



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:	12
	i)	List any four advantages of DBMS.	2M
	Ans:	Advantages of DBMS: 1. Reduction in Redundancy 2. Avoiding Inconsistency 3. Maintaining Integrity 4. Sharing of data 5. Enforcement of Security 6. Transaction support.	<i>(Any 4 Advantages: 1/2 mark each)</i>
	ii)	List any two data model.	2M
	Ans:	Data Models: 1. Hierarchical Model 2. Network Model 3. Relational Model 4. Entity Relationship Model	<i>(Any 2 models: 1 mark each)</i>
	iii)	What is multi-valued dependency?	2M
	Ans:	Multivalued dependencies occur when the presence of one or more rows in a table implies the presence of one or more other rows in that same table. OR A multivalued dependency (MVD): In a relational table R with columns A, B and C then R. A ->> R.B (column A multi-determines column B) is true if and only if the set of B -values matching a given pair of A-values and C- values in R depends only on A-	<i>(Definition 2 marks)</i>



	value and is independent of C-value OR The multivalued dependency $\alpha \twoheadrightarrow \beta$ holds on R if, in any legal relation $r(R)$, for all pairs of tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r such that $t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$ $t_3[\beta] = t_1[\beta]$ $t_3[R - \beta] = t_2[R - \beta]$ $t_4[\beta] = t_2[\beta]$ $t_4[R - \beta] = t_1[R - \beta]$	
iv)	List four DDL commands.	2M
Ans:	DDL commands: 1. Create 2. Alter 3. Rename 4. Drop 5. Truncate	<i>(Any 4 DDL commands: 1/2 mark each)</i>
v)	List DCL commands any four.	2M
Ans:	DCL commands: 1. Grant 2. Revoke 3. Commit 4. Rollback 5. Savepoint	<i>(Any 4 DCL commands: 1/2 mark each)</i>
vi)	What is view?	2M
Ans:	View: A view is a logical extract of a physical relation i.e. it is derived from any base relation. OR View: Views are virtual relations mainly used for security purpose, and can be provided on request by a particular user.	<i>(Definition of view: 2 marks)</i>
vii)	Define cursor? List the two types of cursor.	2M



	Ans:	Cursor: A cursor is a temporary work area created in the system memory when a SQL statement is executed. Types of Cursor: 1.Implicit Cursor 2.Explicit Cursor	<i>(Delimitation of cursor:1 mark, each type:1/2 mark)</i>
	viii)	List different relational algebraic operators any four.	2M
	Ans:	<ol style="list-style-type: none"> 1. Selection(σ) 2. Projection (π) 3. Cartesian Product (X) 4. Natural Join(\bowtie) 5. Union(U) 6. Set Difference (-) 7. Rename (ρ) 	<i>(Any 4 relational operations or symbols:1/2 marks each)</i>
	b)	Attempt any two of the following:	8
	i)	Explain Distributed Database with example.	4M
	Ans:	<p>Distributed Database: Distributed database is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.</p> <p>A Distributed Database Management System (DDBMS) consists of a single logical database that is split into a number of fragments. Each fragment is stored on one or more computers under the control of a separate DBMS, with the computers connected by a communications network. Each site is capable of independently processing user requests that require access to local data and is also capable of processing data stored on other computers in the network.</p> <p>Example:</p> <p>Using distributed database technology, a bank may implement their database system on a number of separate computer systems rather than a single, centralized mainframe. The computer systems may be located at each local branch office: for example, Mumbai, Pune, Nagpur. A network linking the computer will enable the branches to communicate with each other, and DDBMS will enable them to access data stored at another branch office. Thus a client living in Pune can also check his/her account during the stay in Mumbai or Nagpur</p>	<i>(Definition of Distributed database:3 marks, example:1 mark(Any valid Example))</i>
	ii)	Explain DELETE and DROP Command with syntax and example.	4M
	Ans:	<p>DELETE Command: The SQL DELETE Query is used to delete the existing records from a table. You can use WHERE clause with DELETE query to delete selected rows, otherwise all the records would be deleted.</p> <p>Syntax:</p>	<i>(DELETE Command: 2 marks ,DROP</i>



	<p>DELETE FROM table_name WHERE [condition]; Example: 1) To Delete record from customer table with Name as JACK DELETE FROM CUSTOMER WHERE NAME = 'JACK';</p> <p>DROP Command: The SQL DROP Command is use to delete all records and schema of the table.</p> <p>Syntax: DROP Table <table name>;</p> <p>Example: Drop table emp;</p>	<p><i>command: 2 marks)</i></p>
iii)	<p>Explain the exception handling with its two type.</p>	<p>4M</p>
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 ...executable statement;</p> <p>EXCEPTION WHEN ex_name1 THEN ; Error handling statements/user defined action to be carried out; END;</p> <p>Types of Exception:</p> <p>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.</p> <p>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>	<p><i>(Exception explanation: 2 marks, 2 types: 1 mark each)</i></p>
2.	<p>Attempt any four of the following :</p>	<p>16</p>
a)	<p>Explain strong entity and weak entity set.</p>	<p>4M</p>



