


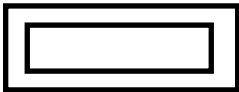
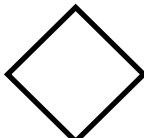
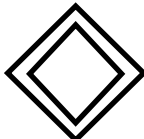

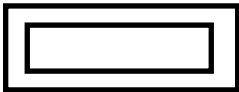
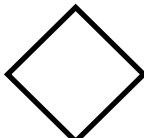
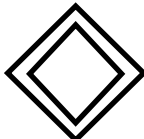

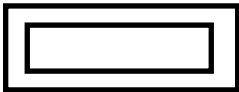
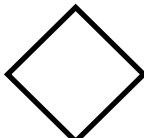
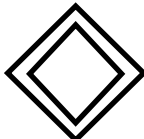


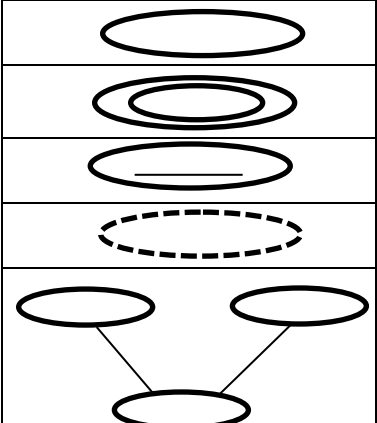
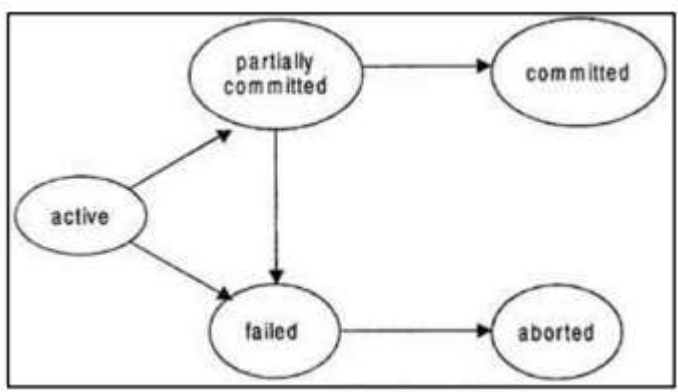
WINTER- 17 EXAMINATION

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.	(a)	Attempt any SIX of the following :		Marks 12										
	(i)	List any four DBMS software.		2M										
	Ans:	Oracle, MySQL, Microsoft Access, Microsoft SQL Server, IBM DB2, FoxPro		(Any 4 Softwares: ½ mark each)										
	(ii)	List and draw any four symbols used in ER model.		2M										
	Ans:		<table><tr><th>Symbol</th><th>Use</th></tr><tr><td></td><td>Entity</td></tr><tr><td></td><td>Weak Entity</td></tr><tr><td></td><td>Relationship</td></tr><tr><td></td><td>Weak Relationship</td></tr></table>	Symbol	Use		Entity		Weak Entity		Relationship		Weak Relationship	(Any 4 symbols: ½ mark each)
Symbol	Use													
	Entity													
	Weak Entity													
	Relationship													
	Weak Relationship													

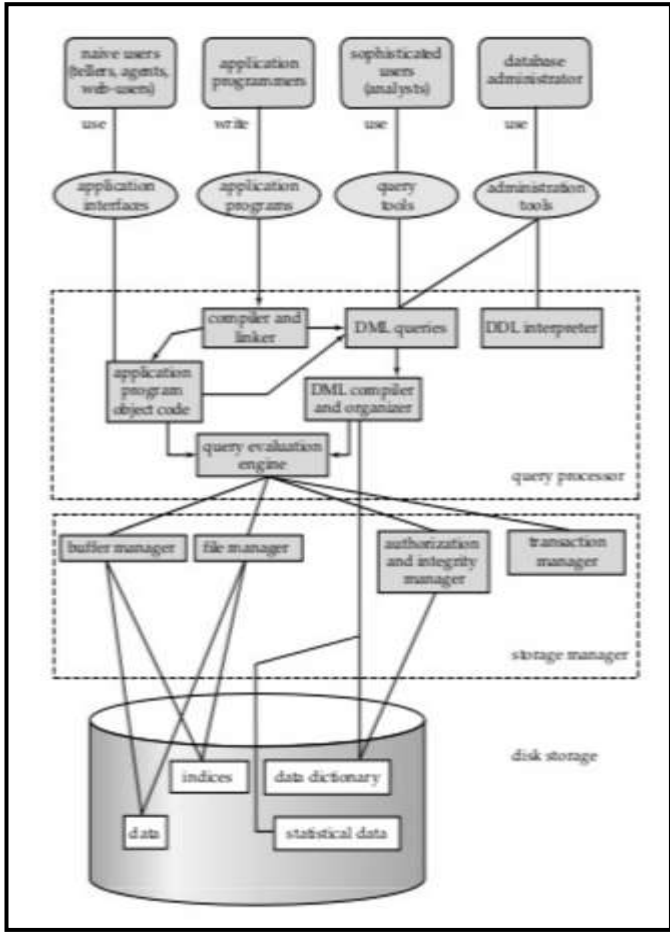
			Attribute	
			Multivalued Attribute	
			Primary key Attribute	
			Derived Attribute	
			Composite Attribute	
	(iii)	Define normalization.		2M
	Ans:	Normalization can be defined as process of decomposition of database tables to avoid the data redundancy.		(Definition: 2 marks)
	(iv)	Define domain and entity.		2M
	Ans:	<ul style="list-style-type: none"> • Domain: All permissible values of attributes is called as a domain. • Entity: An entity is a thing or object in the real world with an independent existence. An entity may be an object with a physical existence. 		(Domain : 1 mark; Entity: 1 mark)
	(v)	List any four DCL commands.		2M
	Ans:	1. GRANT 2. REVOKE 3. SAVE POINT 4. ROLLBACK 5. COMMIT		(Any 4 DCL Commands :½ mark each)
	(vi)	Draw the state diagram of transaction.		2M
	Ans:			(Correct diagram: 2 marks)



