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 judgment 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.

(Note: Consider any relevant example and explanation)

1. i) Attempt any six of the following: Marks 12

a) What is data redundancy?

(Definition of data redundancy-2 Marks)

Ans: Data redundancy means storing same data multiple times at different locations.

b) Describe data independence with its type.

(Definition of Data independence- 1Mark, Types of data independence – 1/2 Mark each)

Ans: Data independence:

The ability to modify a schema definition in one level without affecting a schema definition in next higher level is called data independence.

There are two types of data independence.

1. **Physical data independence**

   Physical data independence is the ability to change internal level without having change in conceptual or external level.

2. **Logical data independence**

   Logical data independence is the ability to change conceptual level without having change in external level or application program.

c) State weak and strong entity set.

(Weak entity set- 1 Mark, Strong entity set- 1 Mark)

Ans: Weak entity set:

Entity set that have no key attributes are called as weak entity set. Weak entity set are dependent on some other strong entities.

Strong entity set:

The entity set that have some key attribute for identifying records are called strong entity set.

d) List any four DDL commands.

(Consider any four DDL commands - 1/2 Mark each)

Ans: DDL Commands:

1. Create
2. Alter
3. Drop
4. Rename
5. Desc
6. Truncate

e) Describe candidate key.

(Explanation of Candidate Key – 2 Marks)

Ans: Candidate key:

Sometimes in relation, there are more than one attributes are having the unique identification property. Those attributes are known as candidate key. E.g. Consider Student(RNO, NAME, PER, BRANCH), if RNO and NAME are unique then both are known as Candidate keys.

f) Explain shared lock and explicit lock.

(Explanation of Shared lock – 1 Mark, explicit lock – 1 Mark)

Ans: Shared Lock:

Shared locks are placed on resources whenever a read operation (SELECT) is performed.
Multiple shared locks can be simultaneously set on a resource.

Explicit Lock:

The technique of lock taken on a table or its resources by a user is called explicit locking.

g) What is select operation in relational algebra? Give one Example.

(Use of select operation – 1 Mark, Example – 1 Mark)

Ans: **Select (σ)**
Select is used to obtain a subset of the tuples of a relation that satisfy a select condition.

OR

The select operation selects tuples that satisfy a given predicate.

Example:

To find all employees having salary more than 10000.

$\sigma_{\text{Salary>10000}}^{\text{(employee)}}$

h) Write the syntax for creating a view.

(Syntax- 2 Marks)

Ans: Syntax for creating view.

Create view <viewname> as select <query>

ii) Attempt any two of the following: **Marks 08**

a) Explain any four functions of DBMS.

(Consider any four function – 1 Mark each (Any other functions of DBMS is considered))

Ans: **Database Communication Interfaces:** The end-user's requests for database access are transmitted to DBMS in the form of communication messages.

Authorization / Security Management: The DBMS protects the database against unauthorized access, either international or accidental. It furnishes mechanism to ensure that only authorized users an access the database.

Backup and Recovery Management: The DBMS provides mechanisms for backing up data periodically and recovering from different types of failures. This prevents the loss of data,

Concurrency Control Service: Since DBMSs support sharing of data among multiple users, they must provide a mechanism for managing concurrent access to the database. DBMSs ensure that the database kept in consistent state and that integrity of the data is preserved.

Transaction Management: A transaction is a series of database operations, carried out by a single user or application program, which accesses or changes the contents of the database. Therefore, a DBMS must provide a mechanism to ensure either that all the updates corresponding to a given transaction are made or that none of them is made.

Database Access and Application Programming Interfaces: All DBMS provide interface to enable applications to use DBMS services. They provide data access via Structured Query Language (SQL). The DBMS query language contains two components: (a) a Data Definition Language (DDL) and (b) a Data Manipulation Language (DML).

Data integrity and consistency: To provide data integrity and data consistency, the dbms uses sophisticated algorithms to ensure that multiple user can access the database concurrently without compromising the integrity of the database.

b) Explain tuple relational calculus with example.

(Description – 2 Marks, example - 2 Marks)

Ans: Tuple relation 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|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 ‘á’ & we use ‘t ε r’ to denote that ‘t’ is in relation ‘r’. There are different symbols with specific meaning which can be used to write tuple calculus expression:-

1. ε - belong to
2. ┐ - not
3. ^ - and
4. V – or
5. => implies

Example: Find records of employees where salary is more than 20000.