	iv) Display the total number of employees whose dept is '50'.	
Ans:	i) Select emp-id from Employee where emp-city='Pune' or emp-city='Nagpur'; OR i) Select emp-id from Employee where emp-city IN('Pune','Nagpur'); ii) Select * from Employee where join-date>'02-Jul-2007'; iii) Update Employee set emp-name='Aarav' where emp-name='Ajit'; iv) Select count(emp-dept) from Employee where emp-dept=50; OR iv) Select count(*) from Employee group by emp-dept having emp-dept=50;	<i>(Each correct query: 1 mark)</i>
d)	Consider following schema. Employee (empname, empid, dob, salary, job) Create a view on employee having attribute (empname, empid, dob, salary, job) where salary is greater than 20,000.	4M
Ans:	Create view EMPVIEW as select empname, empid, dob, salary, job from employee where salary>20000; OR CREATE VIEW EMPVIEW AS SELECT * FROM EMPLOYEE WHERE SALARY > 20000;	<i>(Correct query: 4 marks)</i>
e)	Explain Database security with its requirements.	4M
Ans:	Database Security: Database security refers to the collective measures used to protect and secure a database or database management software from illegal use and malicious threats and attacks. Database security covers and enforces security on all aspects and components of databases like Data stored in database, Database server, DBMS. Data Security Requirements: 1. Authentication: System verifies a user's identity. 2. Authorization: Which database operations that user may perform (like read, update, drop etc.) and which data objects that user may access. 3. Secure Storage of Sensitive Data: Once confidential data has been entered, its integrity and privacy must be protected on the databases and servers wherein it resides. 4. Integrity: Data integrate means that data is protected from deletion and corruption 5. Availability: A secure system makes data available to authorized users, without	<i>(Database security: 2 marks, Requirements : 2 marks)</i>



	delay.	
	6. Confidentiality: A secure system ensures the confidentiality of data. This means that it allows individuals to see only the data they are supposed to see.	
f)	Explain entity integrity constraints with syntax and example.	4M
Ans:	<p>There are two Entity constraints: 1. Primary Key constraint 2. Unique Constraint</p> <p>1. Primary Key constraint: It is use to avoid redundant/duplicate value entry within the row of specified column in table. It restricts null values too.</p> <p>Syntax: CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME PRIMARY KEY);</p> <p>Example: SQL> CREATE TABLE EMP (ID NUMBER (5)CONSTRAINT ID_PK PRIMARY KEY, NAME VARCHAR2 (10), SAL NUMBER (10));</p> <p>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.</p> <p>Syntax: CREATE TABLE TABLE_NAME (COLUMN_NAME DATA_TYPE, COLUMN_NAME DATA_TYPE CONSTRAINT CONSTRAINT_NAME UNIQUE);</p> <p>Example: CREATE TABLE PERSONS (P_ID NUM CONSTRAINT P_UK UNIQUE , FIRSTNAME VARCHAR(20), CITY VARCHAR(20));</p>	(<i>Primary key constraint:2 marks, Unique key constraint:2 marks</i>)
3.	Attempt any four of the following:	16
a)	Give the syntax and example of CREATE and RENAME Commands.	4M
Ans:	<p>The Syntax for the CREATE TABLE command is: CREATE TABLE <table_name> (<column_name1>< datatype>(size), <column_name2>< datatype> (size),</p>	(<i>Each Command Syntax with example: 2 marks</i>)



<column_nameN>< datatype> (size));

For Example: To create the employee table, the statement would be like,

```
CREATE TABLE employee
(emp_id number(5),
name char(20),
deptno number(2),
dob date,
salary number(10,2),
address varchar(30) );
```

The Syntax for the RENAME TABLE command is:

```
RENAME <old_table_name> To < new_table_name>;
```

For Example: To change the name of the table employee to my_employee, the query would be like,

```
RENAME employee TO my_employee;
```

b) Explain PL/SQL Block structure.

4M

Ans:

Block structure of PL/SQL:

DECLARE(Optional)

Declaration Section: all variables, constant cursors etc that are reference in the executable and declarative sections.

BEGIN(Mandatory)

Execution Section: Contains SQL and PL/SQL statements to manipulate data in the block and database.

EXCEPTION(Optional)

Specifies the actions to be perform when errors and abnormal conditions are raised during execution of code.

END(Mandatory);

Each PL/SQL program consists of SQL and PL/SQL statements which from a PL/SQL block.

A PL/SQL Block consists of following sections:

The Declaration section: Declaration of memory variables used later in begin section.

