WINTER – 19 EXAMINATION
Subject Name: Relational Database Management System

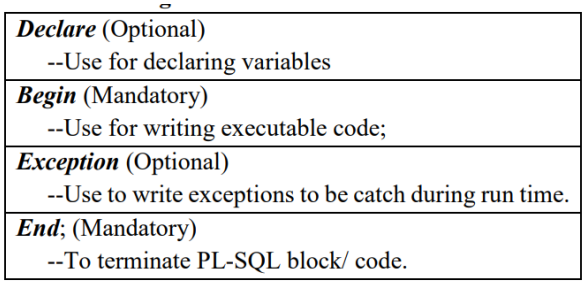
Model Answer

Subject Code: 17332

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 Six of the following:**	12M
	a	**Define Relational Database Management System.**	2M
	Ans	RDBMS is Relational Database Management System which is an environment where data is represented in the form of relations, with enforced relationships between the tables	Correct definition 2M
	b	**List four applications of database management system.**	2M
	Ans	1. Banking 2. Airlines 3. Universities 4. Credit Card transactions 5. Telecommunication 6. Finance 7. Sales. 8. Manufacturing 9. On-line Retailers	1/2M each
	c	**Define updating anomalies.**	2M

	Ans	An anomaly is an inconvenient or error-prone situation arising while processing the tables.Relations that have redundant data may have problems called update anomalies, which are classified as , 1. Insertion anomalies 2. Deletion anomalies 3. Modification anomalies	Correct definition 2M
	d	**List four different parts of SQL.**	**2M**
	Ans	SQL has four parts components: 1. Data Manipulation Language (DML), 2. Data Definition Language (DDL), 3. Data Control Language (DCL), 4. Data Query Language (DQL).	1/2M each
	e	**List four limitations of PL/SQL.**	**2M**
	Ans	1. Hard to maintain temporal tables 2. Partial Control. Due to the hidden business rules, programmers using SQL doesn't have full control over the database. 3. Cost. There are some SQL versions which have high operating cost, so it creates difficulty for some programmers to access those versions. 4. Interface. SQL has a complex interface that creates difficulty for some user to access it.	1/2M each
	f	**Draw the diagram of PL/SQL execution environment.**	**2M**
	Ans		Correct diagram 2M
	g	**Which attribute is used to find out how many rows were fetched from cursor so far? Give example.**	**2M**
	Ans	**%ROWCOUNT** is used to find out how many rows were fetched from cursor so far CREATE TABLE employees_temp AS SELECT * FROM employees;	1M attribute, 1M example

		BEGIN UPDATE employees_temp SET salary = salary * 1.05 WHERE salary < 5000; DBMS_OUTPUT.PUT_LINE('Updated ' || SQL%ROWCOUNT || ' salaries.');END;	
	h)	**What are the limitations of views?(four points)**	**2M**
	Ans	1. **Can't create an index of views:** In SQL, we cannot create an index on views. It is because indexes are not utilized when we query data against the views. 2. **SQL views cannot be updateable in some situations:** Actually, the simple view can be updateable but a view created on a complex SELECT statement with JOIN or SUBQUERY etc. cannot be updateable. 3. **SQL does not support materialized views:** We cannot create materialized views because MySQL does not support it. 4. **Using subquery in the FROM clause of view depends on SQL version:** Actually, we can use a subquery in the FROM clause of view. 5. **Cannot create a TEMPORARY view:** Actually, the definition cannot refer to a TEMPORARY table hence we cannot create a TEMPORARY view. 6. **Cannot associate a trigger with a view:** we cannot associate a trigger with a view	1/2M each
1	**b)**	**Attempt any Two of the following:**	**8M**
	a	**Explain integrity constraints with example.**	**4M**
	Ans	**Integrity constraints:** Not Null constraint, CHECK constraint, Primary Key constraint, Unique Constraint, Referential Integrity Constraint 1. **Not Null**: By default, all columns in tables allows null values. When a NOT NULL constraint is enforced on column or set of columns it will not allow null values. **Syntax:** CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE NOT NULL); **Example:** SQL>CREATE TABLE STUDENT (ROLL_NO NUMBER (5), NAME VARCHAR2(20) NOT NULL); 2. **CHECK:** The constraint defines a condition that each row must satisfy. A single column can have multiple check condition. **Syntax:** CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME CHECK);	1M each constraint with example

		Example: SQL> CREATE TABLE EMP (ID NUMBER (5), NAME VARCHAR2(10), SAL NUMBER(10) CONSTRINT CHK_SAL CHECK (SAL>15000)); 3. **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)); 4. **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. It allows null value. **Syntax:** CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME UNIQUE); **Example:** CREATE TABLE PERSONS (P_ID NUMBER CONSTRAINT P_UK UNIQUE, FIRSTNAME VARCHAR2(20), CITY VARCHAR2(20)); 5. **Referential Integrity Constraint**: It is a relational database concept in which multiple tables share a relationship based on the data stored in the tables, and that relationship must remain consistent. A value of foreign key is derived from primary key which is defined in parent table. **Syntax:** CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME REFERENCES PARENT_TABLE_NAME (PARENT_TABLE_COL_NAME) ON DELETE CASCADE, COLUMN_NAME DATA_TYPE); **Example:** CREATE TABLE DEPARTMENT (EMP_ID NUMBER(5) REFERENCES EMP(EMP_ID), DNO NUMBER(3));	
	b	**List four different types of join and explain it with student example considering Enroll_no, snane, course, city.**	**4M**
	Ans	**Different types of joins:** 1) Natural Join 2) Outer Join 3) Inner Join	Listing 2M, Example 2M