{t|s ε emp(s[salary]>20000)}

c) Explain any four string functions with example.

(Any four function-1 Mark each (Any other String function is considered)

Ans:

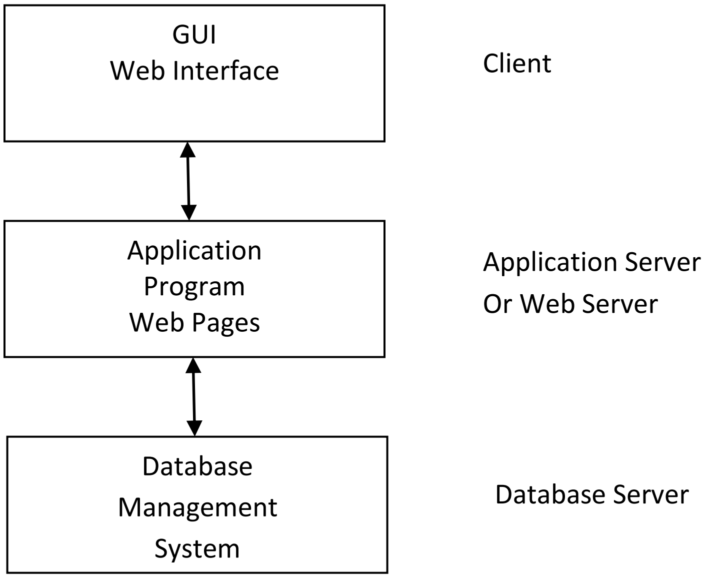
Sr. No	Function	Description
1.	Initcap(str)	Converts first letter of string to capital letter. Example: Select initcap(‘rdbms’) from dual;
2.	Lower(char)	Converts a string to all lowercase characters. Example: Select lower(‘RDBMS’) from dual;
3.	Upper(char)	Converts a string to all uppercase characters. Example: Select upper(‘rdbms’) from dual;
4.	Length(char)	Find outs the length of given string. Example: Select length(‘RDBMS’) from dual;
5.	Ltrim(char,set)	It trim from the left of character string. Example: Select Ltrim(‘manas khan’, ‘manas’) from dual;
6.	Rtrim(char,set)	It trim from the Right of character string. Example: Select Rtrim(‘manas khan’, ‘khan’) from dual;
7.	Lpad(char1,length, char2)	It returns char1, left-padded to given length with the sequence of characters in char2. Example: Select Lpad(‘SKY’, 8, ‘=’) from dual;
8.	Rpad(char1,length ,char2)	It returns char1, right-padded to given length with the sequence of characters in char2. Example: Select Lpad(‘SKY’, 8, ‘=’) from dual;

2. Attempt any four of the following: Marks 16

a) Explain three tier architecture with diagram.

(Explanation – 2 Marks, Diagram – 2 Marks)

Ans: In three tier architecture the communication taken place from client to application server and then application server to database system to access the data. The application server or web server is sometimes called middle layer or intermediate layer. The middle layer which processes applications and database server processes the queries. This type of communication system is used in the large applications or the world web applications. On WWW all clients requests for data and server serves it. There are multiple servers used like fax server, proxy server, mail server etc.



b) Explain normalization with example.

(Explanation – 2 Marks, Example – 2 Marks)

Ans: Normalization:

In relational database design, the process of organizing data to minimize redundancy is known as redundancy. There are various level s under normalization form as: 1NF, 2NF, 3NF ,BCNF .

1. **First normal form:** A table is in the first normal form if it contains no repeating columns.
2. **Second normal form:** A table is in the second normal form if it is in the first normal form and contains only columns that are dependent on the whole (primary) key.

3. **Third normal form:** A table is in the third normal form if it is in the second normal form and all the non-key columns are dependent only on the primary key. If the value of a non-key column is dependent on the value of another non-key column we have a situation known as transitive dependency. This can be resolved by removing the columns dependent on non-key items to another table.