The Begin section: SQL executable statements for manipulating table data should be in

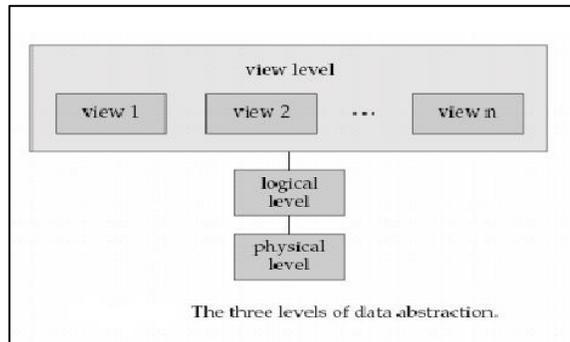
(Each block description: 1 mark, Diagram is optional)



	<p>BEGIN..END block. The Exception section: SQL and/or PL-SQL code to handle errors that may crop up during execution of the above code block. End : Mark the end of PL-SQL block</p>	
c)	Explain tuple relational calculus with example.	4M
Ans:	<p>Tuple relational calculus:-It is a non-procedural calculus. It describes information without giving a specific procedure for obtaining that information. A query in tuple calculus is expressed as $\{t \mid p(t)\}$ i.e. the set of all tuples (t) such that predicate (P[condition]) is true for 't'. We use t[a] to denote the value of tuple on attribute 'a' & we use 't ∈ R' to denote that tuple 't' is in relation 'R'. There are different symbols with specific meaning which can be used to write tuple calculus expression;-</p> <ol style="list-style-type: none"> 1. ∈ belong to 2. ∃ There exists 3. ∀ for all 4. ¬ not 5. => implies 6. ^ and 7. ∨ or <p>Example. Find records of employees where salary is more than 20000. $\{t \mid t \in \text{employee}(t[\text{salary}] > 20000)\}$</p>	<i>(Explanation : 2 marks, any one Example: 2 marks)</i>
d)	Explain the following terms with syntax and example: a) Simple Index b) Composite Index c) Unique Index.	4M
Ans:	<p>a) Simple Index: A simple index is one that is created based on only one column of a table. The basic syntax is as follows: CREATE INDEX index_name ON table_name (column_name); Example: CREATE INDEX ename_idx on emp (ename);</p> <p>b) Composite Index: A composite index is an index created on two or more columns of a table. The basic syntax is as follows: CREATE INDEX index_name on table_name (column1, column2); Example: CREATE INDEX en_idx on emp (ename,job);</p>	<i>(Explanation with syntax and example : 4 marks)</i>



	<p>c) Unique Indexes: 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. The basic syntax is as follows:</p> <p>CREATE UNIQUE INDEX index_name on table_name (column_name);</p> <p>Example: CREATE UNIQUE INDEX ename_idx on emp (ename);</p>	
e)	What is database Trigger? How to create Trigger?	4M
Ans:	<p>A trigger is a PL/SQL block structure which is fired when DML statements like Insert, Delete, Update is executed on a database table. A trigger is triggered automatically when an associated DML statement is executed.</p> <p>Syntax for Creating a Trigger</p> <pre>CREATE OR REPLACE TRIGGER trigger_name [BEFORE/AFTER] [INSERT/UPDATE/DELETE] ON Table_name [FOR EACH STATEMENT/FOR EACH ROW] [WHEN CONDITION] PL/SQL block</pre> <p>Example:</p> <pre>CREATE OR REPLACE TRIGGER trg1 BEFORE INSERT ON EMP FOR EACH ROW BEGIN IF :new.sal<=0 THEN Raise_application_error('Salary should be greater than 0'); END IF; END;</pre>	<p>(Description: 2 marks, Syntax OR Example: 2 marks)</p>
f)	Explain three levels of data abstraction with suitable diagram.	4M
Ans:	<p>Data abstraction is a process in which developer hides the complexity from the user to simplify the user interaction with the system. There are three levels of abstraction:-</p>	<p>(Diagram - 2 mark; Description - 2 marks,)</p>



Physical Level: - The Lowest level of data abstraction describes HOW the data are actually stored.

Logical Level: - The Next higher level of data abstraction describes WHAT the data are stored in the database, & what relationships exist among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures.

View Level: - The highest level of data abstraction describes only part of the entire database. It is user level. The view level of abstraction exists to simplify their interaction with the system. The system may provide many views for the same database.

