence is a user defined schema bound object that generates a sequence of numeric values.	1 M for definition and 1 M for syntax OR 2 M for the correct use of sequence OR Any 2 use 2 M OR

		• Sequences are frequently used in many databases because many applications require each row in a table to contain a unique value and sequences provides an easy way to generate them. • The sequence of numeric values is generated in an ascending or descending order at defined intervals and can be configured to restart when max_value exceeds. OR **Sequence:** • It is database object that generate/produce integer values in sequential order. • It automatically generates primary key and unique key values. • It may be ascending or descending order • It can be used for multiple tables. • Sequence numbers are stored and generated independently of tables	½ M for each points
2.		**Attempt any Three of the following:**	**12M**
	a	**Describe the use of primary key and unique key constraints with example?**	**4M**
	Ans	There are two Entity constraints: 1.Primary Key constraint 2. Unique Constraint **1. Primary Key constraint**: It is use to avoid redundant/duplicate value entry within the row of specified column in table. It restricts null values too. **Syntax:** CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME PRIMARY KEY); **Example:** SQL> CREATE TABLE EMP (ID NUMBER (5)CONSTRAINT ID_PK PRIMARY KEY, NAME VARCHAR2 (10), SAL NUMBER (10)); **2. Unique Constraint:** The UNIQUE constraint uniquely identifies each record in a database table. The UNIQUE and PRIMARY KEY constraints both provide a guarantee for uniqueness for a column or set of columns.	Primary key constraint 2 M, Unique key constraint 2 M

syntax: CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME UNIQUE);

Example: CREATE TABLE PERSONS (P_ID NUM CONSTRAINT P_UK UNIQUE , FIRSTNAME VARCHAR(20), CITY VARCHAR(20));

b **Write any two types of join with example of each?** **4M**

Ans

There are mainly four types of joins that you need to understand. They are:

- INNER JOIN
- FULL JOIN
- LEFT JOIN
- RIGHT JOIN

Listing1 M
And explaining any two with example 3 M

Employee Table:

EmpID	EmpFname	EmpLname	Age	Email ID	Phone No	Address

Project Table

Project ID	EmpID	Client ID	Project Name	ProjectStartDate

Client Table:

Client ID	Client Fame	Client Lame	Age	Client Email ID	Phone No	Address	EmpID

INNER JOIN: This type of join returns those records which have matching values in both tables. So, if you perform an INNER join operation between the Employee table and the Projects table, all the tuples which have matching values in both the tables will be given as output.

SELECT Employee.EmpID, Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName

FROM Employee

INNER JOIN Projects ON Employee.EmpID=Projects.EmpID;

		FULL JOIN: Full Join or the Full Outer Join returns all those records which either have a match in the left (Table1) or the right (Table2) table. For e.g., SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID FROM Employee FULL JOIN Projects ON Employee.EmpID = Projects.EmpID; **LEFT JOIN:** The LEFT JOIN or the LEFT OUTER JOIN returns all the records from the left table and also those records which satisfy a condition from the right table. Also, for the records having no matching values in the right table, the output or the result-set will contain the NULL values. For e.g. SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName FROM Employee LEFT JOIN ON Employee.EmpID = Projects.EmpID ; **RIGHT JOIN:** The RIGHT JOIN or the RIGHT OUTER JOIN returns all the records from the right table and also those records which satisfy a condition from the left table. Also, for the records having no matching values in the left table, the output or the result-set will contain the NULL values. For e.g., SELECT Employee.EmpFname, Employee.EmpLname, Projects.ProjectID, Projects.ProjectName FROM Employee RIGHT JOIN**:** ON Employee.EmpID = Projects.EmpID;	
	c	**Create Sequence seq-1 with starting value 1 and maximum value 20 with an increment of 1. Consider schema Customer (custno, custname, telephone) and use seq-1 for inserting a row in customer table.**	**4M**

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Ans</td><td>Sequence:
• It is database object that generate/produce integer values in sequential order.
• It automatically generates primary key and unique key values.
• It may be ascending or descending order
• It can be used for multiple tables.
• Sequence numbers are stored and generated independently of tables
<table><tr><td>Syntax:
Create sequence <seq_name>
Start with [initial value]
Increment by [value]
Minvalue [minimum value]
Maxvalue [maximum value]
[cycle/no cycle]
[{cache value / No cache}]
[{order / No order}];</td><td>
Create Sequence seq-1
Start with 1
Increment by 1
Minvalue 1
Maxvalue 20;</td></tr></table>For inserting the value using sequence
INSERT INTO Customer VALUES (SEQ-1.NEXTVAL, '1', 'George',1234567890);</td><td>Creating sequence2 M
Inserting in a sequence 2 M</td></tr>
<tr><td></td><td>d</td><td>Consider following schema product (prodid, prod name, rate, qty, manufacturer, qty-in-stock). Write SQL queries for the following
i) Find total number of product manufactured by company “ABC”
ii) Display list of products with highest rated product coming at the top.</td><td>4M</td></tr>
<tr><td></td><td>Ans</td><td>Find total number of product manufactured by company “ABC”</td><td>Each correct queries 2 M</td></tr>
</table>

