2 MARKS BOARD QUESTIONS

1) Data Abstraction
ANS:- Hiding complexity of data structures from end user through different levels is known as data abstraction.
Many end users are not computer trained so it is needed to hide complex data structures from them.
It has 3 levels :
a. Physical level
b. logical level
c. view level

---

2) Instance
ANS:- The data stored in database at a particular moment of time is called instance of database.
Example:
Let's say a table teacher in our database whose name is School, suppose the table has 50 records so the instance of the database has 50 records for now and tomorrow we are going to add another fifty records so tomorrow the instance have total 100 records. This is called an instance

---

3) State any two advantages of DBMS.
ANS:- i)Reduction in Data redundancy
ii)Data consistency and integrity
iii)Data security
iv)Privacy
v)Easy access of data
vi)Easy recovery
vii)Flexibility

4) Define Normalization. Enlist its types.
ANS:- Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.
Types of normalization are:
- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)
- Fourth normal form(4NF)

---

5) Write syntax for creating and Renaming a table.
ANS:- Syntax of the CREATE TABLE statement is as follows: CREATE TABLE table_name
(
column1 datatype (size),
column2 datatype(size),
column3 datatype(size),
....
);
Syntax of RENAME TABLE statement is as follows:
RENAME old_table _name To new_table_name ;

---

6) Enlist arithmetic and logical SQL operators.
ANS:- SQL Arithmetic Operators: Addition Operator (+) Subtraction Operator (-)
Multiplication Operator (+)
Division Operator (-)
Modulus Operator (+)
SQL Logical Operators:
ALL operator
AND operator
OR operator
BETWEEN operator
IN operator
NOT operator
ANY operator
LIKE operator
7) Write syntax for creating and dropping views.
ANS:- CREATE VIEW Syntax:
create view view_name As
select column1, column2…
from table_name
where condition ;
DROP VIEW Syntax:
DROP VIEW view_name;

---

8) State two advantages of PL/SQL
ANS:- Advantages of PL/SQL are:
1) Work can be divided into smaller modules so that it can be manageable and also enhances the readability of the code.
2)It promotes reusability.
3) It is secure, as the code is in the database and hides the internal database details from the user.
4)It improves performance against running SQL queries multiple times.

---

9) List disadvantages of typical file processing system.

ANS:- Disadvantages of file processing system

1) Data redundancy and inconsistency
2) Difficulty in accessing data
3) Data isolation
4) Integrity problems

Atomicity problems

10) Define i)Data Abstraction ii)Data Redundancy

ANS:- 1. Data Abstraction :

Many end users are not computer trained so it is needed to hide complex data structures from them.

Hiding complexity of data structures from end user through different levels is known as data abstraction.

It has 3 levels :

a. Physical level
b. logical level
c. view level

2. **Data redundancy** :

The **repetition of information** is known as redundancy .This redundancy leads to higher storage and access cost.

It may lead to data inconsistency, that is different copies of the same data may have different values.

---

11) Define the term:

i) Candidate key
ii) Primary key

ANS:- Candidate key: In a relation, there may be a primary key or may not, but there may be a key or combination of keys which uniquely identify the record. Such a key is called as Candidate key.

OR

A candidate key is a column, or set of columns, in a table that can uniquely identify any database record without referring to any other data.

The candidate key can be simple (having only one attribute) or composite as well.

For Example, {STUD_NO, COURSE_NO} is a composite candidate key for relation STUDENT_COURSE.

Primary key: A key which is selected by the designer to uniquely identify the entity is called as Primary key. A primary key cannot contain duplicate values and it can never contain null values inside it.
Example, RollNo attribute is a primary key for Relation Student.

---

12) List Four DDL commands with syntax.

ANS:- DDL commands

1. 1.Create

Syntax : create table <table_name>(Column_name1 datatype1, column_name2 Datatype2,…Column_nameN DatatypeN);

Drop

Syntax: drop table <table_name>;

3. **Desc**

   Syntax: describe <table_name>;

   OR

   Desc <table_name>

4. **Truncate**

   Syntax: truncate table <table_name>;

5. **Alter**

Syntax: Alter table <table_name> add Column_name Datatype (size);

---

13) Define Normalization, list its types.

ANS:- Normalization:

Normalization can be defined as process of decomposition/division of database tables to avoid the data redundancy.

Types of Normalization:

1. 1NF
2. 2NF
3. 3NF

BCNF

14) Enlist four aggregate functions.

ANS:-

SUM()
AVG()
MAX()
MIN()
COUNT()

---

15) Define Cursor. List the two types of cursor.

ANS:- Cursor: The Oracle Engine uses a work area for its internal processing in order to execute an SQL statement. This work area is private to SQL"s operations and is called a Cursor.