4) Cross Join

For example, Assume Following two tables student and course

Student

Enroll_no	Sname	Course	city
1	JOHN	CSE	NEW YORK
2	STEVE	CSE	NEW YORK
3	MARK	IT	DALLAS
4	JIMMY	IT	MICHIGAN
5	TIM	CIVIL	DALLAS

Course_info

Course	Duration in hours
CSE	48
IT	45
EXTC	46

SELECT Student.Enroll_no, Student. Sname from Student INNER JOIN Course;

Output:

Enroll_no	Sname	Course	Duration in hours
1	JOHN	CSE	48

			2	STEVE	CSE	48
			3	MARK	IT	46
			4	JIMMY	IT	46
	c	**Explain the loop control structure in PL/SQL with example and two demerits.**				**4M**

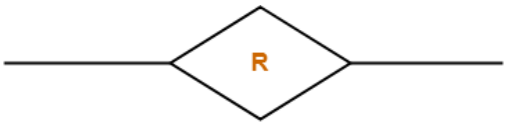
	Ans	**1.simple loop**
		
		General syntax:
		
		initialization;
		loop
		body of the loop
		incr/decr;
		exit when condition;
		end loop;
		
		The looping variable is initialized first. The body of the loop gets executed and the increment or decrement of the looping variable is done. This step executes repeatedly till the exit condition turns true. The condition is checked at the end.
		
		Example:
		
		DECLARE
		a NUMBER:=1;
		BEGIN
		dbms_output.put_line ('Program started.');
		LOOP
		dbms_output.put_line(a);
		a:=a+1;
		EXIT WHEN a>5;
		END LOOP;
		dbms_output.put_line('Program completed');
		END;
		
		1.while loop
		
		General syntax:
		initialization
		while condition
		loop
		body
						Any loop control structure with example 2M, each demerit 1M

		incr/decr end loop The initialization of the loop variable is done first. The condition is checked as the next step. If the condition is true, the statements in the body of the loop get executed. Further, increment and decrement of the looping variable is done. The steps are executed till the condition in the while loop turns false. **2.for loop** **General syntax:** for loop_variable in initial_value ... final_value loop body end loop The looping variable is initialized to the initial_value. It is then compared to the final value. If true, then the body of the loop gets executed. The value of the looping variable is incremented by 1. The looping variable is compared to the final value and the steps repeat till the condition turns false. To print in reverse, the general syntax: forloop_variable in reverse initial_value .. final_value loop body end loop **Demerits:** 1. Possibility of entering an infinite loop if not properly coded 2. Boundary conditions may result in wrong values if not handled properly.	
2.		**Attempt any Four of the following:**	**16M**
	a	**Enlist the six characteristics of database administrator and explain any two of them.**	**4M**
	Ans	**Characteristics of database administrator** **1.Schema Definition** The Database Administrator creates the database schema by executing DDL statements. Schema includes the logical structure of database	Listing 2M, Any one explanation 2M

		table (Relation) like data types of attributes, length of attributes, integrity constraints etc. **2. Storage structure and access method definition** The DBA creates appropriate storage structures and access methods by writing a set of definitions which is translated by data storage and DDL compiler. **3. Schema and physical organization modification** DBA writes set of definitions to modify the database schema or description of physical storage organization. 4**. Granting authorization for data access** The DBA provides different access rights to the users according to their level. Ordinary users might have highly restricted access to data, while you go up in the hierarchy to the administrator, you will get more access rights. Integrity constraints specifications: Integrity constraints are written by DBA and they are stored in a special file which is accessed by database manager while updating data. **5. Routine Maintenance** Some of the routine maintenance activities of a DBA is given below. (i) Taking backup of database periodically (ii) Ensuring enough disk space is available all the time. (iii) Monitoring jobs running on the database. (iv) Ensure that performance is not degraded by some expensive task submitted by some users. **6. Integrity- constraint specification** Integrity constraints are written by DBA and they are stored in a special file, which is accessed by database manager, while updating the data	
	b	**List the types of cardinality relation and explain it with diagram.**	**4M**
	Ans	**Types of cardinality relation** 1. Many-to-Many cardinality (m:n) 2. Many-to-One cardinality (m:1) 3. One-to-Many cardinality (1:n) 4. One-to-One cardinality (1:1) **1)Many-to-Many Cardinality-** By this cardinality constraint,	Each relation with diagram 1M

- An entity in set A can be associated with any number (zero or more) of entities in set B.
- An entity in set B can be associated with any number (zero or more) of entities in set A.

Symbol Used-

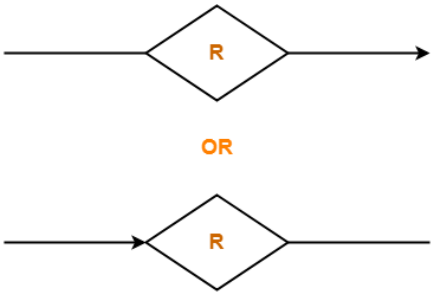


Cardinality Ratio = m : n

2. Many-to-One Cardinality-

By this cardinality constraint,
An entity in set A can be associated with at most one entity in set B.
An entity in set B can be associated with any number (zero or more) of entities in set A.