Example:

A college maintains details of its lecturers' subject area skills. These details comprise:

- Lecturer Number
- Lecturer Name
- Lecturer Grade
- Department Code
- Department Name
- Subject Code
- Subject Name
- Subject Level
- Assume that each lecturer may teach many subjects but may not belong to more than one department.
- Subject Code, Subject Name and Subject Level are repeating fields.

Normalize this data to Third Normal Form.

UNF [unnormalized form]

1. Lecturer Number (PK), Lecturer Name, Lecturer Grade, Department Code, Department Name, Subject Code, Subject Name, Subject Level

1NF

1. Lecturer Number (PK), Lecturer Name, Lecturer Grade, Department Code, Department Name
2. Lecturer Number (FK), Subject Code (PK), Subject Name, Subject Level

2NF

1. Lecturer Number (PK), Lecturer Name, Lecturer Grade, Department Code, Department Name
2. Lecturer Number (FK), Subject Code (PK)
3. Subject Code (PK), Subject Name, Subject Level

3NF

1. Lecturer Number (PK), Lecturer Name, Lecturer Grade
2. Department Code (PK), Department Name
3. Lecturer Number (FK), Subject Code (PK)
4. Subject Code (PK), Subject Name, Subject Level

c) Describe savepoint and rollback commands with example.

(Explanation of Commands -2 Marks, Example – 2 Marks)

Ans: Savepoint:

Savepoints define breakpoints for the transaction to unable partial rollback. Savepoints are treated as marker to divide lengthy transaction to smaller one.

Syntax: savepoint <savepoint_name>;

Rollback:

The rollback command can be used to undo the changes done in current transaction. We can either rollback the entire transaction or till a particular savepoint transaction can be rolled back. If savepoint is given then rollback will be done till the savepoint.

Syntax: rollback; or rollback [to < savepoint>];

Example:

```
Update emp  set empid=124;
Savepoint ss1;
Delete from emp where job='çlerk';
Savepoint ss2;
Rollback to ss1;
```

d) What is data abstraction, instances and schema?

(Data Abstraction – 2 Marks, Instance – 1 Mark, Schema – 1 Mark)

Ans: 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. There are three levels of data abstraction:

1. Physical level
2. Conceptual / logical level

3. View level

Schema:

The overall design of the database is known as schema. The database schemas are partitioned at different level of abstractions.

Instance:

The collection of information stored in the databases at a particular moment is called as an instance.

e) Write a program to find largest of two numbers.

(Correct logic – 2 Marks, Syntax – 2 Marks)

Ans:

```
Declare
A number(3);
B number(3);
Begin
A:=&A;
B:=&B;
If (A>B) then
Dbms_output.put_line(’ Greater number=’||A);
else
Dbms_output.put_line(’ Greater number=’||B);
End if;
End;
```

f) Explain while loop with example.

(Explanation- 2 Marks, Example-2 Marks, consider any other valid example)

Ans: **While loop:**

The while loop is used to iterate the statements till a particular condition is true.

Syntax:

```
While<condition>
Loop
<statements>
End loop;
```

Example:

```
Declare
Num number(3):=5;
Fact number(3):=1;
Begin
While(Num!=0)
Loop
Fact:=Fact*Num;
Num:=Num-1;
End loop;
Dbms_output.put_line('Factorial of 5:'||Fact);
End;
```

3. Attempt any four of the following: Marks 16

a) Explain domain integrity constraint with example.

(Explanation -2 Marks, Example -1 Mark each)

Ans: It is a type of integrity constraint.

Domain integrity constraint is used to maintain the value according to user specification.

It has two types:

a) Not null constraint:

When not null constraint is assigned on a column or set of column, then it will not allow null values.

Syntax:

Create table <tablename>(column1 datatype(size),column2 datatype(size) not null);

Example:

Create table Emp (EmpId number(4),Empname varchar(20) not null);

b) Check constraint:

It defines condition that each row must satisfy. Single column can have multiple check conditions.

Syntax:

Create table <tablename>(column1 datatype(size),column2 datatype(size) check (logical expression));

Example:

Create table Emp (EmpId number(4), Empname varchar(20),sal number(20) check (sal>4000));

b) Compare between network and hierarchical model.