OR

A cursor is a temporary work area created in the system memory when a SQL statement is executed.

Types of cursor are:

1) Implicit cursor
2) Explicit cursor

16) Define i) Data Abstraction ii) Instance.
Ans i) Data Abstraction: Hiding complexity of data structures from end user through different levels is known as data abstraction. Many end users are not computer trained so it is needed to hide complex data structures from them. It has 3 levels : a. Physical level b. logical level c. view level
ii) Instance: The data stored in database at a particular moment of time is called instance of database. Example: Let's say a table teacher in our database whose name is School, suppose the table has 50 records so the instance of the database has 50 records for now and tomorrow we are going to add another fifty records so tomorrow the instance have total 100 records. This is called an instance.

17) State any two advantages of DBMS.
Ans Reduction in Data redundancy
Data consistency and integrity
Data security
Privacy
Easy access of data
Easy recovery
Flexibility

18) Define Normalization. Enlist its types.
Ans Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.
Types of normalization are:
• First normal form(1NF)
• Second normal form(2NF)
• Third normal form(3NF)
• Boyce & Codd normal form (BCNF)
• Fourth normal form(4NF)

19) Enlist arithmetic and logical SQL operators.
Ans SQL Arithmetic Operators:
Addition Operator (+)
Subtraction Operator (-)
Multiplication Operator (+)
Division Operator (-)
Modulus Operator (+)
SQL Logical Operators:
ALL operator

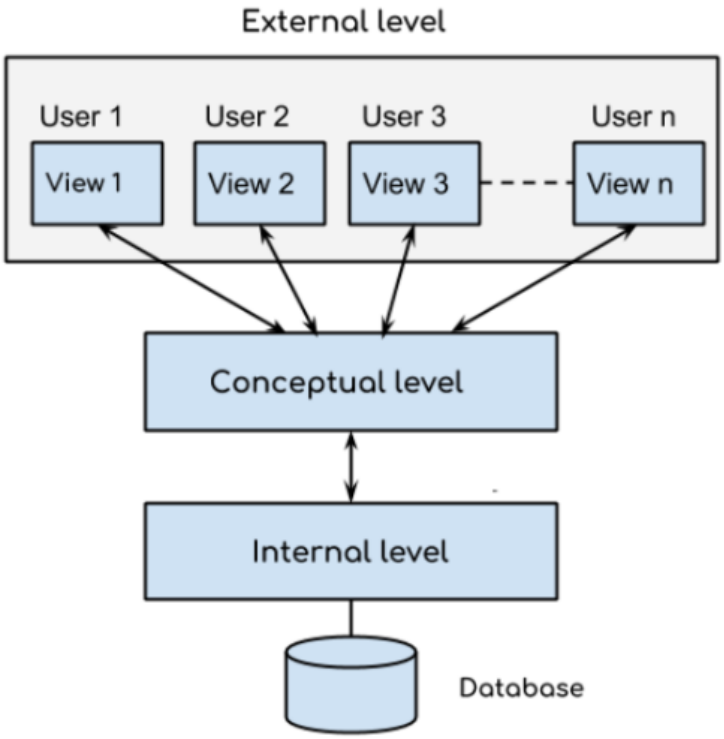
AND operator
OR operator
BETWEEN operator
IN operator
NOT operator
ANY operator
LIKE operator

20) Write syntax for creating and dropping views.
Ans CREATE VIEW Syntax: create view view_name As
select column1, column2…
from table_name
where condition ;
DROP VIEW Syntax:
DROP VIEW view_name;

21) State two advantages of PL/SQL 2 M
Ans Advantages of PL/SQL are:
Work can be divided into smaller modules so that it can be manageable and also enhances the readability of the code.
It promotes reusability.
It is secure, as the code is in the database and hides the internal database details from the user.
It improves performance against running SQL queries multiple times

4MARKS BOARD QUESTIONS

1) Explain three level architecture of Database system.



ANS:-

This architecture has three levels:

1. External level
2. Conceptual level
3. Internal level

1. External level

It is also called view level because several users can view their desired data from this level which is internally fetched from database with the help of conceptual and internal level mapping.

The user doesn't need to know the database schema details such as data structure; table definition etc. user is only concerned about data which is what returned back to the view level after it has been fetched from database which is present at the internal level.

2. Conceptual level

It is also called logical level. The whole design of the database such as relationship among data, schema of data etc. are described in this level.

Database constraints and security are also implemented in this level of architecture

This level is maintained by DBA (database administrator).

3. Internal level

This level is also known as physical level. This level describes how the data is stored in the storage devices.

This level is also responsible for allocating space to the data. This is the lowest level of the architecture.