	(vii)	Explain the use of truncate statement. Give example.	2M																				
	Ans:	TRUNCATE statement removes all rows from a table, but the table structure and its columns, constraints, indexes, and so on remain. Syntax: - TRUNCATE TABLE TABLE_NAME; Example: TRUNCATE TABLE STUDENT;	(Description 1 mark, Example 1 mark)																				
	(viii)	State statement used in PL/SQL to display the output with syntax.	2M																				
		The statement used in PL/SQL to display output is: Syntax: dbms_output.put_line(var/msg);	(Correct statement: 2 marks)																				
	(b)	Attempt any TWO of the following :	Marks 8																				
	(i)	State four differences between DBMS and RDBMS.	4M																				
	Ans:	<table><thead><tr><th>DBMS</th><th>RDBMS</th></tr></thead><tbody><tr><td>Old version of software to handle the databases.</td><td>Latest version of software for handling databases.</td></tr><tr><td>Can relate one table to another table.</td><td>RDBMS can relate one database to another database.</td></tr><tr><td>Data security is low as compare to RDBMS.</td><td>Level of data security is very high as compare to DBMS.</td></tr><tr><td>Data storage capacity is less as compare to RDBMS.</td><td>Data storage capacity is very high.</td></tr><tr><td>Not easy to maintain data integrity.</td><td>Data integrity is one of the most important features of RDBMS. It can be maintained easily in RDBMS.</td></tr><tr><td>Works better in single user or few user systems.</td><td>Works very efficiently and give good performance over the network.</td></tr><tr><td>All Codd's 12 rules are not followed.</td><td>All Codd's 12 rules are followed.</td></tr><tr><td>DBMS normalization process will not be present.</td><td>RDBMS fully support normalization</td></tr><tr><td>e.g. FoxPro, Ms-Access</td><td>e.g. SQL-server, Oracle,IBM-DB2</td></tr></tbody></table>	DBMS	RDBMS	Old version of software to handle the databases.	Latest version of software for handling databases.	Can relate one table to another table.	RDBMS can relate one database to another database.	Data security is low as compare to RDBMS.	Level of data security is very high as compare to DBMS.	Data storage capacity is less as compare to RDBMS.	Data storage capacity is very high.	Not easy to maintain data integrity.	Data integrity is one of the most important features of RDBMS. It can be maintained easily in RDBMS.	Works better in single user or few user systems.	Works very efficiently and give good performance over the network.	All Codd's 12 rules are not followed.	All Codd's 12 rules are followed.	DBMS normalization process will not be present.	RDBMS fully support normalization	e.g. FoxPro, Ms-Access	e.g. SQL-server, Oracle,IBM-DB2	(Any 4 points of differences: 1 mark each)
DBMS	RDBMS																						
Old version of software to handle the databases.	Latest version of software for handling databases.																						
Can relate one table to another table.	RDBMS can relate one database to another database.																						
Data security is low as compare to RDBMS.	Level of data security is very high as compare to DBMS.																						
Data storage capacity is less as compare to RDBMS.	Data storage capacity is very high.																						
Not easy to maintain data integrity.	Data integrity is one of the most important features of RDBMS. It can be maintained easily in RDBMS.																						
Works better in single user or few user systems.	Works very efficiently and give good performance over the network.																						
All Codd's 12 rules are not followed.	All Codd's 12 rules are followed.																						
DBMS normalization process will not be present.	RDBMS fully support normalization																						
e.g. FoxPro, Ms-Access	e.g. SQL-server, Oracle,IBM-DB2																						
	(ii)	Explain Alter Command. Give syntax of add and modify options.	4M																				
	Ans:	Alter Command is use to make necessary changes in existing structure/schema of table. If a DBA/user wants to make changes in a table which was created earlier then one can use	(Description 2 marks, Syntax of																				



		<p>Alter command.</p> <p>The alter command has following 4 options.</p> <p>Add Modify Drop Rename</p> <p>Add option: The add option is use to add new column to existing table. To add new column/attribute one can use following syntax.</p> <p>SQL>Alter table table-name add attribute_name datatype(size);</p> <p>Example:</p> <p>SQL>Alter table student add hobby varchar2(10);</p> <p><i>Add Constraints:-</i></p> <p>Another use of add option is to add constraints on existing attribute. To add constraint following syntax is use.</p> <p>SQL>Alter table table-name add constraint constraint_name constraint_type(list of attribute(s));</p> <p>SQL>Alter table student add constraint pk primary key(roll_no);</p> <p>Modify option: The modify option is use to modify existing column of table. To modify column/attribute one can use following syntax.</p> <p>SQL>Alter table table-name modify attribute_name new_datatype(size);</p> <p>Example:</p> <p>SQL>Alter table student modify hobby varchar2(20);</p>	<p>add: 1 mark, Syntax of modify: 1 mark)</p>
	(iii)	Explain exception handling with its two types.	4M
	Ans:	<p>Exception Handling: Exception is nothing but an error. Exception can be raise when DBMS encounters errors or it can be raised explicitly. When the system throws a warning or has an error it can lead to an exception. Such exception needs to be handled and can be defined internally or user defined. Exception handling is nothing but a code block in memory that will attempt to resolve current error condition.</p> <p>Syntax: DECLARE ; Declaration section</p>	<p>(Description: 2 marks, Two types :2 marks)</p>

		<p>...executable statement; EXCEPTION WHEN ex_name1 THEN ; Error handling statements/user defined action to be carried out; END;</p> <p>Types of Exception: 1) Predefined Exception/system defined exception/named 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. Programs are handled by system defined Exceptions. 2) 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 sequence of statements using: Raise_application_error(Exception_Number, Error_Message);</p>	
2.		Attempt any FOUR of the following :	Marks 16
	(a)	Draw neat labelled diagram of overall DBMS structure.	4M
	Ans:		(Correct Diagram:4 marks)