4.	Attempt any four of the following:	16
a)	<p>Consider student schema(studid, studname, studaddr, studcity, studper)</p> <p>Write relational algebra expression of the following:</p> <p>i) Find the name of the student those who scored first class.</p> <p>ii) Find studid, studaddr from the student database.</p>	4M
Ans:	<p>i) Find the name of the student those who scored first class.</p> $\Pi_{studname} (\sigma_{studper \geq 60} (student))$ <p style="text-align: center;">OR</p> $\Pi_{studname} (\sigma_{studper = \text{"FIRST"}}(student))$ <p>ii) Find studid, studaddr from the student database.</p> $\Pi_{studid, studaddr} (student)$	(Each relational algebra expression: 2 marks)
b)	<p>Write a PL/SQL program to print even or odd number from given range (Accept number range from user).</p> <p>{**NOTE: any relevant program logic shall be considered**}</p>	4M

**Ans:**

```
DECLARE
  A NUMBER :=&A;
  B NUMBER :=&B;
  C NUMBER :=&C;
BEGIN
  IF(C=1) THEN
    FOR I IN A..B LOOP
      IF(MOD(I,2)=0) THEN
        DBMS_OUTPUT.PUT_LINE(I);
      END IF;
    END LOOP;
  ELSE
    FOR I IN A..B LOOP
      IF(MOD(I,2)=1) THEN
        DBMS_OUTPUT.PUT_LINE(I);
      END IF;
    END LOOP;
  END IF;
END;
```

OR**-- PL/SQL code to display even numbers**

```
DECLARE
  A NUMBER :=&A;
  B NUMBER :=&B;
  BEGIN

  FOR I IN A..B LOOP
    IF(MOD(I,2)=0) THEN
      DBMS_OUTPUT.PUT_LINE(I);
    END IF;
  END LOOP;
END;
```

-- PL/SQL code to display odd numbers

```
DECLARE
  A NUMBER :=&A;
  B NUMBER :=&B;
  BEGIN
  FOR I IN A..B LOOP
    IF(MOD(I,2)=1) THEN
      DBMS_OUTPUT.PUT_LINE(I);
```

*(Correct
Program : 4
marks)*



	<p>END IF; END LOOP; END;</p>	
c)	<p>Explain ALTER command with any two options.</p>	<p>4M</p>
Ans:	<p>The SQL ALTER TABLE command is used to modify the definition (structure) of a table by modifying the definition of its columns It can be used for</p> <ol style="list-style-type: none"> 1. To add any new column to a table 2. To change data type or size of already existing data column of a table. 3. To delete a column from a table 4. To add / drop constrains from column of a table. <p>The three options with ALTER command are:</p> <p>1. Add column:-We can add any number of columns in a table using ALTER table command with add clause. Added column becomes last column by default.</p> <p>Syntax to add a column:-</p> <pre>ALTER TABLE <table_name> ADD (<column_name1>< datatype>(size), . . <column_nameN>< datatype>(size));</pre> <p>2. Drop Column: - We can delete the existing column with help of drop clause in the ALTER table command. We can drop one column at a time. After dropping any column from the table, there must be at least one column left in the table.</p> <p>Syntax to drop a column:-</p> <pre>ALTER TABLE <table_name > DROP column <column_name>;</pre> <p>3. Modify column:-We can change the data type and/or size of a column in a table by using modify clause in ALTER table. The size of the column can be increased or decreased if the column contains only null values or if the table has no rows.</p> <p>Syntax to modify a column</p> <pre>ALTER TABLE <table_name > MODIFY(<column_name1>< datatype>(size), <column_name2>< datatype>(size), . . <column_name N>< datatype>(size)) ;</pre>	<p><i>(Explanation :2 marks, any two options:2 marks)</i></p>
d)	<p>Describe ACID properties of transaction.</p>	<p>4M</p>



	Ans:	ACID Properties: 1. Atomicity: When one transaction takes place, many operations occur under one transaction. Atomicity means either all operations will take place properly and reflect in the database or none of them will be reflected. 2. Consistency: Consistency keeps the database consistent. Execution of a transaction needs to take place in isolation. It helps in reducing complications of executing multiple transactions at a time and preserves the consistency of the database. 3. Isolation: It is necessary to maintain isolation for the transactions. This means one transaction should not be aware of another transaction getting executed. Also their intermediate result should be kept hidden. 4. Durability: When a transaction gets completed successfully, it is important that the changes made by the transaction should be preserved in database in spite of system failures.	<i>(Four properties: 1 mark each)</i>
	e)	Explain the following terms with syntax and example. i) Creating snapshot ii) Altering snapshot iii) Dropping a snapshot.	4M
	Ans:	i) Creating Snapshot: Create snapshot command is used to create the snapshot. Syntax:- CREATE SNAPSHOT [schema.] <snapshot name> AS subquery; Example:- Create snapshot emp_snap as select * from emp where deptno=6; ii) Altering snapshot Snapshot can be altered by using ALTER SNAPSHOT command. The only parts of a snapshot that can be altered are its storage parameters, refresh type and refresh start, and next interval. The select for the snapshot, base tables, and other data related items cannot be changed without dropping and recreating the snapshot. Syntax:- ALTER SNAPSHOT <snapshotname> [[PCTFREE <integer>] [PCTUSED <integer>] [REFRESH [FAST/COMPLETE/FORCE]]]; Example:- To change the automatic refresh mode for the emp_data snapshot to fast: ALTERSNAPSHOT emp_data REFRESH FAST;	<i>(Explanation with syntax and example: 4 marks)</i>



iii) Dropping a snapshot

To remove the snapshot DROP SNAPSHOT Command .When snapshot is dropped which a snapshot log had associated with it, only the rows required for maintaining that snapshot are dropped.