Symbol Used-

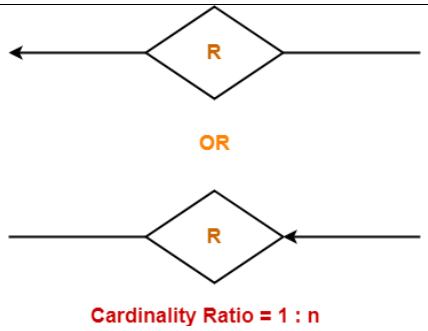


Cardinality Ratio = m : 1

3. One-to-Many Cardinality-

By this cardinality constraint, An entity in set A can be associated with any number (zero or more) of entities in set B.
An entity in set B can be associated with at most one entity in set A.

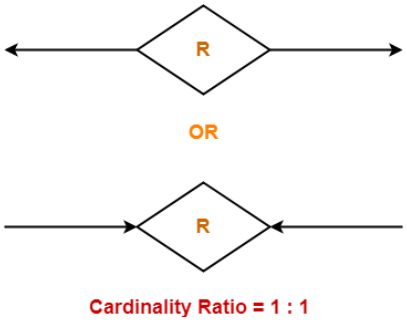
Symbol Used-



4. One-to-One Cardinality-

By this cardinality constraint,
An entity in set A can be associated with at most one entity in set B.
An entity in set B can be associated with at most one entity in set A.

Symbol Used-

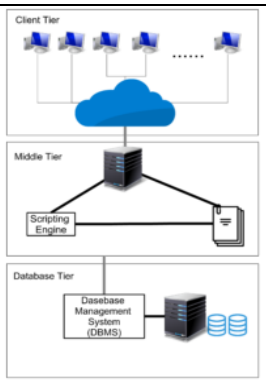


	c	**Explain the command to update the data in the table with example using Data Manipulation Language command.**	**4M**
	Ans	The **UPDATE** statement is used to modify the existing records in a table. **Syntax:** UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition; Example: UPDATE Customers SET ContactName = 'Alfred Schmidt', City= 'Frankfurt' WHERE Customer ID = 1;	Syntax: 2M, example: 2M
	d	**Give difference between Having clause and Where clause.(four points)**	**4M**
	Ans		each point 1M

Having clause	Where clause
The Where clause specifies the criteria which individual records must meet to be selected by a query .it can be used without the GROUP BY clause.	The Having clause cannot be used without the GROUP BY clause.
The WHERE clause selects rows before grouping	The HAVING clause selects rows after grouping
The WHERE clause cannot contain aggregate functions.	The HAVING clause can contain aggregate functions.
WHERE clause is used to impose condition on SELECT statement as well as single row function and is used before GROUP BY clause	HAVING clause is used to impose condition on GROUP Function and is used after GROUP BY clause in the query.
SELECT Column, AVG(Column_name)from Table_mame WHERE Column>value GROUP BY Column)name	SELECT Column, AVG(Column_name)from Table_mame WHERE Column>value GROUP BY Column)name Having Column>value OR < value

	e	**Define synonyms. Explain how to drop synonym with example.**	
	Ans	A **s**ynonym is an alternative name for objects such as tables, views, sequences, stored procedures, and other database objects. **Dropping Synonym:** Once a synonym has been created in Oracle, you might at some point need to drop the synonym. **Syntax** DROP [PUBLIC] SYNONYM [schema .] synonym_name [force]; **Example:** DROP PUBLIC SYNONYM suppliers; This DROP statement would drop the synonym called suppliers that was defined earlier.	definition: 2M, example: 2M)
	f	**Explain the domain relational calculus with example.**	

	Ans	Domain relational calculus uses domain variable that takes values from attributes domain rather than values for entire tuple. Domain relational calculus uses list of attribute to be selected from the relation based on the condition. There are different symbols with specific meaning which can be used to write domain calculus expression**;-** 1. ε- belong to 2. ∃-There exits 3. ∀ – for all 4. ¬ – not 5. => implies 6. ^ -and 7. ∨ - or An expression is of the form $\{< x_1, x_2, \ldots, x_n > \mid P(x_1, x_2, \ldots, x_n)\}$ where the x_i, $1 \leq i \leq n$, represent attributes, and P is a **predicate**. Example : Find branch name, loan number, customer name and amount for loans of over $1200. $\{< b, l, c, a > \mid < b, l, c, a > \in borrow \wedge a > 1200\}$ Where 'b' represents domain of branch name, 'l' represents domain of loan number, 'c' represents domain of customer name and 'a' represents domain of amount from table 'borrow'	Explanation 2M Example: 2M
3.		**Attempt any Four of the following:**	**16M**
	a	**Give difference between two tier client/server and three tier client/server architecture (four points).**	**4M**
	Ans		Any correct 4 points- 1M each

Sr. no	Comparison	Two-tier Architecture	Three -tier Architecture
1	Diagram		 Three Tier Architecture Web