---

2) Write SQL queries for following:

i) Create table student with following attributes using suitable data types. Roll no., as primary key, name, marks as not null and city.

ii) Add column Date of Birth in above student table.

iii) Increase the size of attribute name by 10 in above student table. iv) Change name of Student table to stud.

ANS:- i)

CREATE TABLE Student (Rollno int PRIMARY KEY, name varchar(30) NOT NULL, marks int NOT NULL, city varchar(20));

ii) ALTER TABLE student ADD DateofBirth varchar(20);

iii) ALTER TABLE student Modify name varchar(40);

iv) RENAME Student to Stud;

3) Write and Explain the syntax for creating and dropping indexes with an example.

ANS:- CREATE INDEX

The CREATE INDEX command is used to create indexes in tables. It allows duplicate values. Indexes are used to retrieve data from the database very fast. The users cannot see the indexes; they are just used to speed up searches/queries.

Syntax:

CREATE INDEX index_name

ON table_name (column1, column2, ...);

Example:

The following SQL creates an index named id_firstname on the FirstName column in the Student table:

CREATE INDEX id_firstname

ON Student (FirstName);

DROP INDEX

The DROP INDEX statement is used to delete an index in a table.

Syntax:

DROP INDEX index_name ON table_name;

Example:

DROP INDEX id_firstname ON Student;

4) Write a PL/SQL code to print reverse of a number.

ANS:- PL/SQL code to print reverse of a number:

```
declare
n number;
i number;
rev number:=0;
r number;
begin
n:=&n;
while n>0
loop
r:=mod(n,10);
rev:=(rev*10)+r;
n:=trunc(n/10);
end loop;
```

```
dbms_output.put_line('reverse is '||rev);
end;
```

5) Distinguish between network model and hierarchical model.

ANS:-

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

6) Explain any four String functions with example

ANS:- **i**) Lower(char)-

Returns the input string with all letters in lower case. Example: SQL>Select lower ('RAJESH') from dual;
Output: rajesh

ii) Upper(char)-

Returns the input string with all letters in upper case.

Example: SQL>Select upper ('rajesh') from dual;

Output: RAJESH

iii) Ltrim(char,set)-

It removes or trims from left of character string

. Example: SQL>Select Ltrim('university','univ') from dual;

Output: ersity

iv) Rtrim(char,set)-

It removes or trims from right of character string. Example:

SQL>Select Rtrim('university','sity') from dual;

Output: univer

v) Length(char)-

It returns length of character string.

Example: SQL> Select length('University') from dual; Output:10

vi) Concat(str1,str2,...)-

Returns the string that result from concatenating the arguments.

Example: Select Concat('employee', 'name') from dual;

Output: employeename

vii) Lpad(str, len, padstr)-

Returns the string str, left-padded with the string padstr to a length of len characters.
Example: Select lpad(ename,10.'*') from emp where empno=7782;

viii) Rpad(str,len,padstr)-

Returns the string str, right-padded with the string padstr to a length of len characters. Example: Select rpad(ename,10.'*') from emp where empno=7782;

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

7) Describe exception handling in brief

ANS:- 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.

Syntax:
DECLARE ;
Declaration section
…executable statement;

EXCEPTION

WHEN ex_name1 THEN ;
Error handling statements/user defined action to be carried out; END;

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);

8) Describe commit and rollback with syntax and example

ANS:- **Commit:**

The COMMIT command saves all transactions to the database since the last COMMIT or ROLLBACK command

The syntax: SQL> COMMIT; Or

COMMIT WORK;

Example :

SQL>Commit;

Rollback:

The ROLLBACK command is used to undo transactions that have not already been saved to the database.

The ROLLBACK command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.

The syntax for ROLLBACK is: ROLLBACK

TO SAVEPOINT_NAME; OR

ROLLBACK;

OR

ROLLBACK WORK;

Example: SQL>ROLLBACK;

9) Explain joins in SQL with examples.

ANS:- JOIN:

A SQL join is an instruction to combine data from two sets of data (i.e. two tables). A JOIN clause is used to combine rows from two or more tables, based on a related column between them. SQL Join types are as follows:

1) INNER JOIN or EQUI JOIN:

A join which is based on equalities is called equi join. In equi join comparison operator "=" is used to perform a Join.

Syntax:

SELECT tablename.column1_name,tablename.column1_name

FROM table_name1,table_name2

where table_name1.column_name=table_name2.column_name;

Example:

Select stud_info.stud_name, stud_info.branch_code, branch_details.location

From stud_info, branch_details

Where Stud_info.branch_code=branch_details.branch_code;

2) SELF JOIN:

The SQL SELF JOIN is used to join a table to itself, as if the table were two tables, temporarily renaming at least one table in the SQL statement.