Syntax:-

Drop snapshot <snapshot name>;

Example:-

Drop snapshot emp_snap;

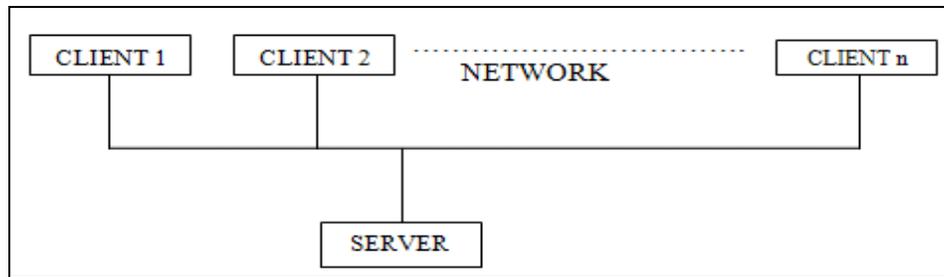
f) Explain client-server architecture.

4M

Ans:

Client server is a system in which all the data & information is stored on the server & all application programs are stored on the client. In this client server relationship several clients can share the same server as shown in the following: -

*(Explanation: 4 marks
Diagram
Optional)*



Computer networking allows some task to be executed on a server system and some tasks on client system. This leads to development of client server architecture. The clients are the machines which requests for the service to the server. Server is the machine which serves to the clients.

There are different types of client/server architecture such as

- Two tier architecture,
- Three tier architecture.

In two tier architecture, client systems directly approach database servers whereas in three tier architecture, there exists a middle layer which acts as application server to receive and send requests from client machine to database server and vice versa.

5. Attempt any four of the following:

16

a) Explain data warehousing and data mining.

4M

Ans: Data Warehousing:-

- A data warehouse is a repository of information gathered from multiple sources, stored under a unified schema, at a single site. Once gathered, data are stored for long time, permitting access to historical data.

(Data Warehousing :2 marks,



- Data warehouses provide the user a single consolidated interface to data, making decision-support queries easier to write.
- Moreover, by accessing information for decision support from a data warehouse, the decision makers ensures that online transaction-processing systems are not affected by decision support workload

Data Mining: Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful and ultimately understandable patterns in data.

- It is known as “Knowledge Discovery in Databases”. When the data is stored in large quantities in data warehouse, it is necessary to dig the data from the warehouse that is useful and required for further use.
- For data mining, different software tools are used to analyze, filter and transfer the data from the data warehouses.

Feature of data mining:

- prediction
- identification
- classification
- optimization

Data Mining:2 marks)

b) Explain any four string functions with example.

4M

Ans:

Function Name	Return Value	Examples	Return Value
LOWER (string_value)	All the letters in 'string_value' is converted to lowercase.	Select LOWER('Good Morning') from dual;	good morning
UPPER (string_value)	All the letters in 'string_value' is converted to uppercase.	Select UPPER('Good Morning')from dual;	GOOD MORNING
INITCAP (string_value)	All the letters in 'string_value' is converted to mixed case.	Select INITCAP('GOOD MORNING') from dual;	Good Morning
LTRIM (string_value, trim_text)	All occurrences of 'trim_text' is removed from the left of 'string_value'.	Select LTRIM ('Good Morning', 'Good)	Morning
RTRIM (string_value, trim_text)	All occurrences of 'trim_text' is removed from the right of 'string_value'.	Select RTRIM ('Good Morning', 'Morning') from dual;	Good

(Explanation :1 mark, Example: 1 mark)



		TRIM (trim_text FROM string_value)	All occurrences of 'trim_text' from the left and right of 'string_value', 'trim_text' can also be only one character long.	Select TRIM ('o' FROM 'Good Morning') from dual;	Gd Mrning
		SUBSTR (string_value, m, n)	Returns 'n' number of characters from 'string_value' starting from the 'm' position.	Select SUBSTR('Good Morning',2,3) from dual;	ood
		LENGTH (string_value)	Number of characters in 'string_value' in returned.	Select Length('Mumbai') from dual;	6
		LPAD (string_value, n, pad_value)	Returns 'string_value' left-padded with 'pad_value'. The length of the whole string will be of 'n' characters.	Select LPAD ('Good', 6, '*') from dual;	**Good
		RPAD (string_value, n, pad_value)	Returns 'string_value' right-padded with 'pad_value'. The length of the whole string will be of 'n' characters.	Select RPAD ('Good', 6, '*') from dual;	Good**

	c)	Explain function in PL/SQL with suitable example	4M
--	----	---	-----------

Ans:	<p>A function (also called a user function or user-defined function) is a set of PL/SQL statements that is called by name. A function returns a value to the environment in which it is called.</p> <p>Syntax:</p> <pre>CREATE [OR REPLACE] FUNCTION function_name [parameters] RETURN return_datatype; IS Declaration_section BEGIN Execution_section Return return_variable; EXCEPTION exception_section Return return_variable; END;</pre> <p>Example :</p>	<p><i>(Explanation :2 marks, Example: 2 marks)</i></p>
-------------	--	--