(Any 4 points -1 Mark each)

Ans:

Sr. No.	Network model	Hierarchical model
1.	Network model is more popular than the hierarchical and relational model.	Hierarchical model is not more popular than network model
2.	It uses client –server architecture	It does not uses client server architecture
3.	One to many and many to many relationship is maintained.	One to many relationship is maintained.
4.	Network model is based on tree like structure with many root.	Hierarchical model is based on tree like structure with one root.
5.	Many children have many parent	One child or many children have only one parent
6.	It upgraded version of hierarchical model so used in network	Main application of hierarchical model is in the mainframe database system.

c) List and explain types of DBMS users.

(List - 2 Marks, Any two types -1 Mark each)

Ans: List of DBMS user.

a) Naive users

b) Application programmers

c) Sophisticated users

d) Specialized users

Explanation:

a) Naive User:

- Naïve users are unsophisticated users.
- They are 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 program.
- Example: Bank cashier.

b) Application programmers:

- Application programmers are the users who write the program.
- These programmers use programming tools to develop the program.
- RAD technology is used to write the program.

c) Sophisticated users:

- 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.
- Some sophisticated users can be analyst.

d) Specialized users:

- These users are not traditional.
- They write some special application programs which are not regular applications.
- Example: such types of applications are CAD, knowledge based and expert system.

d) What are sequences? Write syntax for creating sequence.

(Explanation -2 Marks, Syntax-2 Marks)

Ans: Sequence:

- It is database object that generate/produce integer values in sequential order.
- It automatically generates primary key and unique key values.
- It may be ascending or descending order.

- It can be used for multiple tables.
- Sequence numbers are stored and generated independently of tables.

Syntax:

```
Create sequence <seq_name>
Start with [initial value]
Increment by [value]
Minvalue [minimum value]
Maxvalue [maximum value]
[cycle/nocycle]
[{cache value / No cache}]
[{order / No order}];
```

e) Explain block structure of PL_SQL.

(Syntax-2 Marks, Explanation- 2 Marks)

Ans: Declare

Declaration of memory variables

BEGIN (Mandatory)

SQL executable statements

Exception

Handling errors

END; (Mandatory)

A block begins with a declarative section where variables are declared. This is followed by a section containing the procedural statements surrounded by BEGIN and END keywords. Each block must have Begin and End statements and may optionally include an exception section to handle errors.

f) Write syntax of insert command. Demonstrate with suitable example.

(Any one Syntax -2 Marks, Any one example-2 Marks)

Ans: It is DML statement/command.

Used to insert or add data into table.

Syntax1: syntax for insert command without defining attributes

Insert into <Table_name> values (value1,value2,…..);

Syntax2: syntax for insert command with defining attributes

Insert into <Table_name>(col1,col2,…) values (value1,value2,…..);

Syntax3: To insert values for selected attributes in a table.

Insert into <Table_name>(col1,col2) values (value1,value2);

Syntax4: To insert more than one row.

Insert into <Table_name> values(&attribute1, '&attribute2',…..&attribute n);

Example:

Consider table "**persons**"

Last_Name	First_name	Address	City
Varma	Ram	Airoli	Navi mumbai

Example for both syntax and its output

Example1:

Insert into persons values ('Bhosale','Ajit','Bapat Road','pune');

Example2:

Insert into persons (Last_Name, First_name, Address, City) values ('Suryawanshi','Savita','Meera road','Raigad');

Example3:

Insert into persons (Last_Name, First_name) values ('Suryawanshi','Savita');

Example4:

Insert into persons values ('&Last_Name', '&First_name', '&Address', '&City')

Output:

Last_Name	First_name	Address	City
Varma	Ram	Airoli	Navi mumbai
Bhosale	Ajit	Bapat Road	pune
Suryawanshi	Savita	Meera road	Raigad

4. Attempt any four of the following: ` Marks 16

a) Explain three levels of data abstraction with suitable diagram.