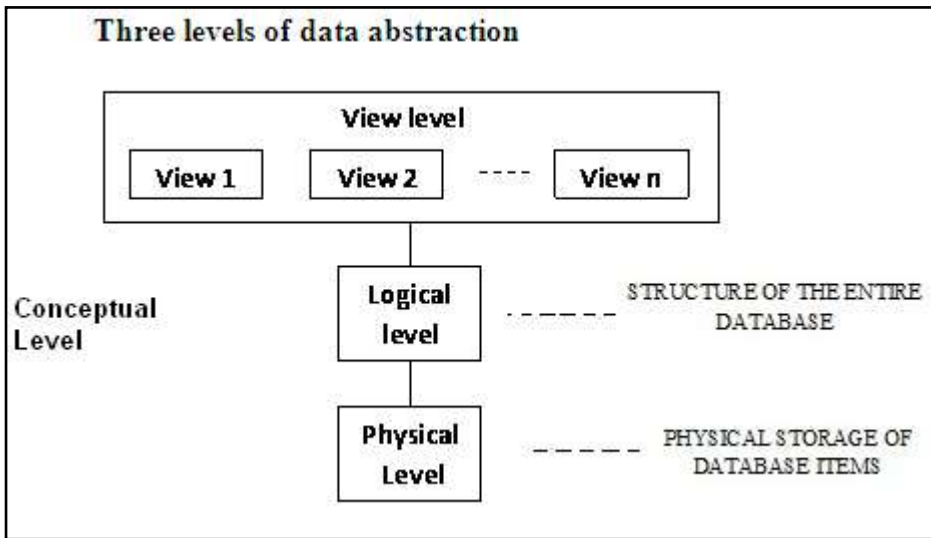
	(b)	Describe following keys : (i) Primary key (ii) Foreign key	4M
	Ans:	<p>(i) Primary Key: A primary key is an attribute in Relation that uniquely identifies the rows in relation. A Primary key does not hold NULL values neither redundant value. Example: CREATE TABLE STUDENT (ROLL_NO NUMBER(3) PRIMARY KEY, NAME VARCHAR2(15));</p> <p>(ii) Foreign Key: Foreign key is an attribute that point to primary key of other relation. Foreign key can contain redundant values.</p> <p style="text-align: center;">OR</p> <p>A Foreign Key is a field (or collection of fields) in one table that refers to the Primary Key in another table.</p> <p>Syntax: CREATE TABLE TABLE_NAME (ATTR1 DATA_TYPES REFERENCES BASE_TABLE (DATATYPE),.....);</p> <p>Example: CREATE TABLE SPORTS (S_ROLL NUMBER(3) REFERENCES STUDENT(ROLL-NO), COURSE VARCHAR2(10));</p>	(Primary key: 2 marks, Foreign Key: 2 marks)
	(c)	Explain functional and transitive dependency.	4M
	Ans:	<p>Functional Dependency: A functional dependency occurs when one attribute in a relation uniquely determine another attribute.</p> <p style="text-align: center;">OR</p> <p>Let 'R' be a relation and let X and Y be any arbitrary attributes of R, then it can be said that Y is functionally dependent on X if and only if, each X value is associated with precisely one Y value. And it can be shown as $X \rightarrow Y$. e.g. emp_id \rightarrow ename meaning ename functionally dependent on emp_id.</p> <p>Transitive Dependency: A Functional dependency is said to be transitive if it is indirectly formed by two functional dependencies. The Transitive dependency exist between $X \rightarrow Z$ if the following three Functional dependencies hold true.</p> <p>1. $X \rightarrow Y$ 2. $Y \text{ does not } \rightarrow X$ 3. $Y \rightarrow Z$ then $X \rightarrow Z$</p>	(Functional Dependency: 2 marks; Transitive dependency : 2 marks)



		<p>The Transitive Dependency can only occur in a relation which has three or more attributes. This dependency needs to be eliminated to make relation in 3NF.</p> <p style="text-align: center;">OR</p> <p>If we have a primary key A and a non-key domain B and C, where C is more dependent on B than A; and B is directly dependent on A, then C can be considered transitively dependent on A.</p> <p>Example: Consider the relation SUPPLIER (SNO, SNAME, CITY, STATUS).</p> <p>In this dependency STATUS is dependent on CITY and CITY is functionally dependent on SNO, which shows a transitive dependency i.e. STATUS is transitively depending on SNO(i.e. via CITY).</p>	
	(d)	Explain group by, order by clause of SQL with example.	4M
	Ans:	<p>Group by: The GROUP BY STATEMENT is used with aggregate functions (MAX, MIN, SUM, COUNT etc.) to group the result-set by one or more columns;</p> <p>Syntax: Select Column-Name(s) from Table-Name where condition <i>GROUP BY</i> Column-Name;</p> <p>Example: Select COUNT (Roll_no), Branch from Student <i>GROUP BY</i> Branch;</p> <p>Order By: To achieve a specific order then one can use the ORDER BY clause. ORDER BY clause allows sorting by one or more columns. Records can be displayed in ascending or descending order.</p> <p>Syntax: SELECT Column-Name(s) from Table-name where condition <i>ORDER BY</i> Column-name(s) asc/desc;</p> <p>Example: SELECT ROLL_NO, PERCENTAGE FROM STUDENT <i>ORDER BY</i> PERCENTAGE;</p>	(Group by Clause: 2 marks; Order by clause: 2 marks)
	(e)	Describe synonyms. How to create and drop synonym?	4M
	Ans:	<p>A synonym is an alias to a table, view, materialized view sequence, procedure, function. A</p> <p>To Create Synonym following syntax is used. Syntax: create synonym synonym_name for object; Example:</p>	(Description :2 marks, creating synonym: 1 mark,



		Create <i>synonym</i> std for student; <i>To Drop Synonym following syntax is used.</i> Syntax: Drop Synonym Synonym_name; Drop <i>synonym</i> std;	Dropping Synonym :1 mark)
	(f)	Explain implicit and explicit cursors.	4M
	Ans:	<p>Implicit Cursor:</p> <p>Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement.</p> <p>Whenever a DML statement (INSERT, UPDATE and DELETE) is issued, an implicit cursor is associated with this statement. For INSERT operations, the cursor holds the data that needs to be inserted. For UPDATE and DELETE operations, the cursor identifies the rows that would be affected.</p> <p>In PL/SQL, you can refer to the most recent implicit cursor as the SQL cursor, which always has the attributes like %FOUND, %ISOPEN, %NOTFOUND, and %ROWCOUNT. The following table provides the description of the most used attributes</p> <p>Explicit cursor:</p> <p>Explicit cursors are programmer defined cursors for gaining more control over the context area. An explicit cursor should be defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row.</p> <p>Working with an explicit cursor involves four steps:</p> <ul style="list-style-type: none">• Declaring the cursor for initializing in the memory Cursor cursor_name IS select_statement;• Opening the cursor for allocating memory Open cursor name;• Fetching the cursor for retrieving data Fetch cursor name INTO variable1,variable2...;• Closing the cursor to release allocated memory Close cursor name;	(Implicit Cursor :2 marks; Explicit Cursor : 2 marks)