						Database Application		
			2	**Architecture Type**	Client -Server Architecture	Web -based application		
			3	**Working**	Client will hit request directly to server and client will get response directly from server, The direct communication takes place between client and server. There is no intermediate between client and server. Because of tight coupling a 2 tiered application will run faster.	Here in between client and server middle ware will be there, if client hits a request it will go to the middle ware and middle ware will send to server and vice versa.		
			4	**Layers**	2-tier means 1. Design layer/Client Application (Client Tier 2. Data layer/Database (Data Tier)	3-tier means 1. Design layer /presentation 2. Business layer or Logic layer / data access tier 3. Data layer / data tier.		
			5	**Security**	Less secured as client can talk to database directly	Highly secured as client is not allowed to talk to database directly		
			6	**Scalability**	Poor	Excellent as requests can be load balanced between servers		
			7	**Reusability**	Mostly clients are monolithic and thereby reusability not possible	Reusability more with services implementation		

			8	**Advantages**	1. Easy to maintain and modification is bit easy. 2. Communication is faster.	3. Better Re-usability. 4. Improve Data Integrity. 5. Improved Security – Client is not direct access to database. 6. Forced separation of user interface logic and business logic. 7. Business logic sits on small number of centralized machines (may be just one). 8. Easy to maintain, to manage, to scale, loosely coupled etc.	
			9	**Disadvantages**	1. In two tier architecture application performance will be degrade upon increasing the users. 2. Cost-ineffective	Increase Complexity/Effort	
	b	**Describe 2NF with suitable example.**					**4M**

	Ans	**Second Normal Form (2NF):** A relation is said to be in the second normal form if it is in first normal form and all the non key attributes are fully functionally dependent on the primary key. **Example:** If in the relation Supp(SNO,SNAME, LOCATION,PNO,QTY),the attributes SNAME AND LOCATION depends on SNO and QTY dependence on (SNO, PNO) so the table can be split up into two tables as Supplier (SNO,SNAME,LOCATION) and SP(SNO,PNO,QTY) and now both the tables are in second normal form.	Correct explanation-2M any example-2M

SNO	SNAME	LOCATION
S1	abc	Mumbai
S2	Pqr	Pune
S3	Lmn	Delhi

Supplier

SNO	PNO	QTY
S1	P1	200
S2	P2	300
S3	P3	400

SP

	c	**Explain referential integrity constraints with example of student database.**	**4M**
	Ans	**Referential integrity constraint:** • It is used to establish a parent child relationship between two tables. • A value of foreign key is derived from the primary key.	Correct explanation-2M example 2M

		• Primary key is defined in a parent table and foreign key is defined in child table. The child table contains the values for foreign key column which are present in parent table's primary key column but no other than that. **Syntax:** Create table name (column datatype size references Parenttablename (primary key attribute)….) **Example:** Create table student (rollid number (4) not null, deptid number(4) references dept (deptno), SName varchar2(10)); **After table creation the foreign key is added as:** Alter table student add constraint fk_dept foreign key (deptid) references dept (deptno);	
	d	**Explain Range searching operators and patterns matching operator in SQL with example.**	**4M**
	Ans	**SQL LIKE Operator- patterns matching operator** The LIKE operator is used to list all rows in a table whose column values match a specified pattern. It is useful when you want to search rows to match a specific pattern, or when you do not know the entire value. For this purpose we use a wildcard character '%'. **For example:** To select all the students whose name begins with 'S' SELECT first_name, last_name FROM student_details WHERE first_name LIKE 'S%'; The output would be similar to: **first_name last_name** ---------- ---------- Stephen Fleming Shekar Gowda The above select statement searches for all the rows where the first letter of the column first_name is 'S' and rest of the letters in the name can be any character. There is another wildcard character you can use with LIKE	Like operator explanation with example- 2M & Between-and operator explanation with example- 2M

operator. It is the underscore character, ' _ '. In a search string, the underscore signifies a single character.

For example: to display all the names with 'a' second character,

```
SELECT first_name, last_name
FROM student_details
WHERE first_name LIKE '_a%';
```

The output would be similar to:

first_name	last_name
Rahul	Sharma

NOTE: Each underscore act as a placeholder for only one character. So you can use more than one underscore. Eg: ' __i% '-this has two underscores towards the left, 'S__j%' - this has two underscores between character 'S' and 'i'.

SQL BETWEEN ... AND Operator- Range searching operators

The operator BETWEEN and AND, are used to compare data for a range of values.

For Example: to find the names of the students between age 10 to 15 years, the query would be like,

```
SELECT first_name, last_name, age
FROM student_details
WHERE age BETWEEN 10 AND 15;
```

The output would be similar to:

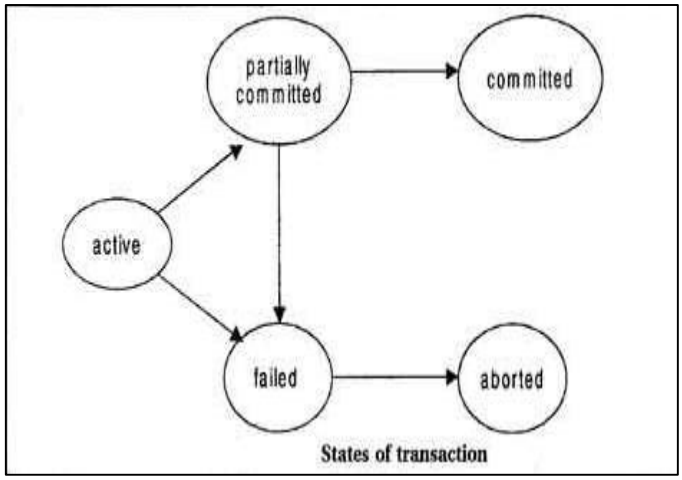
first_name	last_name	age
Rahul	Sharma	10
Anjali	Bhagwat	12
Shekar	Gowda	15

	e	**Define sequence. Give syntax to create sequence command and explain it with example.**	**4M**

	Ans	**Sequence :** 1. Sequence creates a series of values which are computer generated and which can be inserted into a table. 2. Oracle provides an object called as a Sequence that can generate numeric values. The value generated can have maximum of 38 digits. 3. These numbers can be ascending or descending order. 4. Provide intervals between numbers. **Operations allowed on sequence:** **1) Create a sequence :** Create sequence < sequence name > [incremented by <integer value> start with <integer value>Maxvalue<integer value>/nomaxvalue minvalue<integer value>/nonvalue cycle/ no cycle] cache<integer value>/Nocache order/no order **INCREMENT BY:** Specifies the interval between sequence number. It can be any positive or negative value but not zero. If this clause is omitted the default value is 1. **MINIVALUE:** Specifies the sequence minimum value. **NOMINVALUE:** Specifies the maximum value of 1 for an ascending sequence and – (10) ^26 for a descending sequence. **MAXVALUE:** Specifies the maximum value that a sequence can generate. **NOMAXVALUE:** Specifies a maximum of 10^27 for an ascending sequence or -1 for a descending sequence is the sequence minimum value(1) and for a descending sequence, it is the maximum value(-1). **CYCLE:** Specifies that the sequence continues to generate repeat values after reaching either it's maximum. **NOCYCLE:** Specifies that a sequence cannot generate more values after reaching the maximum value. **Example:** Create sequence addr_sqe increment by 1 start with 1 minivalue 1 Maxville 999 cycle;	Definition-1M, syntax-2M, example-1M
	f	**What is the procedure to write the PL/SQL code?**	
	ANs	The Declaration section: Code block start with a declaration section, in which memory variables, constants, cursors and other oracle objects can be declared and if required initialized.	

		The Begin section: Consist of set of SQL and PL/SQL statements, which describe processes that have to be applied to table data. Actual data manipulation, retrieval, looping and branching constructs are specified in this section. The Exception section: This section deals with handling errors that arise during execution data manipulation statements, which make up PL/SQL code block. Errors can arise due to syntax, logic and/or validation rule The End section: This marks the end of a PL/SQL block. ***Declare*** (Optional) --Use for declaring variables ***Begin*** (Mandatory) --Use for writing executable code; ***Exception*** (Optional) --Use to write exceptions to be catch during run time. ***End***; (Mandatory) --To terminate PL-SQL block/ code.	
4.		**Attempt any Three of the following:**	**16M**
	a	**Explain need of data mining over data warehousing in detail.**	**4M**
	Ans	**Need of Data Mining:** • Data Warehousing is the process of extracting and storing data to allow easier reporting. Whereas Data mining is the use of pattern recognition logic to identify trends within a sample data set, a typical use of data mining is to identify fraud, and to flag unusual patterns in behaviour. For Example, Credit Card Company provide you an alert when you are transacting from some other geographical location which you have not used previously. This fraud detection is possible because of data mining. • The main difference between data warehousing and data mining is that data warehousing is the process of compiling and organizing data into one common database, whereas data mining is the process of extracting meaningful data from that database. Data mining can only be done once data warehousing is complete.	Any 4 points of need of mining

		• Data warehouse is the repository to store data. On the other hand, data mining is a broad set of activities used to uncover patterns, and give meaning to this data. • Data warehousing is merely extracting data from different sources, cleaning the data and storing it in the warehouse. Whereas data mining aims to examine or explore the data using queries. For example A data warehouse of a company stores all the relevant information of projects and employees. Using Data mining, one can use this data to generate different reports like profits generated etc. • Data warehouse is architecture whereas; data mining is a process that is an outcome of various activities for discovering the new patterns. • A data warehouse is a technique of organizing data so that there should be corporate credibility and integrity, but, Data mining is helpful in extracting meaningful patterns those are not found, necessarily by only processing data or querying data in the data warehouse. • Data warehouse contains integrated and processed data to perform data mining at the time of planning and decision making, but data discovered by data mining results in finding patterns that are useful for future predictions. • Data warehouse supports basic statistical analysis. The information retrieved from data mining is helpful in tasks like Market segmentation, customer profiling, credit risk analysis, fraud detection etc. • Data warehousing is the process of pooling all relevant data together, whereas Data mining is the process of analysing unknown patterns of data. Data warehouses usually store many months or years of data. This is to support historical analysis. Data mining is the use of pattern recognition logic to identify trend within a sample data set.	
	b	**Describe weak entity and strong entity.**	**4M**
	Ans	**Strong Entity Set:** An entity set that has sufficient attributes to form a primary key is called as strong entity set. **OR** An entity set that have a primary key of its own is referred as Strong entity set. **Weak Entity set:** An Entity set that does not have a Primary key of its own is referred as a weak entity set. **OR** An entity set that does not have sufficient attribute to form a primary key is called as Weak Entity Set.	Description of weak entity - 2M, strong entity-2M