(Diagram- 1 Mark, Each level-1 Mark)

Ans: Three levels are as follows:

1) Physical level

2) Logical level

3) View level

Diagram:

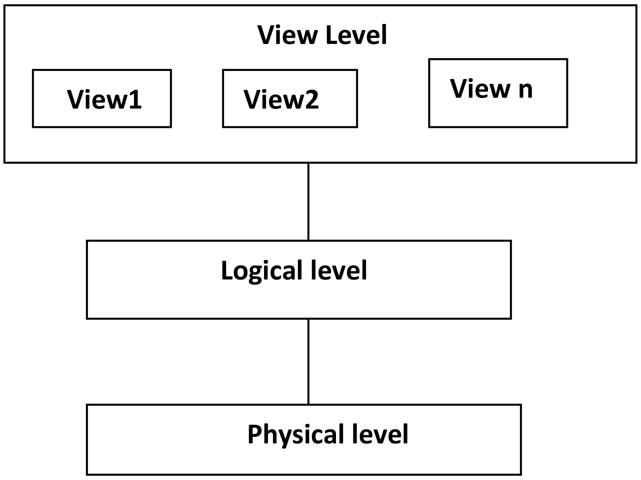


Fig .Three levels of data abstraction

Explanation:

1) Physical Level:

- 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.

2) Logical Level:

- The level next to physical level is called logical level.
- This level defines what data stored in the database and what are the relationships among these data.

- Fully decides the structure of the entire database.

3) View Level:

- This level is used to show the part of database to user.
- There are 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.

b) Identify the laws for following relation:

i) $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$

ii) $R \bowtie S = S \bowtie R$

iii) $R{}^\wedge\phi=\phi$

iv) $R \cup \phi = R$

(For each laws -1 Mark)

Ans: i) $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$: Associative law

ii) $R \bowtie S = S \bowtie R$: Commutative Law

iii) $R \cap \phi = \phi$: Domination Law

iv) $R \cup \phi = R$: Identity Law

c) List and explain set of operators in SQL with example.

(Each operator 1 Mark)

{Note: Any other operators like Arithmetic, Comparison, Logical operators should be consider}

Ans: 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

emp1
ename
abc
xyz
lmn

emp2
ename
pqr
xyz

1) **Union** : The Union of 2 or more sets contains all elements, which are present in either or both. Union works as or.

Eg select ename from emp1 union select ename from emp2;

The output considering above data is :

```
Ename
abc
xyz
lmn
pqr
```

2) **Union all** : The Union of 2 or more sets contains all elements, which are present in both, including duplicates.

Eg select ename from emp1 union all select ename from emp2;

The output considering above data is :

```
Ename
abc
xyz
lmn
pqr
xyz
```

3) **Intersection:** The intersection of 2 sets includes elements which are present in both.

Eg select ename from emp1 intersection select ename from emp2;

The output considering above data is:

```
Ename
 Xyz
```

4) **Minus:** The minus of 2 sets includes elements from set1 minus elements of set2.

Eg select ename from emp1 minus select ename from emp2;

The output considering above data is:

```
Ename
 abc
 lmn
```

d) Explain any four aggregate functions with their examples.

(Any four function-1 Mark each, Meaning with syntax - ½ Mark each , Example - ½ Mark each)

Ans: Aggregate Functions: Avg, sum, min, max, count, variance and stddev.

1) Avg(column_name):

It is used to calculate average of values in column

or

Returns average value of column.

Example:

```
Select avg(salary) from Emp;
```

2) Sum(column):

used to calculate sum/addition of values of column.

Or

Returns addition of column of number data type.

Example:

```
select sum(salary) from Emp;
```

3) Min(column):

Used to find lowest value in column

Or

Return lowest value from column.

Example:

```
select min(salary) from Emp;
```

4) Max(column):

Used to find highest value in column.

or

Returns heighest value in column.

Example:

```
select max(salary) from Emp;
```

5) Count(column_name):

count the number of values in a column.

Example:

select count(name) from Emp;

6) Count(*):

count the total number of rows.

Example:

select count(*) from Emp;

7) Count(Distinct column_name):

count the total number of values without duplication.

Example: select count(distinct name) from Emp;

e) What is snapshot? Write the syntax for creating a snapshot.

(Explanation- 2 Marks, Syntax- 2 Marks)

Ans: **Snapshot:**

- 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,
- Views or either rows using join, grouping and selection criteria.
- Useful in distributed environment

It has two types:

Simple snapshot and complex snapshot. Simple snapshot related to single table and complex snapshot related to joined tables.

Syntax:

```
Create snapshot snapshot_name
Pctfree n
Pctused n
Tablespace tablespace_name
Storage clause
Refresh
Star with date
As select statement;
```

OR

```
Create snapshot snapshot_name As select statement;
```

f) What are exceptions? Explain its types.