Syntax:

SELECT a.column_name, b.column_name

FROM table1 a, table1 b

WHERE a.common_filed = b.common_field;

Example:

Select x.stud_name, y.stud_name

from stud_info x, stud_info y Where

x.leader= y.stud_id;

3) LEFT OUTER JOIN:

A left outer join retains all of the rows of the “left” table, regardless of whether there is a row that matches on the “right” table.

Syntax:

Select column1name,column2name

from table1name any_alias1 ,table2name any_alias2
on any_alias1.columnname(+) = any_alias2.columnname;

OR

```
Select column1name,column2name
from table1name left outer join table2name
 on table1name.columnname= table2name.columnname;
```

Example:

```
select last_name, department_name
from employees e, departments d
on e.department_id(+) = d.department_id;
```

OR

```
 select last_name, department_name
from employees left outer join departments
on employees.department_id = departments.department_id;
```

4) RIGHT OUTER JOIN:

A right outer join retains all of the rows of the "right" table, regardless of whether there is a row that matches on the "left" table.

Syntax:

```
Select column1name, column2name
from table1name any_alias1, table2name any_alias2
on any_alias1.columnname =any_alias2.columnname (+);
```

OR

```
Select column1name, column2name
from table1name any_alias1 right outer join table2 name any_alias2
 on any_alias1.columnname =any_alias2.columnname;
```

Example:

Select last_name,department_name from employees e, departments d on e.department_id = d.department_id(+);

OR

Select last_name, department_name

from employees e right outer join departments d

on e.department_id = d.department_id;

5) **NON EQUI JOIN:**

Non equi joins is used to return result from two or more tables where exact join is not possible.

Syntax:

Select aliasname.column1name, aliasname.column2name from tablename alias where <condition using range>;

For example:

In emp table and salgrade table. The salgrade table contains grade and their low salary and high salary. Suppose you want to find the grade of employees based on their salaries then you can use NON EQUI join.

Select e.empno, e.ename, e.sal, s.grade

from emp e, salgrade s

where e.sal between s.lowsal and s.hisal;

10) Explain function in PL/SQL with example.

ANS:- Function:

Function is a logically grouped set of SQL and Pl/SQL statements that perform a specific task. A function is same as a procedure except that it returns a value. A function is created using the CREATE FUNCTION statement.

Syntax:

```
CREATE [OR REPLACE] FUNCTION function_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN return_datatype
{IS | AS}
BEGIN
< function_body > END [function_name];
```

Where,

- *function-name* specifies the name of the function.
- [OR REPLACE] option allows the modification of an existing function.
- The optional parameter list contains name, mode and types of the parameters. IN represents the value that will be passed from outside and OUT represents the parameter that will be used to return a value outside of the procedure.
- The function must contain a **return** statement.
- The *RETURN* clause specifies the data type you are going to return from the function.
- *function-body* contains the executable part.
- The AS keyword is used instead of the IS keyword for creating a standalone function.

Example:

```
CREATE OR REPLACE FUNCTION Success_cnt

RETURN number

IS    cnt number(7) := 0;

BEGIN

SELECT count(*) into cnt

FROM candidate where result='Pass';

  RETURN cnt;
```

11) Explain three level architecture of Database

system.