3.		Attempt any FOUR of the following :	Marks 16
	(a)	Describe data abstraction with neat diagram.	4M
	Ans:	<p>Data Abstraction: Data abstraction can be defined as the process of hiding the complexity of data and representing the data which needs to be shown to users.</p> <div style="text-align: center;">  </div> <p>Explanation:</p> <ol style="list-style-type: none"> Physical Level: <ul style="list-style-type: none"> It is lowest level of abstraction. This level defines lowest complicated data structure of database system. This level hidden from user. It defines how the data are stored. Logical Level: <ul style="list-style-type: none"> The level next to physical level is called logical level. This level defines what data stored in the database and what the relationships among these data are. Fully decides the structure of the entire database. View Level: <ul style="list-style-type: none"> This level is used to show the part of database to user. There is more complexity in physical as well as logical level so user should not interact with complicated database. So different view of database can be created for user to interact with database easily. 	(Data Abstraction description : 3 marks, Diagram:1 mark)



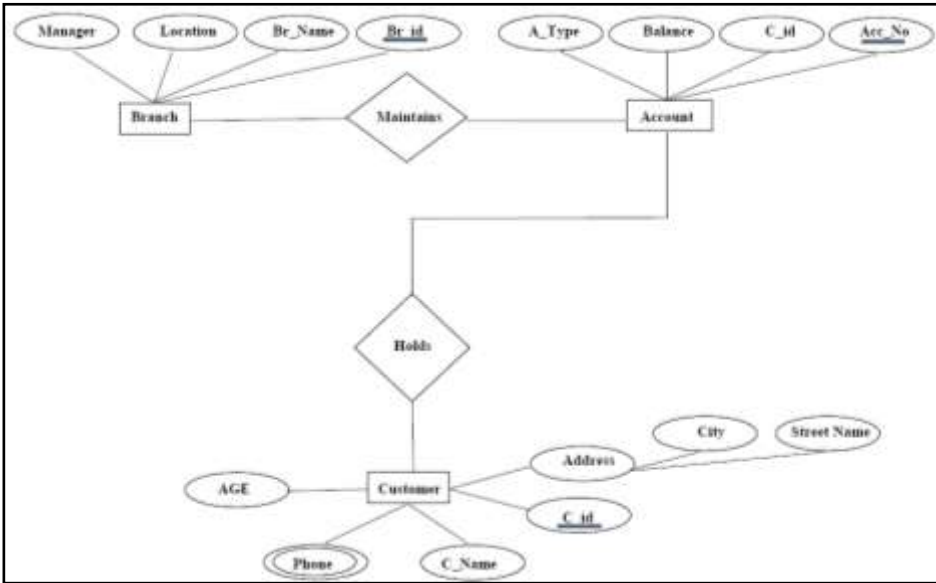
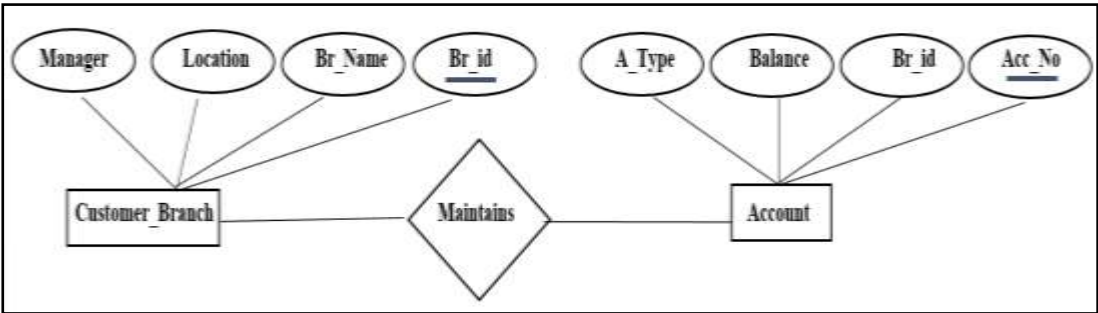
	(b)	State and explain 1NF and 2NF with example.	4M																																												
	Ans:	<p>First Normal Form (1NF): A relation R is said to be in first normal form (1NF) if the domain of all attributes of R are atomic.</p> <p>Example: Supplier (sno, sname, location, pno, qty)</p> <table><tr><th>SNO</th><th>SNAME</th><th>LOCATION</th><th>PNO</th><th>QTY</th></tr><tr><td>S1</td><td>Abc</td><td>Mumbai</td><td>P1</td><td>200</td></tr><tr><td>S2</td><td>Pqr</td><td>Pune</td><td>P2</td><td>300</td></tr><tr><td>S3</td><td>Lmn</td><td>Delhi</td><td>P1</td><td>400</td></tr></table> <p>The above relation is in 1NF as all the domains are having atomic value. But it is not in 2NF.</p> <p>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.</p> <p>Example: In the above relation SNAME, LOCATION depends on SNO and QTY 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.</p> <div><div>SP</div><table><tr><th>SNO</th><th>SNAME</th><th>LOCATION</th></tr><tr><td>S1</td><td>Abc</td><td>Mumbai</td></tr><tr><td>S2</td><td>Pqr</td><td>Pune</td></tr><tr><td>S3</td><td>Lmn</td><td>Delhi</td></tr></table></div> <div><div>Supplier</div><table><tr><th>SNO</th><th>PNO</th><th>QTY</th></tr><tr><td>S1</td><td>P1</td><td>200</td></tr><tr><td>S2</td><td>P2</td><td>300</td></tr><tr><td>S3</td><td>P1</td><td>400</td></tr></table></div> <p>(1NF: 2 marks, 2NF :2 marks)</p>	SNO	SNAME	LOCATION	PNO	QTY	S1	Abc	Mumbai	P1	200	S2	Pqr	Pune	P2	300	S3	Lmn	Delhi	P1	400	SNO	SNAME	LOCATION	S1	Abc	Mumbai	S2	Pqr	Pune	S3	Lmn	Delhi	SNO	PNO	QTY	S1	P1	200	S2	P2	300	S3	P1	400	
SNO	SNAME	LOCATION	PNO	QTY																																											
S1	Abc	Mumbai	P1	200																																											
S2	Pqr	Pune	P2	300																																											
S3	Lmn	Delhi	P1	400																																											
SNO	SNAME	LOCATION																																													
S1	Abc	Mumbai																																													
S2	Pqr	Pune																																													
S3	Lmn	Delhi																																													
SNO	PNO	QTY																																													
S1	P1	200																																													
S2	P2	300																																													
S3	P1	400																																													
	(c)	Explain ACID properties of transaction.	4M																																												
	Ans:	<p>1. Atomicity. It implies that the transactions are atomic. It means that either all transactions are executed and reflected properly in the database, or none them.</p> <p>2. Consistency. The consistency property of the transaction implies that if the transaction was in a consistent state before the start of the transaction, then on termination of transaction, the database should also be in a consistent state.</p> <p>3. Isolation. Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions T_i and T_j , it appears to T_i that either T_j finished execution before T_i started, or T_j started execution after T_i finished. Thus,</p>	(4 Properties: 1 mark each)																																												