	c	**Explain the DROP command with example.**	**4M**
	Ans	**DROP Command:** The SQL DROP Command is use to delete all records and schema of the table. **Syntax:** **DROP Table <table name>;** **Example:** Drop table emp;	Description of drop-2M, example-2M
	d	**Draw a neat labelled state diagram of transaction, list five steps of transaction and explain it.**	**4M**
	Ans	 States of transaction **A transaction must be in one of the following states:** **1. Active**: the initial state, the transaction stays in this state while it is executing. **2**. **Partially committed**: after the final statement has been executed. **3. Failed**: when the normal execution can no longer proceed. **4. Aborted**: after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction. **5. Committed**: after successful completion. A transaction has committed only if it has entered the committed state. Similarly, a transaction has aborted only if it has entered the aborted state. A transaction is said to have terminated if has either committed or aborted. A transaction starts in the active state. When it finishes its final statement, it enters the partially committed	Diagram -1M List-1M Explain -2M

		state. At this point, the transaction has completed its execution, but it is still possible that it may have to be aborted, since the actual output may still be temporarily hiding in main memory and thus a hardware failure may preclude its successful completion.	
	e	**State the importance of views. Give its syntax, explain its advantages.**	**4M**
	Ans	To the database user, the view appears just like a real table, with a set of named columns and rows of data. SQL creates the illusion of the view by giving the view a name like a table name and storing the definition of the view in the database. Views are used for security purpose in databases, views restricts the user from viewing certain column and rows means by using view we can apply the restriction on accessing the particular rows and columns for specific user. Views display only those data which are mentioned in the query, so it shows only data which is returned by the query that is defined at the time of creation of the View. **Syntax for creating view.** Create view <viewname> as select <query>; **OR** Example : Create viewemp_info as select Emp_no, Emp_name from Employee; **Advantages of views** **Security**: Each user can be given permission to access the database only through a small set of views that contain the specific data the user is authorized to see, thus restricting the user's access to stored data **Query Simplicity** A view can draw data from several different tables and present it as a single table, turning multi-table queries into single-table queries against the view. **Structural simplicity** Views can give a user a "personalized" view of the database structure, presenting the database as a set of virtual tables that make sense for that user. **Consistency** A view can present a consistent, unchanged image of the structure of the database, even if the underlying source tables are split, restructured, or renamed.	Importance of views -1M, syntax1M, advantages-2M

Data Integrity

If data is accessed and entered through a view, the DBMS can automatically check the data to ensure that it meets the specified integrity constraints.

Logical data independence.

View can make the application and database tables to a certain extent independent. If there is no view, the application must be based on a table. With the view, the program can be established in view of above, to view the program with a database table to be separated.

	f	**Give differences between shared lock and exclusive lock.**	**4M**

Shared lock	Exclusive lock.
Shared locks are placed on resources whenever a read operation (select) is performed.	Exclusive locks are placed on resources whenever a write operation (INSERT, UPDATE And DELETE) are performed
Multiple shared locks can be simultaneously set on a resource	Only one exclusive lock can be placed on a resource at a time. i.e. the first user who acquires an exclusive lock will continue to have the sole ownership of the resource, and no other user can acquire an exclusive lock on that resource
On Select sql operation shared lock is used	On INSERT, UPDATE And DELETE sql operation exclusive lock is used
Syntax: lock table table_name in share mode	Syntax: lock table table_name in exclusive mode
It can lock the transaction only for reading. This lock opens a table/database in read mode.	Syntax: lock table table_name in exclusive mode

Any correct 4 points- 1M each

5.		**Attempt any Three of the following:**	**12- M**
	a	**Enlist the types of database users and explain any two of them along with their interfaces.**	**4M**