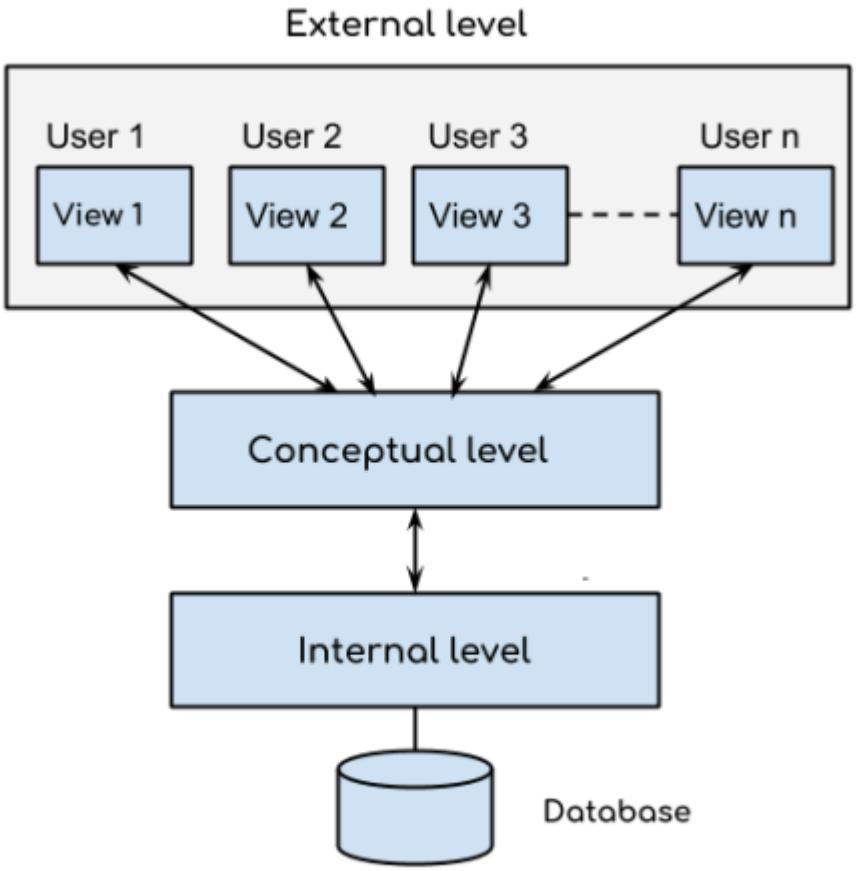


Fig: Three Level Architecture of DBMS

This architecture has three levels:

1. External level

2. Conceptual level

3. Internal level

1. External level It is also called view level because several users can view their desired data from this level which is internally fetched from database with the help of conceptual and internal level mapping. The user

doesn't need to know the database schema details such as data structure; table definition etc. user is only concerned about data which is what returned back to the view level after it has been fetched from database which is present at the internal level. External level is the top level of the three level DBMS architecture.

2. Conceptual level It is also called logical level. The whole design of the database such as relationship among data, schema of data etc. are described in this level. Database constraints and security are also implemented in this level of architecture.

This level is maintained by DBA (database administrator).

3. Internal level This level is also known as physical level. This level describes how the data is stored in the storage devices. This level is also responsible for allocating space to the data. This is the lowest level of the architecture.

12)Write SQL queries for following:

i) Create table student with following attributes using suitable data types. Roll no., as primary key, name, marks as not null and city.

ii) Add column Date of Birth in above student table.

iii) Increase the size of attribute name by 10 in

above student table.

iv) Change name of Student table to stud

Ans : i)

```
CREATE TABLE Student
( Rollno int PRIMARY KEY,
name varchar(30) NOT NULL,
marks int NOT NULL,
city varchar(20)
);
```

ii)

```
ALTER TABLE student
ADD DateofBirth varchar(20);
```

iii)

```
ALTER TABLE student
Modify name varchar(40);
```

iv)

```
RENAME Student to Stud;
```

13) Write and Explain the syntax for creating and dropping indexes with an example

Ans: CREATE INDEX

The CREATE INDEX command is used to create indexes in tables. It allows duplicate values. Indexes are used to retrieve data from

the database very fast. The

users cannot see the indexes; they are just used to

speed up searches/queries.

Syntax:

CREATE INDEX index_name

ON table_name (column1, column2, ...);

Example:

The following SQL creates an index named

id_firstname on the FirstName column in

the Student table:

CREATE INDEX id_firstname

ON Student (FirstName);

DROP INDEX

The DROP INDEX statement is used to delete an index

in a table.

Syntax:

DROP INDEX index_name ON table_name;

Example:

DROP INDEX id_firstname ON Student;

14) Write a PL/SQL code to print reverse of a number

Ans: PL/SQL code to print reverse of a number:

declare

n number;

i number;

```
rev number:=0;
r number;
begin
n:=&n;
while n>0
loop
r:=mod(n,10);
rev:=(rev*10)+r;
n:=trunc(n/10);
end loop;
dbms_output.put_line('reverse is '||rev);
end;
```

15) Write down any four Dr. E.F Codd's rules

Ans: Dr. E. F. Codd's Rules:

Rule 1: The information Rule: all data viewed to users.

Rule 2: Guaranteed Access Rule: all user get access to database

Rule 3: Systematic treatment of null values: null value should be empty

Rule 4: Dynamic online Catalog: record all transactions in database

Rule 5: Data Sub language rule: use only one language

Rule 6: View updating rule: table and view updated simultaneously

Rule 7: High level insert, delete and update: multiple

insert delete update

Rule 8: Physical data independence: hardware change

Rule 9: Logical data independence: structure change

Rule 10: Integrity independence: store correct data

Rule 11: Distribution independence: distributed database

Rule 12: No subversion rule: no version of language used.

16) State the use of group by and order by clauses.

Ans: Group by Clause:

Group by clause is used to collect the data as multiple records and group them to produce the result.

Syntax:

1. SELECT column_name, function(column_name)
2. FROM table_name
3. WHERE condition
4. GROUP BY column_name;

Ex: select avg(sal) from emp_details group by deptno;

Order by Clause:

To view the data in sorted order, the order by clause is used.

By default, the data is sorted in ascending order.