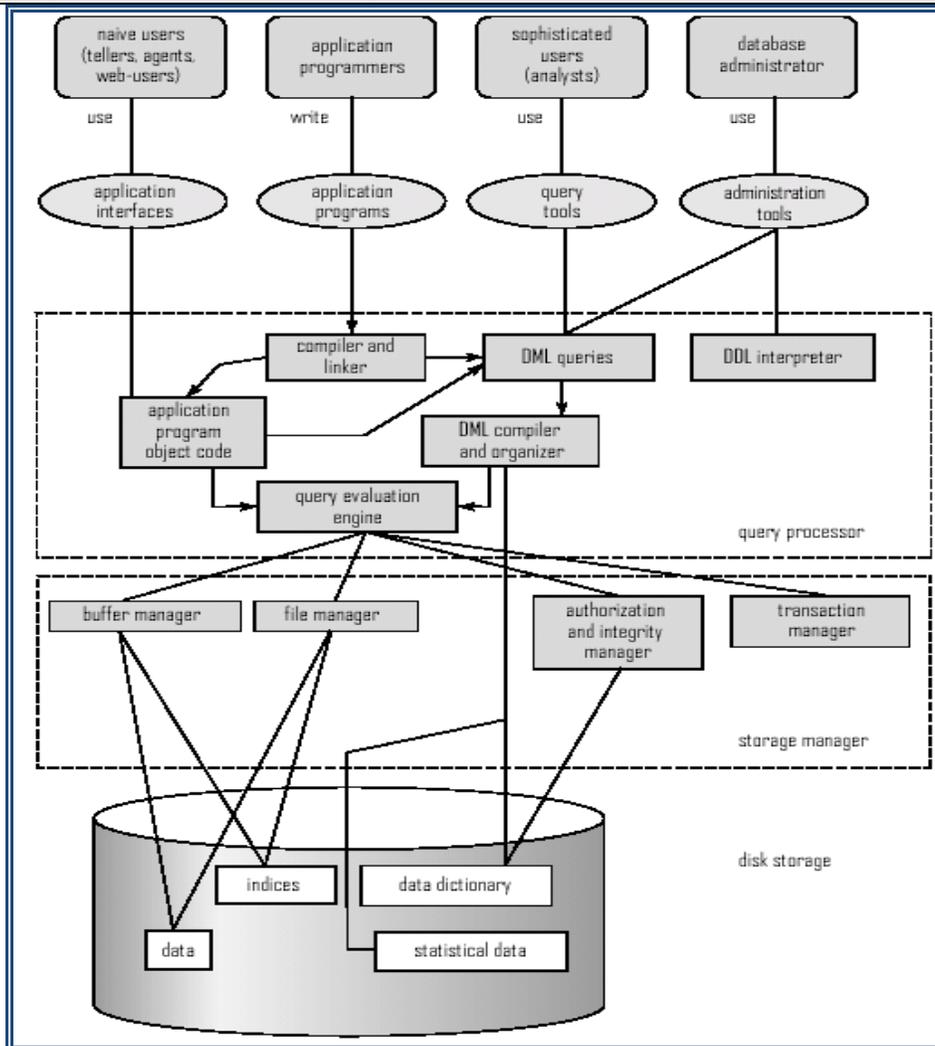
	<pre>CREATE OR REPLACE FUNCTION employer_details_func RETURN VARCHAR(20); IS ename VARCHAR(20); BEGIN SELECT fname INTO ename FROM emp WHERE empID = '100'; RETURN ename; END;</pre>	
d)	Explain two locking strategies.	4M
Ans:	<p>Implicit Locking: Oracle engine automatically locks table data while executing SQL statements</p> <p>Explicit Locking: The technique of lock taken on a table or its resources by a user is called Explicit Locking. Users can lock tables they own or any tables on which they have been granted table privileges (such as select, insert, update, delete). Explicit locking done by two ways as</p> <p>1) The Select.....For Update statement It is used for acquiring exclusive row level locks in anticipation of performing updates on records.</p> <p>2) Using lock table statement: To manually override Oracle's default locking strategy by creating a data lock in a specific mode.</p> <p>Syntax:</p> <pre>LOCK TABLE <TableName> [, <TableName>]... IN { ROW SHARE ROW EXCLUSIVE SHARE UPDATE SHARE SHARE ROW EXCLUSIVE EXCLUSIVE} [NOWAIT]</pre>	<i>(Explanation of each strategy: 2 marks)</i>
e)	Explain BCNF with example. [Any Example should be considered]	4M
Ans:	<p>BCNF: Definition: A relation R is in Boyce-Codd normal form (BCNF) if and only if every determinant is a candidate key.</p> <p style="text-align: center;">OR</p> <p>BCNF: A relation R is in BCNF if whenever on nontrivial functional dependency $X \rightarrow A$ holds in R, then X is a superkey of R.</p>	<i>(Explanation :2 marks, Example: 2 marks)</i>