		Select count(prodid) from product Where manufacturer=”ABC”; **Display list of products with highest rated product coming at the top** SELECT TOP 10 prodid, Product Name, rate FROM Product ORDER BY rate DESC;	
3.		**Attempt any Three of the following:**	**12M**
	a	**Write a PL/SQL code to find sum of numbers from 1 to 20.**	**4M**

	Ans	declare
		i number(10);
		ans number(10);
		begin
		ans:=0;
		i:=1;
		while i<=20 loop
		ans:=ans+i;
		i:=i+1;
		end loop;
		dbms_output.put_line('Sum of 1 to 20 numbers is:'||ans);
		end;
			Correct syntax 2M,
			Correct logic 2M
			Any other logic can be considered

	b	**Describe system and object privileges and also describe use of Grant and Revoke commands with suitable example.**	**4M**
	Ans	**System Privileges:** System privileges are privileges given to users to allow them to perform certain functions that deal with managing the database and the server. e.g: Create user, Create table, Drop table etc. **Object Privileges:**	System privileges1M Object Privileges 1M Grant command use 1/2M,

		Object privileges are privileges given to users as rights and restrictions to change contents of database object – where database objects are things like tables, stored procedures, indexes, etc. Ex.Select, inserts, delete, update, execute, references etc. **Grant:** This command is used to give permission to user to do operations on the other user's object. **Syntax: Grant** <object privileges> **on** <object name> **to** <username>[with grant option] ; **Example:** Grant select, update on emp to user1; **Revoke:** This command is used to withdraw the privilege that has been granted to a user. **Syntax: Revoke** <object privileges> **on** <object name> **from** <username> ; **Example:** Revoke select, update on emp from user1;	example 1/2 M Revoke command use 1/2M, example 1/2 M
	c	**Describe concept of subqueries with example.**	**4M**
	Ans	Subquery is a select statement that is embedded in a clause of another SELECT statement i.e. nesting of queries or query within query. Types of subqueries **1) Single row subqueries** **2) Multiple row subqueries** **3) Multiple column subqueries** **Single row subqueries:** A single row subquery is one that returns one row from inner SELECT statement. This type of subquery uses single row operators = , > , >= , < , <= , < > **Syntax:** SELECT column_name1… column_name n FROM <table_name> WHERE column1 operator (SELECT column from <table_name> where condition); **Example :** Display the employee details whose job title is the same as that of employee 1005. Select empno,ename,job,salary,deptno	Subquery 1M Each type syntax or example 1M

From emp

Where job=(select job from emp where empno=1005);

Multiple row subqueries: Subqueries that return more than one row are called multiple-row subqueries. Multiple row operators are used instead of a subquery, with a multiple row subquery.

Operator	Meaning
IN	Equal to any member in the list.
ANY	Compare value to each value returned by the subquery.
ALL	Compare value to every value returned by the subquery.

Syntax:

SELECT column_name1… column_name n

FROM <table_name>

WHERE column1 operator (SELECT column from <table_name> where condition);

Example
Find the employees who earn the same salary as minimum salary for departments.

Select empno,ename,job,salary,deptno

From emp

Where salary IN (select min(salary) from emp group by deptno);

Multiple column subqueries
Queries that return the values from more than one column are called multiple column subqueries.

Syntax:

SELECT column_name1, ….. column_name n

FROM <table_name>

WHERE (column_name, column_name) IN

		(SELECT column_name, column_name….from <table_name> Where <condition>); **Example:** Display the name, department number, salary and commission of any employee whose salary and commission matches both the commission and salary of any employee in department 10 Query: Select empno,deptno,salary,comm From emp Where (salary,comm) IN (select salary,comm from emp where deptno=10);	
	d	**Write syntax and example of create and drop synonym.**	**4M**
	Ans	**Syntax:** CREATE [OR REPLACE] [PUBLIC] SYNONYM [schema.] synonym name FOR [schema.] object_name; **OR** create synonym name for object_name **Example:** Create synonym new_employee for employee **Syntax to drop synonym:** Drop synonym <synonym name>; **Example:** Drop synonym employee;	Create synonym syntax 1M, example 1M Drop synonym syntax 1M, Example 1M
4.		**Attempt any Three of the following:**	**12M**
	a	**Write steps to create execute and delete stored procedure.**	**4M**
	Ans	**Step 1:** **Stored Procedure creation:** A stored procedure has header, a declaration section, an executable section and optional exception-handling section. Syntax:-	Stored procedure creation 2M