		each transaction is unaware of other transactions executing concurrently in the system.	
		4. Durability. After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.	
	(d)	Consider the following schema : Depositor (ACC_no, Name, PAN, Balance) Create a view on depositor having attributes (ACC_no, PAN) where balance is greater than 1,00,000.	4M
	Ans:	Create view v1 as select ACC_no, PAN from depositor where balance>100000;	(For correct command: 4marks)
	(e)	Explain for loop in PL/SQL with example.	4M
	Ans:	{{**Note: Any other relevant example shall be considered. **}} For Loop : <ul style="list-style-type: none">• The FOR LOOP statement specifies a range of integers, then execute a sequence of statements once for each integer in the range.• To reverse the range , “REVERSE” keyword is used. The syntax is : FOR i IN lower_bound..upper_bound LOOP <loop_body><loop body/statements> END LOOP; OR For counter IN [reverse] lower_bound..upper_bound LOOP <loop body/statements>; END LOOP; Example: begin For a in 10..20 loop --output 10 to 20 elemets dbms_output.put_line('value of a : ' a); End Loop; End;	(Loop Explanation : 2marks, Example : 2marks)
	(f)	Describe shared lock and exclusive locking strategy in brief.	4M
	Ans:	1. Shared Lock. 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	(shared lock:2 marks, exclusive



		<p>could be set for table or table row. Lock is released or unlocked at the end of transaction. To lock table in shared mode one can use following syntax: Lock table table_name in <i>Share</i> Mode;</p> <p>2. Exclusive Lock: If a transaction T_i has obtained an exclusive-mode lock (denoted by X) on item Q, then T_i can both read and write Q. Exclusive Lock is provided to the writers of the data. When this lock is set on a object or transaction, it means that only writer, who has set the lock can change the data, and if other users cannot access the locked object. Lock is released at the end of change in transaction. To lock table in shared mode one can use following syntax: Lock table table_name in <i>Exclusive</i> Mode;</p>	lock: 2marks)
4.		Attempt any FOUR of the following :	Marks 16
	(a)	Explain any four functions of database administrator.	4M
	Ans:	<p>Ans: Functions of DBA:</p> <p>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.</p> <p>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.</p> <p>3. Schema and physical organization modification: DBA writes set of definitions to modify the database schema or description of physical storage organization.</p> <p>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.</p> <p>5. 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.</p> <p>6. Routine Maintenance: Some of the routine maintenance activities of a DBA is given below.</p> <ul style="list-style-type: none">• Taking backup of database periodically• Ensuring enough disk space is available all the time.• Monitoring jobs running on the database.• Ensure that performance is not degraded by some expensive task submitted by some users.	(Explanation of any four functions: 1mark each)

	(b)	Draw an E-R diagram for customer branch and account relationship.	4M
	Ans:	<p>{{**Note: Any other relevant diagram shall be considered**}}</p>  <p style="text-align: center;">OR</p> 	(Appropriate diagram-4marks)
	(c)	Explain any four integrity constraints.	4M
	Ans:	<p>{{** Note: Either syntax or example can be considered**}}</p> <p>Integrity constraints: Not Null constraint, CHECK constraint, Primary Key constraint, Unique Constraint, Referential Integrity Constraint</p> <p>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.</p> <p>Syntax :</p> <pre>CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE NOT NULL);</pre> <p>Example:</p>	(Any Four Constraints explanation :1mark each)



	<p>SQL>CREATE TABLE STUDENT (ROLL_NO NUMBER(5), NAME VARCHAR2(20) <i>NOT NULL</i>);</p> <p>2. CHECK: The constraint defines a condition that each row must satisfy. A single column can have multiple check condition.</p> <p>Syntax:-</p> <p>CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME CHECK <CONDITION>);</p> <p>Example:</p> <p>SQL> CREATE TABLE EMP (ID NUMBER(5), NAME VARCHAR2(10), SAL NUMBER(10) CONSTRAINT CHK_SAL <i>CHECK</i> (SAL>15000));</p> <p>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.</p> <p>Syntax:</p> <p>CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME PRIMARY KEY);</p> <p>Example:</p> <p>SQL> CREATE TABLE EMP (ID NUMBER (5) CONSTRAINT ID_PK PRIMARY KEY, NAME VARCHAR2 (10), SAL NUMBER (10));</p> <p>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.</p> <p>Syntax:</p> <p>CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME UNIQUE);</p> <p>Example:</p> <p>CREATE TABLE PERSONS (P_ID NUMBER CONSTRAINT P_UK UNIQUE,</p>	
--	--	--



		<p>FIRSTNAME VARCHAR2(20), CITY VARCHAR2(20));</p> <p>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.</p> <p>Syntax:</p> <p>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);</p> <p>Example:</p> <p>CREATE TABLE DEPARTMENT (EMP_ID NUMBER(5) REFERENCES EMP(EMP_ID), DNO NUMBER(3));</p>																	
	(d)	Explain set operators with example.	4M																
	Ans:	<p>Set operators combine the results of two component queries into a single result. Queries containing set operators are called as compound queries. Set operators in SQL are represented with following special keywords as: Union, Union all, intersection & minus. Consider data from two tables emp1 and emp2 as</p> <div><table><tr><th>Emp1</th><th>Emp2</th></tr><tr><td><table><tr><th>Ename</th></tr><tr><td>abc</td></tr><tr><td>xyz</td></tr><tr><td>lmn</td></tr></table></td><td><table><tr><th>Ename</th></tr><tr><td>pqr</td></tr><tr><td>xyz</td></tr></table></td></tr></table><p>1) Union : The Union of two or more sets contains all elements, which are present in either or both. Union works as or. E.g. select ename from emp1 union select ename from emp2; The output considering above data is : Output:</p><table><tr><th>Ename</th></tr><tr><td>abc</td></tr><tr><td>xyz</td></tr><tr><td>lmn</td></tr><tr><td>pqr</td></tr></table></div>	Emp1	Emp2	<table><tr><th>Ename</th></tr><tr><td>abc</td></tr><tr><td>xyz</td></tr><tr><td>lmn</td></tr></table>	Ename	abc	xyz	lmn	<table><tr><th>Ename</th></tr><tr><td>pqr</td></tr><tr><td>xyz</td></tr></table>	Ename	pqr	xyz	Ename	abc	xyz	lmn	pqr	(Each operator: 1 mark)
Emp1	Emp2																		
<table><tr><th>Ename</th></tr><tr><td>abc</td></tr><tr><td>xyz</td></tr><tr><td>lmn</td></tr></table>	Ename	abc	xyz	lmn	<table><tr><th>Ename</th></tr><tr><td>pqr</td></tr><tr><td>xyz</td></tr></table>	Ename	pqr	xyz											
Ename																			
abc																			
xyz																			
lmn																			
Ename																			
pqr																			
xyz																			
Ename																			
abc																			
xyz																			
lmn																			
pqr																			