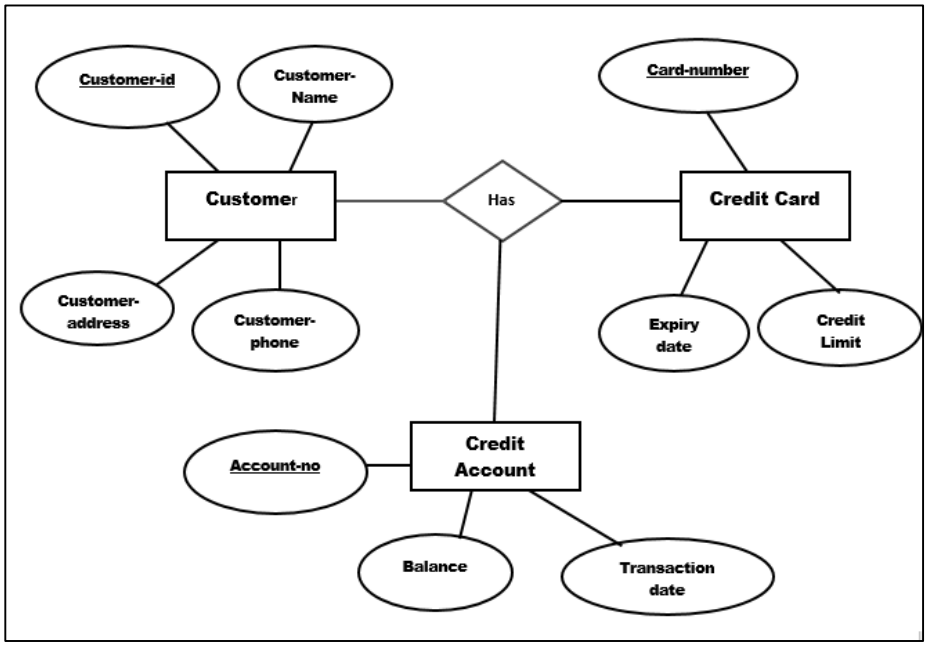
	Ans	**Types of Database users** 1. Naive users 2. Application programmers 3. Sophisticated users 4. Specialized users 5. Database administrator **Naive users (Consider any 2 types)** Naïve users are unsophisticated users. They interact with the system through the application program. They give data as input through application program or get output data which is generated by application programs. **Sophisticated users** Interact with the system by making the requests in the form of query language. These queries are then submitted to the query processor. Query processor converts the DML statements into lower level interactions which are understandable by storage manager. **Application programmers** Application programmers are the users who write the application programs. These programmers use programming tools to develop the program. RAD technology is used to write the program. **Specialized users** These users are not traditional. They write some special application programs which are not regular applications like CAD, knowledge based and expert system. **Database administrator:** Responsible for managing whole database system, create and maintains database. Manages users who can access the database and manages integrity issue. Manages performance of system as and when required.	List -2 M, Explanation of Any 2 types- 2M
	b	**Give diffence between Primary key and Foreign key.**	**4M**
	Ans		Any 4 points- 4M

Primary key	Foreign key
A primary key is an attribute in Relation that uniquely identifies the rows in relation.	A Foreign Key is a field (or collection of fields) in one table that refers to the Primary Key in another table.

			A Primary key does not hold NULL values neither redundant value.	Foreign key can contain redundant values.
			Syntax: CREATE TABLE TABLE_NAME (ATTR1 DATA_TYPES PRIMARY KEY, ATTR2 DATA_TYPES……..);	Syntax: CREATE TABLE TABLE_NAME (ATTR1 DATA_TYPES REFERENCES BASE_TABLE (DATATYPE)...);
			Example: CREATE TABLE STUDENT (ROLL_NO NUMBER(3) PRIMARY KEY, NAME VARCHAR2(15));	Example: CREATE TABLE SPORTS (S_ROLL NUMBER(3) REFERENCES STUDENT(ROLL-NO), COURSE VARCHAR2(10));
	c	**Explain any four data functions in SQL.**		**4M**
	Ans	**i) Lower (char)-** Returns the input string with all letters in lower case. **Example:** SQL>Select lower ('RAJESH') from dual; **ii) Upper (char)-**Returns the input string with all letters in upper case. **Example:** SQL>Select upper ('rajesh') from dual; **iii) Ltrim (char, set)-** It removes or trims from left of character string. **Example:** SQL>Select Ltrim('university','univ') from dual; **iv) Rtrim (char, set)-** It removes or trims from right of character string. **Example:** SQL>Select Rtrim('university','sity') from dual; **v) Length(char)-**It returns length of character string. **Example:** SQL> Select length('University') from dual; **vi) Concat (str1, str2...)-**Returns the string that result from concatenating the arguments. **Example:** Select Concat('employee', 'name') from dual; **vii) Avg** – calculates the average of a set of values.	1 function – 1 M, consider any 4 String, Arithmetic, Date and time, Aggregate Functions

		Example : Select Avg(unitsinstock) from products; **viii)Count** – counts rows in a specified table or view. **Example:** Select Count(unitsinstock) from products; **ix)Min** – gets the minimum value in a set of values. **Example:** Select Min(unitsinstock) from products; **x) Max** – gets the maximum value in a set of values. **Example:** Select Max(unitsinstock) from products;	
	d	**Define snapshot. List three types of snapshot with its syntax and explain it with example.**	**4M**
	Ans	**Snapshots:** It is also known as materialized view. It is a copy of either an entire single table or set of its rows or collection of tables, View or either rows using join, grouping and selection criteria. **Types of Snapshots** 1. Simple Snapshot 2. Complex Snapshots 3. Read-Only Snapshots **Syntax :** CREATE DATABASE database_snapshot_name ON (NAME =logical_file_name, FILENAME ='os_file_name') [,...*n*] AS SNAPSHOT OF source_database_name [;] **Example :** Create snapshot emp_snap refresh with rowed as select * from emp;	Definition1M, list-1M,syntax-1M, example-1M
	e	**Explain implicit cursor & explicit cursor with example.**	**4M**
	Ans	**Implicit Cursors**: When DML statements like Delete, Insert, Update and Select statements are executed, implicit statements are created to process these statements. Oracle provides few attributes called as implicit cursor attributes to check the status of DML operations. The cursor attributes available are %FOUND, %NOTFOUND, %ROWCOUNT, and %ISOPEN. The values of the cursor attributes always refer to the most recently executed SQL statement. **Explicit Cursors:** When precise control is needed over query processing, cursor can be explicitly declared in the declarative part of any PL/SQL block, subprogram, or package.	Implicit cursor-2M, Explicit Cursor-2M