	Example: Consider the relation SUPPLIER (SNO, SNAME, PH_NO, CITY) having SNO and SNAME unique. In this there are 2 determinants SNO, SNAME as PH_NO and CITY dependence upon them and both are candidate keys. So this is in BCNF.	
f)	Consider the following database schema: Emp(Empno,Ename,job,mgr,joindate,salary,comm.,deptno). Write the SQL queries for the following: i) Write a query to find list of employees whose salary is not less 5000. ii) Write a query to find list of employees whose job title is either “Manager” or “Analyst”. iii) Change the location of deptno 40 to Pune from Chandrapur. iv) Display the Ename and salary of employees who earn more than Rs. 50,000 and are in deptno 10 or 30.	4M
Ans:	i) select * from Emp where salary >= 5000; ii) select * from Emp where job='Manager' or job='Analyst'; OR ii) select * from Emp where job in('Manager' , 'Analyst'); iii) update Emp set location='Pune' where deptno=40; {consider location attribute in Emp table} iv) select Ename,Salary from Emp where salary >50000 and(deptno = 10 or deptno=30);	(Each Correct query: 1 mark)
6.	Attempt any four of the following:	16
a)	Draw overall structure of DBMS.	4M



Ans:



(Correct Diagram : 4 marks)

b) Explain Domain integrity constraint with syntax and example.

4M

(Any correct syntax should be considered)

Ans:

1. Not Null constraint: This constraint ensures all rows in the table contain a definite value for the column which is specified as not null. Which means a null value is not allowed.

syntax: create table <table name>(Column_name Datatype (Size) [CONSTRAINT constraint name] NOT NULL);

Example: To create a employee table with Null value, the query would be like

```
CREATE TABLE employee
( id number(5),
  name char(20) CONSTRAINT nm_nn NOT NULL,
  dept char(10),
```

(each constraint :Description 1 mark, Example 1 mark] (any other example can be considered)



```
age number(2),
salary number(10),
location char(10)
);
```

OR

For Example: To create a employee table with Null value, the query would be like

```
CREATE TABLE employee
( id number(5),
name char(20) NOT NULL,
dept char(10),
age number(2),
salary number(10),
location char(10)
);
```

1) **Check constraint:** it defines a condition that each row must satisfy. A single column can have multiple check constraints that reference the column in its definition.

Syntax at table creation:

```
Create table <table_name>
(column_name1 datatype(size) constraint <constraint_name> check <condition or
logical expression>,
---
column_name n datatype(size)
);
```

Example:

```
create table emp( empno number(5), ename varchar2(25), salary number(7,2)
constraint emp_sal_ck check(salary > 5000), job varchar2(15) );
```

After table creation

Syntax:

```
Alter table <table_name> add constraint<constraint_name> check <condition>;
```

Example:

```
alter table emp add constraint emp_deptno_ck check(deptno>5);
```

c) **Give the use of grant and revoke command with syntax and example.**

(Any correct syntax should be considered)

4M

Ans: **1. GRANT** command is used to provide access or privileges on the database objects to the users.

The Syntax for the GRANT command is:

*(Explanation:
2 marks,
Syntax: 1
mark,*



GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
[WITH GRANT OPTION];

The Example for the GRANT command is:

GRANT insert,update
ON emp
TO staff ;

2. REVOKE command removes user access rights or privileges to the database objects.

The Syntax for the REVOKE command is:

REVOKE privilege_name
ON object_name
FROM {user_name |PUBLIC |role_name};

The Example for the REVOKE command is:

REVOKE insert,update
ON emp
FROM staff ;

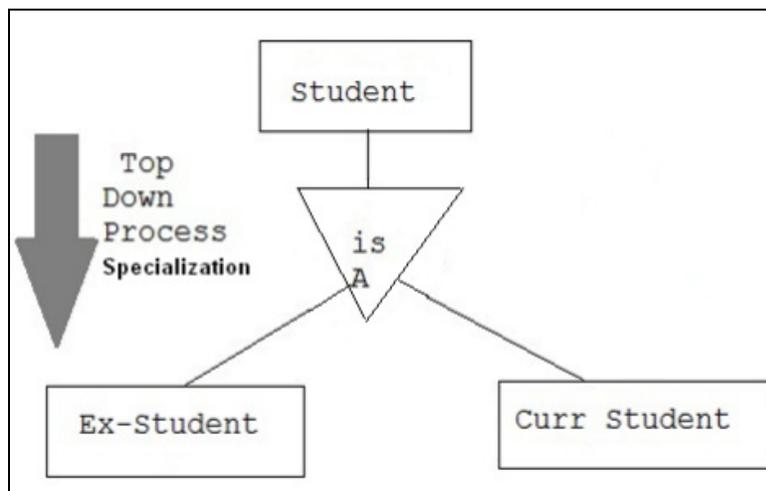
*Example :1
mark)*

d) Explain the term specialization and generalization with suitable example.