Syntax:

5. SELECT expressions

6. FROM tables

7. [WHERE conditions]

8. ORDER BY expression [ASC | DESC];

Ex: select deptno from emp_details order by deptno;

Ex: select deptno from emp_details order by deptno

desc; (for descending order)

17) Explain the steps of cursor implementation with syntax and example.

Ans: Cursor Implementation:

Steps to create Cursor:

a. Declaring cursor:

Cursor is declared in the declaration section.

Syntax: cursor <cursor_name>is<select query>;

Ex: cursor a is select ename from emp_details where

empno=3;

b. Opening cursor:

After declaring the cursor, the cursor needs to open.

Syntax: open <cursor_name>;

Ex: open a;

c. Fetching a record from cursor:

Once the cursor is declared and opened, we need to

get records or rows from

the cursor. These records are accessed using the

FETCH statement.

Syntax: fetch <cursor_name> into <variable_list>;

Ex: fetch a into name;

d. Closing cursor:

Once the cursor is opened and processing is over, we need to close it.

Syntax: close <cursor_name>;

Ex: close a;

Example:

```
declare
name emp_details.ename%type;
cursor a is select ename from emp_details where
empno=3;//cursor declaration
begin
open a;//opening the cursor
loop
fetch a into name;//fetching the rows from cursor
update emp_details set comm=3000 where empno=3;
exit when a%notfound;
dbms_output.put_line('record updated');
end loop;
close a;//closing the cursor
end;
```

18) Explain ACID properties of transaction.

Ans: ACID Properties of Transaction:

1. Atomicity

2. Consistency

3. Isolation

4. Durability

1. Atomicity:

Atomicity means all the operations included in the

single transaction gets executed at a

time or none.

2. Consistency:

Consistency means update or edits the same data

stored at different locations.

3. Isolation:

Isolation means all the transactions gets executed

independent of each other.

4. Durability:

Durability means data can be saved in database

permanently until user change it

19) Describe any four responsibilities of Database Administrator.

Ans: Responsibilities of Database Administrator (DBA):

1. Schema Definition:

Database or schema can be designed or defined by

DBA.

2. Creating storage structure:

DBA allocate or decide the space to store the

database.

3. Create grant access methods:

Different access methods to access the database can be granted by DBA to the users.

4. Schema modification:

The database or schema which is already defined can be modified by DBA as per the requirements.

5. Granting authorization:

To access the different databases, DBA can grant the authorization to authorized users only.

6. Performance tuning:

The problems/errors arise in database accessing; can be resolved by DBA to increase the performance.

7. Regular maintenance:

DBA can monitor the transactions in database and maintain the database error free by doing the regular maintenance.

20) Explain Primary and Unique key constraint with syntax.

Ans: Primary key constraint:

Primary key constraint applied to any column can't accept the duplicate and null

values.

This constraint can be applied at the time of table

creation.

Syntax for Primary Key:

```
CREATE TABLE <Table_Name>
(
Column1 datatype,
Column2 datatype,CONSTRAINT <Name> PRIMARY
KEY (Column name)
);
```

Example:

```
CREATE TABLE CUSTOMERS
( ID INT NOT NULL,
 AGE INT NOT NULL,
 ADDRESS CHAR (25),
 PRIMARY KEY (ID) );
```

OR

```
CREATE TABLE CUSTOMERS
 (
 ID INT NOT NULL,
 NAME VARCHAR (20) NOT NULL,
 AGE INT NOT NULL,
 ADDRESS CHAR (25) ,
 SALARY DECIMAL (18, 2),
 PRIMARY KEY (ID, NAME)
```

);

Unique key constraint:

Unique key constraint applied to any column can't accept the duplicate values (only accepts unique values).

This constraint can be applied at the time of table creation.

Syntax for Unique key:

```
CREATE TABLE <Table_Name>
(
Column1 datatype,
Column2 datatype,CONSTRAINT <Name> UNIQUE
KEY (Column name)
);
```

Example:

```
CREATE TABLE students
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
City varchar (255),
UNIQUE (S_Id)
)
```

OR

```
CREATE TABLE students
```

(

S_Id int NOT NULL,

LastName varchar (255) NOT NULL,

FirstName varchar (255),

City varchar (255),

CONSTRAINT uc_studentId UNIQUE (S_Id,

LastName)

)

21) Write and Explain the syntax for creating database trigger.

Ans: Database trigger:

Triggers can be referred as stored procedures that are fired or executed when anINSERT, UPDATE or DELETE statement is given against the associated table.

Syntax:

create trigger [trigger_name]

[before | after]

{insert | update | delete}

on [table_name]

[for each row]

[trigger_body]

Explanation of syntax:

1. create trigger [trigger_name]: Creates or replaces an existing trigger with the

trigger_name.