		<p>2) Union all : The Union of 2 or more sets contains all elements, which are present in both, including duplicates.</p> <p>E.g. select ename from emp1 union all select ename from emp2; The output considering above data is: Output:</p> <table><tr><td>Ename</td></tr><tr><td>abc</td></tr><tr><td>xyz</td></tr><tr><td>lmn</td></tr><tr><td>pqr</td></tr><tr><td>xyz</td></tr></table> <p>3) Intersection: The intersection of two sets includes elements which are present in both. E.g. select ename from emp1 intersect select ename from emp2; The output considering above data is:</p> <table><tr><td>Ename</td></tr><tr><td>xyz</td></tr></table> <p>4) Minus: The minus of two sets includes elements from set1 minus elements of set2. E.g. select ename from emp1 minus select ename from emp2; The output considering above data is:</p> <table><tr><td>Ename</td></tr><tr><td>abc</td></tr><tr><td>lmn</td></tr></table>	Ename	abc	xyz	lmn	pqr	xyz	Ename	xyz	Ename	abc	lmn	
Ename														
abc														
xyz														
lmn														
pqr														
xyz														
Ename														
xyz														
Ename														
abc														
lmn														
	(e)	Define index. Explain types of index.	4M											
	Ans:	<p>Index: An index is a schema object that can speed up the retrieval of rows by using pointer. An index provides direct and fast access to rows in a table. Indexes are created explicitly Or automatically. Indexes are used to speed up searches/queries. Types of Index:</p> <ul style="list-style-type: none">1. Simple index2. Unique index3. Composite index <p>1. Simple index(Single column): An index created on single column of a table is called a Simple Index.</p>	<p>(Definition: 2marks, any two types: 2 marks)</p>											



	<p>Syntax: Create index index_name on <tablename><column name>; E.g.: Create index idx on employee (empno);</p> <p>2. Unique indexes are used not only for performance, but also for data integrity. A unique index does not allow any duplicate values to be inserted into the table.</p> <p>Syntax : Create unique index index_name on table_name(column_name); E.g.: Create unique index index_empno on emp(empno);</p> <p>3. Composite (concatenated): Indexes that contain two or more columns from the same table which are useful for enforcing uniqueness in a table column where there's no single column that can uniquely identify a row.</p> <p>Syntax: Create index index_name on <tablename><Column_name1, Column_name2>; E.g.: Create index idx on employee (ename, empno);</p> <p style="text-align: center;">OR</p> <p>Automatic Index: - An index created automatically when a user defines a primary key or unique constraint in a table definition.</p> <p>Manual Index: - User can create non – unique index on a attribute to speed up access to the rows. It has 3 types</p> <ul style="list-style-type: none">• Simple Index• Unique Index• Composite Index	
(f)	Write a PL/SQL program using while loop to display n even numbers.	4M
Ans:	<pre>/** Note: Any relevant logic can be considered**} 1. To Display even numbers till n SQL> DECLARE 2 I NUMBER:= &I; 3 CNT NUMBER:=1; 4 BEGIN 5 WHILE(CNT <= I) LOOP 6 IF(MOD(CNT,2)=0) THEN 7 DBMS_OUTPUT.PUT_LINE(CNT); 8 END IF; 9 CNT:=CNT+1; 10 END LOOP; 11 END; Enter value for i: 10 2 4 6</pre>	(Correct logic:2 marks, correct syntax:2 marks)



		<p>8 10 PL/SQL procedure successfully completed.</p> <p style="text-align: center;">OR</p> <p>2. To Display n even numbers SQL> DECLARE 2 OP NUMBER:= 2; 3 I NUMBER:=&I; 4 BEGIN 5 WHILE(I>0) LOOP 6 IF(MOD(OP,2)=0)THEN 7 DBMS_OUTPUT.PUT_LINE(OP); 8 END IF; 9 I:=I-1; 10 OP:=OP+2; 11 END LOOP; 12 END; Enter value for i: 10 2 4 6 8 10 12 14 16 18 20 PL/SQL procedure successfully completed.</p>	
5.		Attempt any FOUR of the following :	Marks 16
	(a)	Explain three tier architecture with suitable diagram.	4M
	Ans:	<p>{{**Any other relevant diagram shall be considered}}</p> <p>The three tier architecture includes the following:</p> <ul style="list-style-type: none">• Clients contain GUI interfaces and some additional application-specific business rules.• The intermediate server also known as application server accepts requests the clients, processes the requests and sends database commands to the database server. It acts as a conduit for passing (partially processed data from the database server to the clients, when it may be processed further and filtered to be presented to users in GUI format.• Thus the user interfaces, application rules and the database comprises of the three tier architecture	(Explanation: 2marks, diagram:2 marks)