4M

Ans:

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, some higher level entities may not have lower-level entity sets at all.

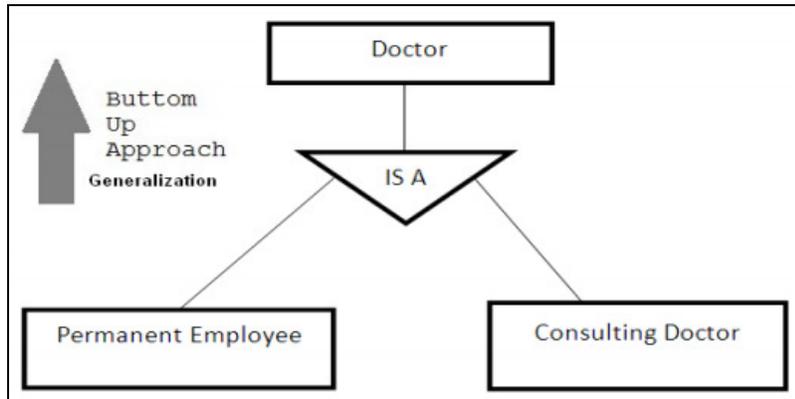


Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine

*(For Each
Explanation:
1 mark,
example :1
mark)*



with other lower level entity to make further higher level entity.



e) Explain Inner join and Outer join with example.
(Any one outer join should be considered)

4M

Ans: **INNER Join:** This is a simple JOIN in which the result is based on matched data as per the condition specified in the query.

Inner Join Syntax :

```

SELECT column_name_list
from table_name1
INNER JOIN
table_name2
on table_name1.column_name = table_name2.column_name;
  
```

Inner Join Example :

```

SELECT * from emp inner join dept on emp.id = dept.id;
  
```

Outer Join is based on both matched and unmatched data. Outer Joins subdivide further into,

- Left Outer Join
- Right Outer Join
- Full Outer Join

Left Outer Join

The left outer join returns a result table with the **matched data** of two tables then remaining rows of the **left** table and null for the **right** table's column.

Left Outer Join syntax :

```

SELECT column-name-list from table-name LEFT OUTER JOIN table-name2
on table-name1.column-name = table-name2.column-name;
  
```

Left Outer Join Example:

(Inner join :Explanation 1 mark, Example 1 mark, Outer join : Explanation 1 mark, Example 1 mark)



SELECT * FROM emp LEFT OUTER JOIN dept ON (emp.id=dept.id);

Right Outer Join

The right outer join returns a result table with the **matched data** of two tables then remaining rows of the **right table** and null for the **left** table's columns.

Right Outer Join Syntax:

select column-name-list from *table-name1* **RIGHT OUTER JOIN** *table-name2*
on table-name1.column-name = table-name2.column-name;

Right Outer Join Example:

SELECT * FROM emp **RIGHT OUTER JOIN** dept on (emp.id=dept.id)

Full Outer Join

The full outer join returns a result table with the **matched data** of two table then remaining rows of both **left** table and then the **right** table.

Full Outer Join Syntax :

select column-name-list from *table-name1* **FULL OUTER JOIN** *table-name2*
on table-name1.column-name = table-name2.column-name;

Full Outer Join Example:

select empname,sal from *emp* **FULL OUTER JOIN** *dept* on emp.id = dept.id;

f) Explain loop control structure used in PL/SQL.

4M

Ans:

LOOP: PL/SQL LOOP statement is an iterative control statement that allows to execute a sequence of statements repeatedly like WHILE and FOR loop.

Syntax:

LOOP
sequence_of_statements;
END LOOP;

Example:

i number :=1;
Loop
dbms_out.put_line('Good Morning');
i :=i+1;
Exit when i=10
End Loop

WHILE: If it is not known in advance how many times a sequence of statements needs to execute. In such cases, one should use PL/SQL WHILE LOOP statement.

Syntax:

(For any one structure description : 2 marks , syntax or example 2 marks)



WHILE condition
LOOP
sequence_of_statements;
END LOOP;

Example:

```
i number := 1;  
while (i<=10) Loop  
dbms_output.put_line(i);  
i :=i+1;  
End Loop;
```

3) FOR: FOR loop is an iterative statement that execute a sequence of statements a fixed number of times.

Syntax:

```
FOR loop_counter IN [REVERSE] lower_bound .. higher_bound  
LOOP  
sequence_of_statements;  
END LOOP;
```

Example:

```
For i in 1..10 loop  
dbms_output.put_line(i);  
end loop;
```