(Any relevant Explanation-2 Marks (Consider either Syntax or Example), Types-1 Mark each (2 types))

Ans: Exception is nothing but an error.

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 exception condition.

Syntax:

```
DECLARE
Declaration section
…executable statement;
EXCEPTION
WHEN ex_name1 THEN
-Error handling statements/user defined action to be carried out;
END;
```

For Example: Suppose a NO_DATA_FOUND exception is raised in a proc, we can write a code to handle the exception as given below.

```
BEGIN
Execution section
EXCEPTION
WHEN NO_DATA_FOUND THEN
dbms_output.put_line ('A SELECT...INTO did not return any row.');
END;
```

Types:

1) **Predefined Exception/system defined exception/named exception:**

Are always automatically raised whenever related error occurs. The most common errors that can occurs 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(errorno,errorname);

5. Attempt any four of the following: Marks 16

a) Explain ACID properties.

(Each property – 1 Mark)

Ans: ACID Properties:

• **Atomicity**. Either all operations of the transaction are reflected properly in the database, or none are.

• **Consistency**. Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database.

• **Isolation**. Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions *Ti* and *Tj* , it appears to *Ti* that either *Tj* finished execution before *Ti* started, or *Tj* started execution after *Ti* finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.

• **Durability**. After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

b) What is cursor? Explain types of cursors.

(Cursor definition – 2 Marks, Each type – 1 Mark)

Ans: Oracle creates a memory area, known as context area, for processing an SQL statement, which contains all information needed for processing the statement, for example, number of rows processed, etc.

A cursor is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the **active set**.

You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors:

- Implicit cursors
- Explicit cursors
- **Implicit Cursor :**

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement.

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.

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

Attribute	**Description**
%FOUND	Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE.
%NOTFOUND	The logical opposite of %FOUND. It returns TRUE if an INSERT, UPDATE, or DELETE statement affected no rows, or a SELECT INTO statement returned no rows. Otherwise, it returns FALSE.
%ISOPEN	Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement.
%ROWCOUNT	Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SELECT INTO statement.

- **Explicit cursor :**

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.

The syntax for creating an explicit cursor is :

Cursor cursor_name IS select_statement;

Working with an explicit cursor involves four steps:

- Declaring the cursor for initializing in the memory
- Opening the cursor for allocating memory
- Fetching the cursor for retrieving data
- Closing the cursor to release allocated memory

c) Explain 'Group by' in SQL with suitable example.

(Explanation of Group by- 2 Marks, Example – 2 Marks)

Ans: The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups.

The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

SELECT dept, SUM(salary) from emp group by dept;

dept	SUM(salary)
10	25000
20	30000

d) Describe following terms:

i) unique indexes

ii) composite indexes.

(Each index – 2 Marks)

Ans: **i) 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:

Create unique index index_name on table_name(column_name);

The example is as follows:

Create unique index index_empno on emp(empno);

ii) **Composite index** is an index on two or more columns of a table. To create a composite index, take into consideration the columns that you may use very frequently in a query's WHERE clause as filter conditions. If there are two or more columns that are frequently used in the WHERE clause as filters, the composite index would be the best choice.

The basic syntax is as follows:

```
Create index index_name on table_name(column1,column2);
```

The example is as follows:

```
Create index index_empdept on emp(empno,deptno);
```

e) Write a PL-SQL program to print numbers from 1 to 15 using for loop.

(Correct logic – 2 Marks, Syntax – 2 Marks)

Ans:

```
begin
For i in 1..15 Loop
dbms_output.put_line(i);
end loop;
end;
```

f) What are synonyms? Write a syntax for creating a synonym.

(Synonym definition -2 Marks, Correct syntax -2 Marks)

Ans: A **synonym** is an alternative name for objects such as tables, views, sequences, stored procedures, and other database objects.

You generally use synonyms when you are granting access to an object from another schema and you don't want the users to have to worry about knowing which schema owns the object.

Syntax:

```
CREATE [OR REPLACE] [PUBLIC] SYNONYM [schema.] synonym_name
FOR [schema.] object_name;
```

OR

```
create synonym synonym_name  for object_name;
```

6. Attempt any four of the following: Marks 16

a) Explain shared and exclusive type of lock.