		An explicit cursor is defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row. A suitable name is provided for the cursor.	
	f	**Explain the importance of shared lock with example.**	**4M**
	Ans	**Shared Lock:** • It can lock the transaction only for reading. This lock opens a table/database in read mode. • If a transaction Ti has obtained a shared-mode lock (denoted by S) on item Q, then Ti can read, but cannot write, Q. Shared Lock is provided to the readers of the data. These locks enable all the users to read the concurrent data at the same time, but they are not allowed to change/ write the data or obtain exclusive lock on the object. • It could be set for table or table row • Lock is released or unlocked at the end of transaction. **For Example**: Lock table employee in Share Mode;	Explanation3M , Example – 1M
6.		**Attempt any Three of the following:**	**16M**
	a	**Explain any four functions of database administrator.**	**4M**
	Ans	**Functions of database administrator** **1.Schema Definition** The Database Administrator creates the database schema by executing DDL statements. Schema includes the logical structure of database table (Relation) like data types of attributes, length of attributes, integrity constraints etc. **2. Storage structure and access method definition The DBA** Creates appropriate storage structures and access methods by writing a set of definitions which is translated by data storage and DDL compiler. **3. Schema and physical organization modification** DBA writes set of definitions to modify the database schema or description of physical storage organization. **4. Granting authorization for data access** The DBA provides different access rights to the users according to their level. Ordinary users might have highly restricted access to data,	Any 4functions 1M each

		while you go up in the hierarchy to the administrator, you will get more access rights. **5. Routine Maintenance** Some of the routine maintenance activities of a DBA is given below. (i) Taking backup of database periodically (ii) Ensuring enough disk space is available all the time. (iii) Monitoring jobs running on the database. (iv) Ensure that performance is not degraded by some expensive Task submitted by some users. **6. Integrity- constraint specification:** Integrity constraints are written by DBA and they are stored in a special file, which is accessed by database manager, while updating the data.	
	b	**Draw ER diagram for managing credit card account.**	
	Ans		
	c	**Explain the requirement to maintain database security.**	**4M**
	Ans	Database Security can be maintained by following ways: • Triggers can be written for Imposing security authorizations, Preventing invalid transactions, Enforcing referential integrity, Event logging and storing information on table access, Auditing. • Database Administrator has the authority to grant privileges or permissions to other users, public or a specific role to either execute a specific task within the database or to have access into the database to carry out some particular query.	Explanation - 4M

		• The DBA ensures this periodically backing up the database on magnetic tapes or remote servers. In case of failure, such as virus attack database is recovered from this backup. • Views are created for security reasons. View is a logical copy of physical table. It doesn't exist physically. With the help of view, we can give restricted access to users. When view is used, underlying table is invisible, thus increasing security.	
	d	**Name and explain the command used to undo the changes done in the current transaction.**	**4M**
	Ans	Rollback Command is used to undo the changes done in current transaction. Rollback: The ROLLBACK command is the transactional command used to undo transactions that have not already been saved to the database. This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued. Syntax : ROLLBACK;	Identification of Rollback command 1M, explanation3M
	e	**List the different types of exception handling and explain it with example.**	**4M**

	Ans	**Exception Handling types**
		
		1. Predefined Exception
		2. User defined Exception
		
		Predefined Exception :
		
		Are always automatically raised whenever related error occurs. The most common errors that can occur during the execution of PL/SQL. Not declared explicitly i.e. cursor already open, invalid cursor, no data found, zero divide and too many rows etc. are handled by system defined Exceptions.
		
		User defined exception:
		
		It must be declare by the user in the declaration part of the block where the exception is used. It is raised explicitly in using statements Raise_application_error(Exception_Number, Error_Message);
		
		For example : Program to handle Zero divide exception (Any other example can be considered)
		
		DECLARE
		a int:=10;
		b int:=0;
		answer int;
			list types – 1M, Explanation1M any valid example -2M

```
BEGIN
  answer:=a/b;
  dbms_output.put_line(‘ result after division is'||answer);
exception
  WHEN zero_divide THEN
  dbms_output.put_line('dividing by zero please check the values again');
END;
```

f **Give difference between procedure and triggers.** **4M**

Ans

Procedure	Triggers
Procedures are executed when they are called	Triggers are fired when particular SQL commands (DML) are executed
Procedure do not have events and related actions	Triggers have events and related actions
Procedure are called explicitly	Triggers are called implicitly
Procedures can accept parameters	Triggers cannot accept parameters
Procedures may return values	Trigger returns exception or status of current event
Syntax : CREATE OR REPLACE PROCEDURE Procedure_name[AS] BEGIN --- sql statements END;	Syntax : CREATE [OR REPLACE] TRIGGER trigger_name {BEFORE | AFTER | INSTEAD OF } {INSERT [OR] | UPDATE [OR] | DELETE} [OF col_name] ON table_name [REFERENCING OLD AS o NEW AS n] [FOR EACH ROW] WHEN (condition) BEGIN --- sql statements END;

Any 4 valid points-4M