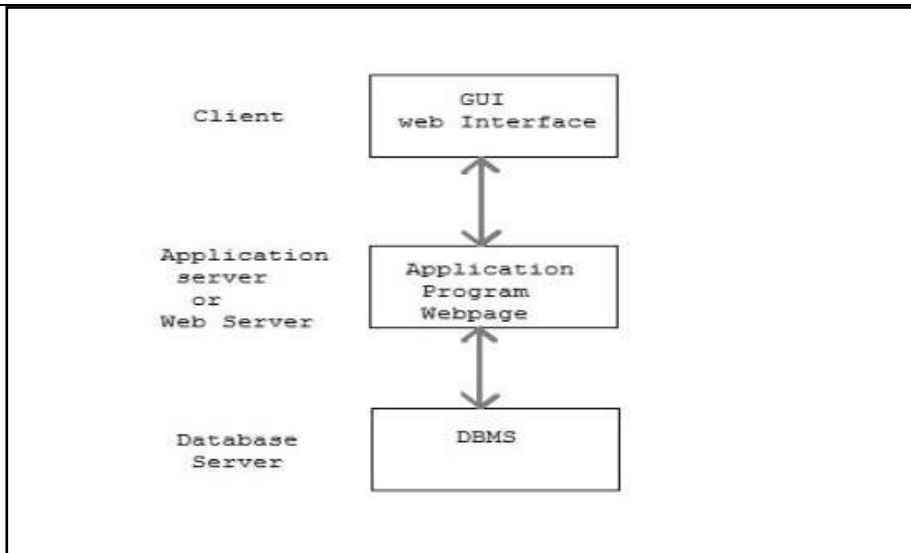


Figure: Three Tier Architecture

	(b)	Explain on delete cascade clause with example.	4M
	Ans:	<p>The term On Delete cascade referential action is used for a foreign key to delete data from multiple related tables. On delete cascade clause allows automatic deletion of child records when parent record is deleted.</p> <p>When a relation is created between two tables, it is permissible to delete records in child table, but a vice-versa operation is not allowed i.e. deleting records from parent table when it is referenced from some other table is not allowed.</p> <p>E.g. :</p> <p>Using Create Statement:</p> <p>Create table emp (ename varchar2(10), eid number(5), deptno number(10) references dept(deptno) on delete cascade);</p> <p>Using Alter Statement:</p> <p>Alter table Emp add constraint foreign key emp(deptno) references dept(deptno) on delete cascade;</p> <p>Here dept table is the parent table which contain column, deptno as primary key and this column is referred in Emp table i.e child table as foreign key with on delete cascade option. So if any deletion occurs at dept table, related data get deleted from Emp table simultaneously</p>	(Explanation: 3 marks, example: 1 mark)
	(c)	<p>Consider following schema :</p> <p>Employee (emp_no, emp_name, dept, salary, designation, dept_location)</p> <p>Solve following queries :</p> <p>(i) List all managers in Mumbai location.</p> <p>(ii) Set salary of all 'project leaders' to ₹ 70,000.</p> <p>(iii) List employees with having alphabet 'A' as second letter in their name.</p> <p>(iv) Display details of those employees who work in Mumbai or Chennai.</p>	4M



	Ans:	i) SQL> Select emp_name from Employee where designation='manager' and dept_location= 'Mumbai'; ii) SQL> Update Employee Set salary = 70000 Where designation = 'project leader'; iii) SQL> Select emp_name from Employee where emp_name like ' _A%'; iv) SQL> Select * from Employee where dept_location IN ('Mumbai', 'Chennai'); OR SQL> Select * from Employee where dept_location='Mumbai' or dept_location='Chennai';	Each correct query: 1mark each)
	(d)	Describe Grant and Revoke commands.	4M
	Ans:	1. GRANT is a DCL command used to provide access or privileges on the database object to the users. Syntax of GRANT command: GRANT privilege_name ON object_name TO {user_name PUBLIC role_name} [WITH GRANT OPTION]; Example: Grant update, select on emp to u1 2. REVOKE is also a DCL command which removes user access rights or privileges to the database objects. Syntax of REVOKE command REVOKE privilege_name ON object_name FROM {user_name PUBLIC role_name}; Example: Revoke update on emp from u1;	(Explanation of each command with syntax or example: 2 marks each)
	(e)	Define sequences. Create sequence for student table.	4M
	Ans:	A Sequence is a user created database object. It is used to generate unique integers. The sequence is generated and incremented or decremented by an internal oracle routine. This save a time reduces application code. Sequence numbers are stored and generated independently of tables. It can be used for multiple tables. 1) Syntax to Create a sequence : Create sequence < sequence name > [Incremented by <integer value>][start with <integer value>][Maxvalue <integer value>] [minvalue <integer value>][cycle/ nocycle]; INCREMENT BY: Specifies the interval between sequence numbers. It can be any positive or negative value but not zero. If this clause is omitted the default value is 1. MINVALUE: Specifies the sequence minimum value. MAXVALUE: Specifies the maximum value that a sequence can generate. CYCLE: Specifies that the sequence continues to generate repeat values after reaching either its maximum.	(Definition: 2marks, Creating: 2 marks)



		NOCYCLE: Specifies that a sequence cannot generate more values after reaching the maximum value. Create sequence addr_sqe increment by 1 start with 1 minvalue 1 maxvalue 999 cycle; To Create sequence for student Table: SQL> Create sequence stud_seq start with 1 increment by 1 maxvalue 50 nocycle;													
	(f)	Define database trigger. Compare database trigger and procedures.	4M												
		<p>Database trigger is a PL/SQL block that is executed on an event in the database. The event is related to a particular data manipulation of a table such as inserting, deleting or updating a row of a table.</p> <p>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;</p> <p style="text-align: center;">Comparison between triggers and procedures</p> <table><tr><th>Triggers</th><th>Procedures</th></tr><tr><td>Triggers are called implicitly by oracle</td><td>Procedures are called explicitly.</td></tr><tr><td>Trigger can only be executed whenever an event (insert, delete, and update) is fired on the table on which it is defined.</td><td>Procedure can be executed whenever it is needed with the help of the exec command,</td></tr><tr><td>Trigger does not require parameters for execution</td><td>Procedure can take input parameters</td></tr><tr><td>Trigger returns exception or status of current event.</td><td>Procedures may return values</td></tr><tr><td>Triggers cannot be scheduled like procedures</td><td>Procedures can be scheduled through a job to execute on a predefined time</td></tr></table>	Triggers	Procedures	Triggers are called implicitly by oracle	Procedures are called explicitly.	Trigger can only be executed whenever an event (insert, delete, and update) is fired on the table on which it is defined.	Procedure can be executed whenever it is needed with the help of the exec command,	Trigger does not require parameters for execution	Procedure can take input parameters	Trigger returns exception or status of current event.	Procedures may return values	Triggers cannot be scheduled like procedures	Procedures can be scheduled through a job to execute on a predefined time	(Definition: 2marks, Comparison: 2marks)
Triggers	Procedures														
Triggers are called implicitly by oracle	Procedures are called explicitly.														
Trigger can only be executed whenever an event (insert, delete, and update) is fired on the table on which it is defined.	Procedure can be executed whenever it is needed with the help of the exec command,														
Trigger does not require parameters for execution	Procedure can take input parameters														
Trigger returns exception or status of current event.	Procedures may return values														
Triggers cannot be scheduled like procedures	Procedures can be scheduled through a job to execute on a predefined time														
6.		Attempt any FOUR of the following :	Marks 16												
	(a)	Describe data mining and data warehousing.	4M												
	Ans:	<p>1. Data Mining: The process of extracting valid, previously unknown, comprehensible, and actionable information from large databases and using it to make crucial business decisions is known as Data Mining. Data mining is concerned with the analysis of data and the use of software techniques for</p>	(Data Mining :2marks , Data Warehouse												