2. [before | after]: This specifies when the trigger will be executed.

3. {insert | update | delete}: This specifies the DML operation.

4. on [table_name]: This specifies the name of the table associated with the trigger.

5. [for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.

6. [trigger_body]: This provides the operation to be performed as trigger is fired

Example:

Given Student Report Database, in which student marks assessment is recorded. In such schema, create a trigger so that the total and percentage of specified marks is automatically inserted whenever a record is insert.

Here, as trigger will invoke before record is inserted so, BEFORE Tag can be used.

```
create trigger stud_marks
before INSERT
on
Student
for each row
```

set Student.total = Student.subj1 + Student.subj2 +

Student.subj3, Student.per

= Student.total * 60 / 100;

22) Explain Database Recovery techniques in detail.

Ans: Database Recovery Techniques:

Database recovery techniques are used to restore the

original data in system from

backup.

Backward and forward recovery is two types of

database recovery.

Recovery Techniques:

1. Log based recovery.

2. Shadow paging recovery

3. Checkpoints

1. Log based recovery:

It records sequence of log records, which includes all

activities done by database

users.

It records the activities when user changes the

database.

In case of database failure, by referring the log records

users can easily recover the

data.

2. Shadow paging recovery:

This technique is the alternative for log based recovery.

In this technique, database is divided into pages that can be stored on the disk.

The page table is used to maintain the record of location of pages.

In case of database failure, page table is used to recover the parts of database.

3. Checkpoints:

Checkpoint records all committed transactions into logs.

When system fails, it check log to determine recovery action.

23) Draw the overall architecture of DBMS. Explain storage manager and query processor components.

Ans: Storage manager components:

1. Buffer Manager

The Buffer Manager allocates the space to the buffer to store data in it.

2. File Manager

The task of the file manager is to manage the space allocation in disk for storing the information and also the data structures used for representing that information.

3. Authorization and Integrity Manager

Allows only authorized users to access data and should be hidden from the public

users. The Authorization and Integrity Manager verifies the authority of the user trying to access the data and it also checks the integrity constraints when the database is modified.

4. Transaction Manager

A transaction in DBMS is nothing but a very small unit of the program. The Transaction Manager manages all the transaction (program) execution.

Fig: Overall architecture of DBBMS

Query Processor Components:

Query processor handles the query processing. It processes the query given by the user.

1. DDL Interpreter:

DDL expands to Data Definition Language. DDL Interpreter interprets the DDL statements such as schema definition statements like create, delete, etc.

2. DML Compiler:

DML expands to Data Manipulation Language. DML Compiler compiles (or translates) the DML statements such as select, update and delete statements into low

Query Processor Components:

Query processor handles the query processing. It processes the query given by the user.

1. DDL Interpreter:

DDL expands to Data Definition Language. DDL Interpreter interprets the DDL statements such as schema definition statements like create, delete, etc.

2. DML Compiler:

DML expands to Data Manipulation Language. DML Compiler compiles (or translates) the DML statements such as select, update and delete statements into low level instructions which is nothing but the machine-readable object code to make it executable.