		CREATE OR REPLACE PROCEDURE <procedure_name>(<Argument> {IN | OUT | IN OUT}<Data type>…){IS | AS}
		
		Variable declarations;
		
		Constant declarations;
		
		BEGIN
		<PROCEDURE_BODY>
		
		EXCEPTION
		
		Exception pl/sql block;
		
		END ;
		
		Step 2:
		
		Executing Stored Procedure:
		
		Use EXCE command with help of any application program
		
		Ex:
		
		EXEC use_test
		
		Invoke this procedure from PL/SQL code block
		
		DECLARE
		
		BEGIN
		
		use_test
		
		END;
		
		Step 3 : delete stored procedure
		Syntax : drop procedure <procedure_name>;
		Ex: drop procedure use_test;
			Execute stored procedure 1M
			
			Delete procedure 1M

	b	**Describe simple and composite index.**	**4M**

	Ans	1) Simple index (Single column): An index created on single column of a table is called a Simple Index. **Syntax:** Create index index_name on <tablename><column name>; E.g.: Create index on employee (empno); **Composite (concatenated)**: Indexes that contain two or more columns from the same table which are useful for enforcing uniquely identify a row. **Syntax**: Create index index_name on <tablename><Column_name1, Column_name2>; E.g.: Create index on employee (ename, empno);	For each type description 1M, Syntax or example 1M
	c	**Consider the following schemas** **Student(rollno, name, dt_of_birth,telephone)** **Marks (rollno, sub1_marks, sub2_marks, per) Write SQL queries for the following.** **i) Display student's rollno, name, and marks of both subjects for all students.** **ii)Delete all those students records who secured less than 35%** **iii)Display all the students whose name start with 'A'** **iv)Update telephone number of student with rollno 101 as 9800010111**	**4M**

	Ans	i) Select Student.rollno, Student.name, marks.sub1_marks, marks.sub2_marks from Student, marks where Student.rollno=marks. Rollno;
		
		ii)Delete from Student where rollno=(Select rollno from marks where per<35);
		
		iii) Select name from Student where name like 'A%';
		
		(OR)
		
		Select * from Student where name like 'A%';
		
		iv) Update Student set telephone=9800010111 where rollno=101;
			Each correct Query 1M

	d	**Describe types and causes of failure in database environment.**	**4M**
	Ans	**Types and causes of Failure in database environment:** **1. Hardware Failure/System crash** There is a hardware malfunction that causes the loss of the content of volatile storage, and brings transaction processing to a halt. The content of non-volatile storage remains intact, and is not corrupted or changed. **2. Software Failure** The database software or the operating system may be corrupted or failed to work correctly, that may causes the loss of the content of volatile storage, and results into database failure**.** **3. Media Failure** A disk block loses its content as a result of either a head crash or failure during a data-transfer operation. **4. Network Failure** A problem with network interface card or network connection can cause network failure. **5. Transaction Failure** **i) Logical error:** the transaction can no longer continue with its normal execution because of some internal condition, such as wrong input values, data not found, data overflow or resource limit exceeded. **ii) System error:** A system entered in state like deadlock **6.Application software Error:** -The problem with software accessing the data from database. -This may cause database failure as data cannot be updated using such application to it. -Logical error in program cause one or more transaction failure. **7. Physical disaster** The problem caused due to flood, fire, earthquake etc.	Any 4 types and cause 1M each
	e	**Write a PL/SQL code to raise zero_divide exception, in case of division of a number by another.**	**4M**

	Ans	**DECLARE**
		A number:=20;
		B number:=0;
		C number;
		BEGIN
		dbms_output.put_line('First Num : '||A);
		dbms_output.put_line('Second Num : '||B);
		C:= A / B; --Raise Exception
		dbms_output.put_line(' Result ' || C);
		Result will not be displayed
			Correct syntax 2M,
			
			Correct logic2M
			
			Any other logic can be considered.