		finding hidden and unexpected patterns and relationships in sets of data. The focus of data mining is to find the information that is hidden and unexpected. 2.Data Warehouse: Data warehouse is defined as "A subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process." Data warehouse contains huge amount of data which can be in structured, semi structured or unstructured fashion. In Data warehouse the volume of data is considered and in most of the case the size of data is in XETABYTE.	:2marks)										
	(b)	Explain any 4 types of attributes.	4M										
	Ans:	Types of attributes: 1. Simple attributes: An attributes which cannot be divided into subparts are known as Simple attributes. E.g. RollNo. 2. Composite attributes: It can be further divided into subparts (that is, other attributes). e.g Name can be divided as First_name,Middle_name,Last_name 3. Single-valued attribute: The attributes that hold a single value for a single entity are single-valued attributes. E.g. Branch 4. Multivalued attributes: There may be instances where an attribute has a set of values for a specific entity. E.g. TelephoneNo 5. Derived attributes: The value for such attribute can be derived from the values of other related attributes or entities. E.g. Age attribute can be derived from attribute DOB 6. Key attributes: An attribute which denotes the primary key of an entity set. example- student_enrollment_no in student table.	(Any Four attributes: 1mark each)										
	(c)	Explain drop and delete commands with syntax. State the difference between them.	4M										
	Ans:	Drop command: When we need to delete a database object from a database the SQL DROP statement is use to remove such database object with its structure. This command also removes all data indexes, triggers constraints and permissions and specifications for those object. Syntax: Drop dabase_object <object name>; Example: Drop table emp; Drop sequence sql; Delete Command: Delete command is used to delete the rows in the table. DELETE removes one or more records from given table. Syntax: DELETE from<table_name> [where <conditions>]; Delete From student where roll_no = 1234; Difference between DROP and DELETE <table><tr><th>DROP</th><th>DELETE</th></tr><tr><td>It is a DDL Statement.</td><td>It is DML statement.</td></tr><tr><td>It removes entire table at once from the disk.</td><td>It deletes one or more rows at a time</td></tr><tr><td>Entire structure of database object is removed from disk</td><td>Table structure remains on the disk with other rows.</td></tr><tr><td>Syntax :</td><td>Syntax :</td></tr></table>	DROP	DELETE	It is a DDL Statement.	It is DML statement.	It removes entire table at once from the disk.	It deletes one or more rows at a time	Entire structure of database object is removed from disk	Table structure remains on the disk with other rows.	Syntax :	Syntax :	(Explanatio n:1mark each, differences: 2marks)
DROP	DELETE												
It is a DDL Statement.	It is DML statement.												
It removes entire table at once from the disk.	It deletes one or more rows at a time												
Entire structure of database object is removed from disk	Table structure remains on the disk with other rows.												
Syntax :	Syntax :												



		Drop database_object <object name>;	DELETE from <table_name> [where <conditions>];	
		Example : Drop table employee;	Example : DELETE from employee where sal>2000;	
	(d)	Explain any four string functions with example.		4M
	Ans:	<p>String functions along with examples are:</p> <p>i) Lower(char)- Returns the input string with all letters in lower case. Example: SQL>Select lower ('RAJESH') from dual; Output: rajesh</p> <p>ii) Upper(char)-Returns the input string with all letters in upper case. Example: SQL>Select upper ('rajesh') from dual; Output: RAJESH</p> <p>iii) Ltrim(char,set)- It removes or trims from left of character string. Example: SQL>Select Ltrim('university','univ') from dual; Output: ersity</p> <p>iv) Rtrim(char,set)- It removes or trims from right of character string. Example: SQL>Select Rtrim('university','sity') from dual; Output: univer</p> <p>v) Length(char)-It returns length of character string. Example:SQL> Select length('University') from dual; Output:10</p> <p>vi) CONCAT(str1,str2,...)-Returns the string that results from concatenating the arguments. Example: Select Concat('employee', 'name') from dual; Output: employeename</p> <p>vii) LPAD(str, len, padstr)-Returns the string str, left-padded with the string <i>padstr</i> to a length of <i>len</i> characters. Example: Select lpad(ename,10,'*') from emp where empno=7782;</p> <p>viii) RPAD(str,len,padstr)-Returns the string str, right-padded with the string <i>padstr</i> to a length of <i>len</i> characters. Example: Select rpad(ename,10,'*') from emp where empno=7782;</p> <p>viii) Substr(Char,m,n)-It returns a portion of char, beginning at a character m, n character long. Example: Select substr('College',3,4) from dual; Output: lleg</p>		(Any four functions: 1mark each)



	(e)	Explain PL/SQL block structure.	4M
	Ans:	<div style="border: 1px solid black; padding: 10px; text-align: center;"><p>DECLARE Declaration of variables BEGIN PL/SQL execution statements EXCEPTION PL/SQL Exception handles block END;</p></div> <p style="text-align: center;">PL/SQL Block Structure</p> <p>PL/SQL block structure follow divide and conquer approach to solve the problem stepwise.</p> <p>Each block consists of 3 sub parts:</p> <ol style="list-style-type: none">1) Declaration Section: This section starts with key word DECLARE. It is optional section and defines all variables, cursors, sub-programs and other elements to be used in the program.2) Executable Section: This section is enclosed between the keywords BEGIN and END. It is a mandatory section. It consists of executable PL/SQL statements of the program. It should have at least one executable line of code which may be a NULL command to indicate that nothing should be executed.3) Exceptional Handling Section: This section starts with key word exception. This section is again optional and contains exception that handles errors in the program.	(Explanation: 3 marks, Diagram: 1 mark)
	(f)	Write PL/SQL program to calculate factorial of a given number.	4M
	Ans:	<pre>/** Note: Any relevant logic can be considered**} SQL> DECLARE 2 FACT NUMBER:= 1; 3 I NUMBER:=&I; 4 BEGIN 5 WHILE(I>0) LOOP 6 FACT:= FACT*I; 7 I:=I-1; 8 END LOOP; 9 DBMS_OUTPUT.PUT_LINE(FACT); 10 END; 11 / Enter value for i: 5 120 PL/SQL procedure successfully completed.</pre>	(Correct program: 4 marks)