(Each lock- 2 Marks)

Ans: There are various modes in which a data item may be locked. In this section, we restrict our attention to two modes:

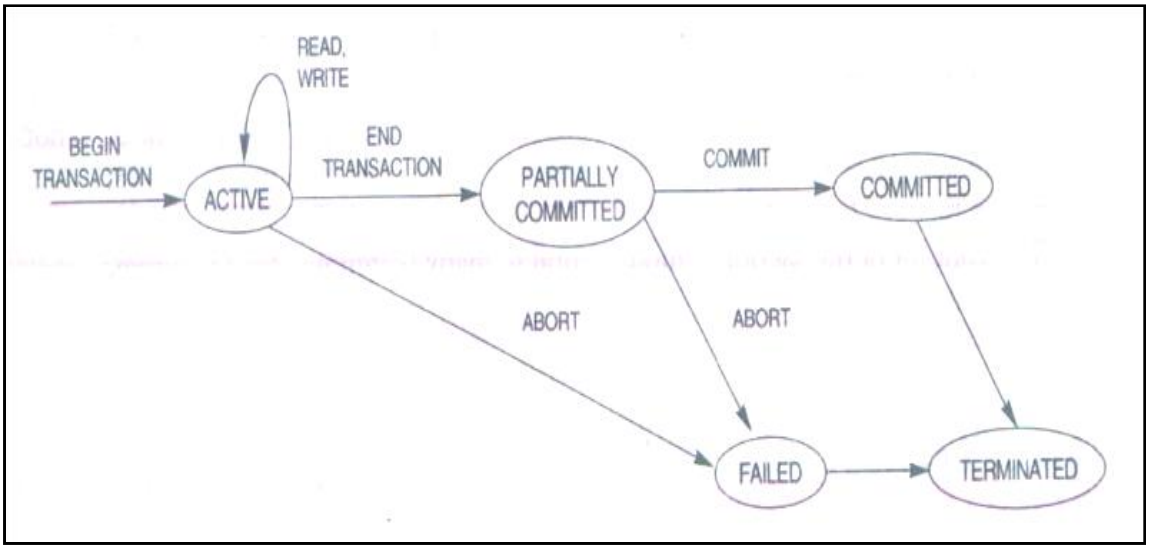
1. Shared. If a transaction *Ti* has obtained a **shared-mode lock** (denoted by S) on item *Q*, then *Ti* can read, but cannot write, *Q*. Shared Lock is provided to the readers of the data. These locks enable all the users to read the concurrent data at the same time, but they are not allowed to change/ 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.

2. Exclusive. If a transaction *Ti* has obtained an **exclusive-mode lock** (denoted by X) on item *Q*, then *Ti* 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.

b) Explain states of transaction with neat diagram.

(Diagram – 2 Marks, Explanation of states- 2 Marks)

Ans:



- **Active**, the initial state; the transaction stays in this state while it is executing
- **Partially committed**, after the final statement has been executed
- **Failed**, after the discovery that normal execution can no longer proceed
- **Aborted or terminated** after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction
- **Committed**, after successful completion

c) Explain trigger with suitable example.

(Trigger definition – 2 Marks, Example -2 Marks)

(Consider any valid example)

Ans: A trigger is a pl/sql block structure which is fired when a DML statements like Insert, Delete, Update is executed on a database table. A trigger is triggered automatically when an associated DML statement is executed.

Syntax for Creating a Trigger

```
 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;
```

OR

```
CREATE TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF}
{INSERT | UPDATE | DELETE}
```

```
 [OF col_name]
 ON table_name
[FOR EACH ROW]
 WHEN (condition)
 BEGIN
  --- sql statements
 END;
```

Example:

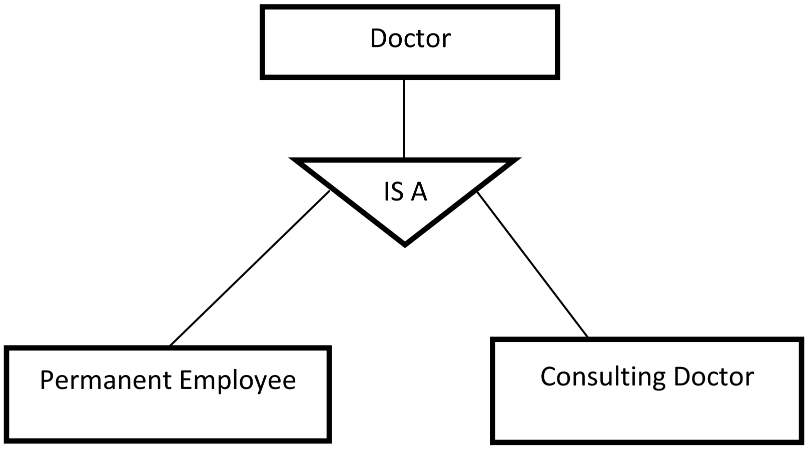
```
CREATE or REPLACE TRIGGER After_Update_Row_product
AFTER
insert On product
FOR EACH ROW
BEGIN
INSERT INTO product_check
Values('After update, Row level',sysdate);
END;
```

d) Explain generalization of ER with suitable diagram.

(Explanation- 2 Marks, Diagram- 2 Marks)

Ans: Generalization is abstracting process of viewing set of object as a single general class.

It means Generalization is the result of taking union of two or more entity set to produce a higher level entity set.



Generalization is a bottom up approach. **Generalization** is containment relationship that exists between a higher-level entity set and one or more lower-level entity sets.

Higher- and lower-level entity sets also may be designated by the terms **superclass** and **subclass**, respectively. The doctor entity set is the superclass of the permanent employee and consulting doctor subclass.

e) Explain BCNF with example.

(Definition – 2 Marks, Example – 2 Marks)

Ans: Usually tables that are in Third Normal Form are already in Boyce Codd Normal Form. Boyce Codd Normal Form (BCNF) is considered a special condition of third Normal form. A table is in BCNF if every determinant is a candidate key. A table can be in 3NF but not in BCNF. This occurs when a non key attribute is a determinant of a key attribute.

OR

A relation schema R is in BCNF with respect to a set F of functional dependencies if, for all functional dependencies in F+ of the form $\alpha \rightarrow \beta$, where $\alpha \subseteq$
R and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \rightarrow \beta$ is a trivial functional dependency (that is, $\beta \subseteq \alpha$).
- α is a super key for schema R.

A database design is in BCNF if each member of the set of relation schemas that constitutes the design is in BCNF.

Example:

As an illustration, consider the following relation schemas and their respective

- Customer-schema = (customer-name, customer-street, customer-city)
  customer-name → customer-street customer-city
- Branch-schema = (branch-name, assets, branch-city)
  branch-name → assets branch-city
- Loan-info-schema = (branch-name, customer-name, loan-number, amount)
  loan-number → amount branch-name

We claim that Customer-schema is in BCNF. We note that a candidate key for the schema is customer-name. The only nontrivial functional dependencies that hold on Customer-schema have customer-name on the left side of the arrow. Since customer-name is a candidate key, functional dependencies with customer-name on the left side do not violate the definition of BCNF. Similarly, it can be shown easily that the relation schema Branch-schema is in BCNF.

f) What is concurrent schedule? Explain with example.

(Explanation – 2 Marks, Any valid Example - 2 Marks)

Ans: It is mechanism to achieve isolation i.e. to control the interaction among the concurrent transactions in order to prevent them from destroying the consistency of the database.

A schedule S is serial if, for every transaction T participating in the schedule, all the operations of T are executed consecutively in the schedule; otherwise, the schedule is called concurrent.

T1	T2

read(A)
A := A –50
write(A)	

	read(A)
	temp := A * 0.1
	A := A temp
	write(A)

read(B)
B := B + 50
write(B)	

	read(B)
	B := B + temp
	write(B)

Let T1 andT2 be two transactions that transfer funds from one account to another. Transaction T1 transfers $50 from account A to account B. It is defined as

T1: read(A);

A := A − 50;

write(A);

read(B);

B := B + 50;

write(B).

Transaction T2 transfers 10 percent of the balance from account A to account B. It is defined as

```
T2: read(A);
temp := A * 0.1;
A := A − temp;
write(A);
read(B);
B := B + temp;
write(B);
```