		EXCEPTION
		WHEN ZERO_DIVIDE THEN
		dbms_output.put_line(' Trying to Divide by zero :: Error ');
		END;	

5.		**Attempt any Two of the following:**	**12M**
	a	**Write SQL statements for following** i) Create table student with rollno, name, d-o-b, percentage, assign rollno as primary key. ii) Add new column email in student table. iii) Delete table 'student' with its structure and data.	**6M**
	Ans	i) Create table student with rollno, name, d-o-b, percentage, assign rollno as primary key. **Ans :** Create table student(rollno number(5) primary key, name varchar2(20), d-o-b date, percentage number(6,2)) ; ii) Add new column email in student table. **Ans :** Alter table student add email varchar2(30); iii) Delete table 'student' with its structure and data. **Ans :** Drop table student;	**Each query 2M**
	b	**a) Consider following schema:** **employee{empid,empname,designation,salary,deptno}** **dept { deptno,deptname,location}** **Write SQL queries for following :** **i)Find maximum salary for deptno=10;** **ii Increase salary of all employee by 5%** **iii)Get the names of all 'Manager'** **iv) Display deptnames located at 'Pune' and 'Nagpur'.**	**6M**
	Ans	**i)**Find maximum salary for deptno=10; **Ans:**Select max(salary) from employee where deptno=10; **ii)** Increase salary of all employee by 5% **Ans:** Update employee set salary=salary+(salary*0.05);	**Query i) 1M** **Query ii) 2M** **Query iii) 1M**

		iii) Get the names of all 'Manager'. **Ans:** Select empname from employee where designation="Manager"; **iv)** Display deptnames located at 'Pune' and 'Nagpur'. **Ans:** Select deptname from dept where location='Pune' or location='Nagpur';	**Query iv) 2M**
	c	**Write a PL/SQL code to create a function name square_no to calculate square of number and also have another PL/SQL code to call this function.**	**6M**

	Ans	**PL/SQL code for function to calculate square of a number:**
		create or replace function square_no(n in number)
		return number is sqrno number;
		begin
		sqrno := n*n;
		return(sqrno);
		end;
		PL/SQL code to call above function :
		declare
		n1 number;
		sno number;
		begin
		n1 := &n1;
		sno := square_no(n1);
		dbms_output.put_line("Number="||n1);
		dbms_output.put_line("Square =;||sno);
		end;
			PL/SQL code for defining function : 3M
			
			PL/SQL code for calling above function : 3M

6.		**Attempt any Two of the following:**	**12M**
	a	**Consider schema 'employee' created by 'user1'** **Write SQL queries for following :** i) Grant 'select' and 'insert' permissions to user2. ii) Assign all privileges for the user user3. iii) Remove 'select' permission from user2 for table 'employee'. iv) Grant 'update' permission to user2 and user3 v) Remove all permission from user3.	**6M**

		vi) Assign 'resource' permission to user2.	
	Ans	i) Grant 'select' and 'insert' permissions to user2. **Ans :** grant select, insert on employee to user2; ii) Assign all privileges for the user user3. **Ans :** grant all on employee to user3; iii) Remove 'select' permission from user2 for table 'employee'. **Ans :** revoke select on employee from user2; iv) Grant 'update' permission to user2 and user3 **Ans :** grant update on employee to user2, user3; v) Remove all permission from user3. **Ans :** revoke all on employee from user3; vi) Assign 'resource' permission to user2. **Ans :** grant resource to user2;	**Each Query 1M**
	b	**Create a trigger which invokes on updation of record on emp table.**	**6M**

	Ans	create trigger trigger_update
		on emp
		after update
		as
		begin
		Select * from employee;
		end;
		end;
			Note: any example which can execute trigger before or after updation can be considered.
			Correct Logic 3M, Correct syntax 3M

	c	**Consider following schema:** **Person {personid,name,address,city,telephone}** **Write PL/SQL queries for following:**	**6M**

		i) Create sequence seq-pid with start value 100 and maximum value 120 and increment by 1. Use seq-pid to insert personid into table person. **ii)** Create view view-person containing details of persons from city "Mumbai" and "Pune" **iii)** Create synonym syn-person on table person owned by user 'Scott' delete synonym syn-person.	
	Ans	i) Create sequence seq-pid with start value 100 and maximum value 120 and increment by 1. Use seq-pid to insert personid into table person. **Ans:** Create sequence seq_pid start with 100 increments by 1 maxvalue 120. Insert into person (personid) values (seq_pid.nextval); ii)Create view view-person containing details of persons from city "Mumbai" and "Pune" **Ans:** Create view person as select * from person where city='Mumbai' or city='Pune'; **Or** Create view view_peson as select * from persons where city in ('Mumbai','Pune'); iii) Create synonym syn-person on table person owned by user 'Scott' delete synonym syn-person. **Ans:** create synonym syn_person for scott.person; Drop synonym syn_person;	**Each query 2M**