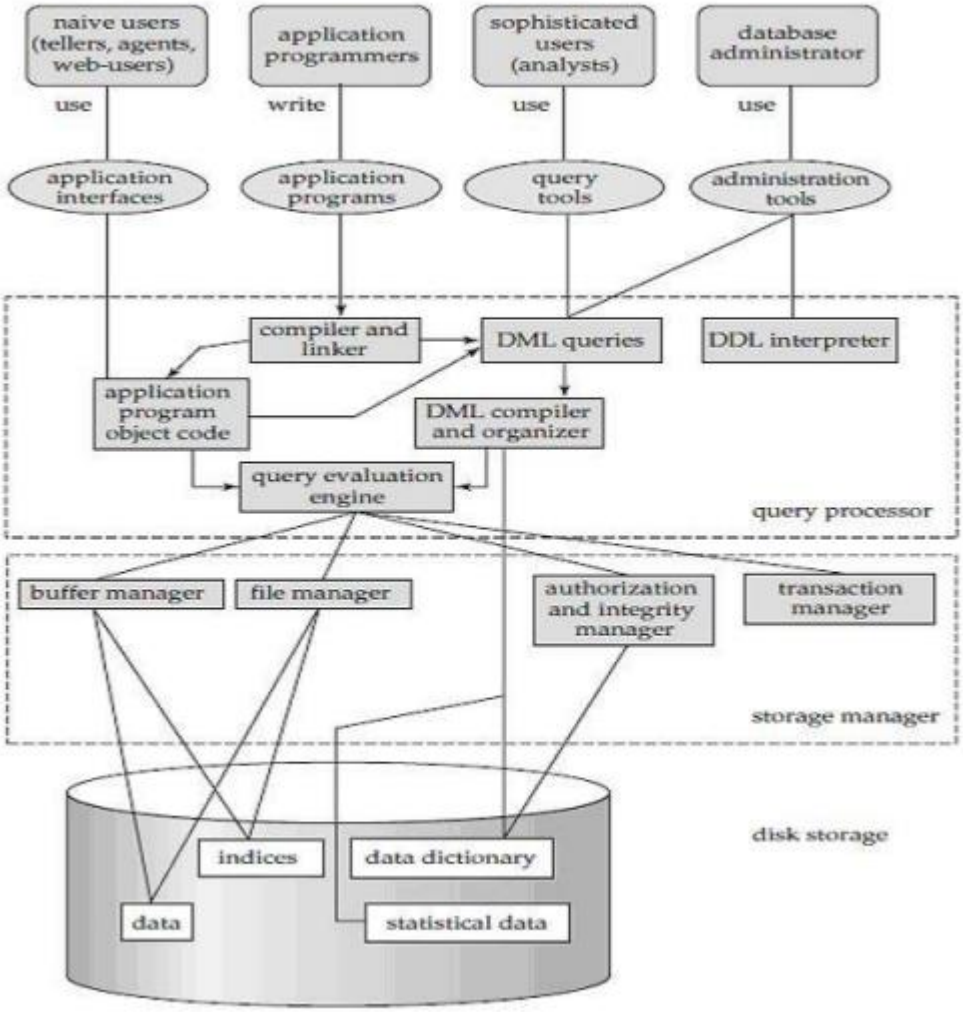
The DML compiler is also responsible for query optimization.

3. Query Evaluation Engine:

The Query Evaluation Engine is also referred as "Query execution engine".

Query Evaluation Engine first interprets the SQL commands to access the data from the database and then it returns the answer to that

query.



24) Write the SQL queries for following EMP table. Emp (empno, deptno, ename, salary, designation, city.) i) Display average salary of all employees. ii) Display names of employees who stay in Mumbai or Pune. iii) Set the salary of employee 'Ramesh' to 50000. iv)Display names of

employees whose salaries are less than 50000. v)

Remove the Record of employees whose deptno is

10. vi) Remove the column deptno from EMP table.

Ans: i. select avg(salary) from emp;

ii. select ename from emp where city='Mumbai' or

city='Pune';

iii. update emp set salary=50000 where

ename='Ramesh';

iv. select ename from emp where

salary<50000;

v. delete from emp where deptno=10;

vi. alter table emp drop column deptno;

25) Write and Explain the syntax for creating,

Altering and dropping the sequence.

Ans: Syntax for creating sequence: CREATE

SEQUENCE sequence_name START WITH

initial_value

INCREMENT BY increment_value MINVALUE

minimum value

MAXVALUE maximum value CYCLE|NOCYCLE ;

.where as:

sequence_name: Name of the sequence.

initial_value: starting value from where the sequence

starts.

Initial_value should be greater than or equal

to minimum value and less than equal to maximum

value.

increment_value: Value by which sequence will

increment itself.

Increment_value can be positive or negative.

minimum_value: Minimum value of the sequence.

maximum_value: Maximum value of the sequence.

cycle: When sequence reaches its set_limit

it starts from beginning.

nocycle: An exception will be thrown

if sequence exceeds its max_value.

Example:

CREATE SEQUENCE sequence_1

start with 1

increment by 1

minvalue 0

maxvalue 100

cycle;

Alter sequence:

Syntax:

alter sequence <sequence_name> maxvalue

<number>;

Alter sequence can change the maxvalue in the

sequence created.

Dropping sequence:

Syntax:

drop sequence <sequence_name>;

To drop the sequence the DROP command is used.

26) Write SQL queries for following. Consider table

stud (roll no, name, subl,

sub2, sub3)

i) Display names of student who got minimum mark in

subl.

ii) Display names of students who got above 40 marks

in sub2.

iii) Display count of Students failed in sub2.

iv) Display average marks of subl of all students.

v) Display names of students whose name start with 'A'

by arranging them

in ascending order of subl marks.

vi) Display student name whose name ends with h' and

subject 2 marks are between 60 to 75.

27) Write a PL/SQL code to check whether specified

employee is present in

Emp table or not. Accept empno from user. If

employee does not exist

display message using exception handling.

Ans: PL/SQL Program:

declare

```
no emp.empno%type;
begin
no:=&no;
put_line('Empno not present');select empno into no
from emp where empno=no;
dbms_output.put_line('Empno is present: '||no);
exception
when no_data_found then
dbms_output. line (‘Empno n
end;
```



```
        END;
    /
```