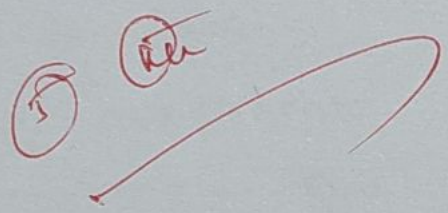
Assignment - 1

Name : R. Akshitha

VTU : 29072

course code : 10211CA207

course title : Database management systems.



-plain in detail about Data base archi-cture with neat diagram.

Database Architecture describes how a atabase management system is designed, ow its components are arranged, and how sers interact with the database.

It defines:

- physical structure → How data is actually stored
- Logical structure → How data is organized logically.
- External Interaction → How users/applications access data.

Key components of Database Architecture.

1. users - End users or applications that request data.
2. DBMS software - Manages data storage, retrieval, security, and occurency.
3. Database (storage) - Stores actual data in files, tables and indexes.
4. Application layer - Handles business logic and rules.

Types of Database Architecture

1. single Tier (1- Tier)

- use directly interacts with DBMS.
- Example: using SQL * plus or MS Access on local machine.
- Not secure for multi-user systems.

2. TWO Tier (2 Tier)

- Divided into client and server
- The client application directly communicates with the database server using SQL queries.
- Example: A Java application connecting directly to oracle DB.
- faster than 1-tier, but still limited scalability.

3. Three Tier (3-Tier)

- Most widely used in modern web & enterprise systems.
- Divided into:

(a) presentation layer (client layer)

- user interface: web browser, mobile app, desktop app.
- collects user input and shows results.

(b) Application layer (Business logic layer)

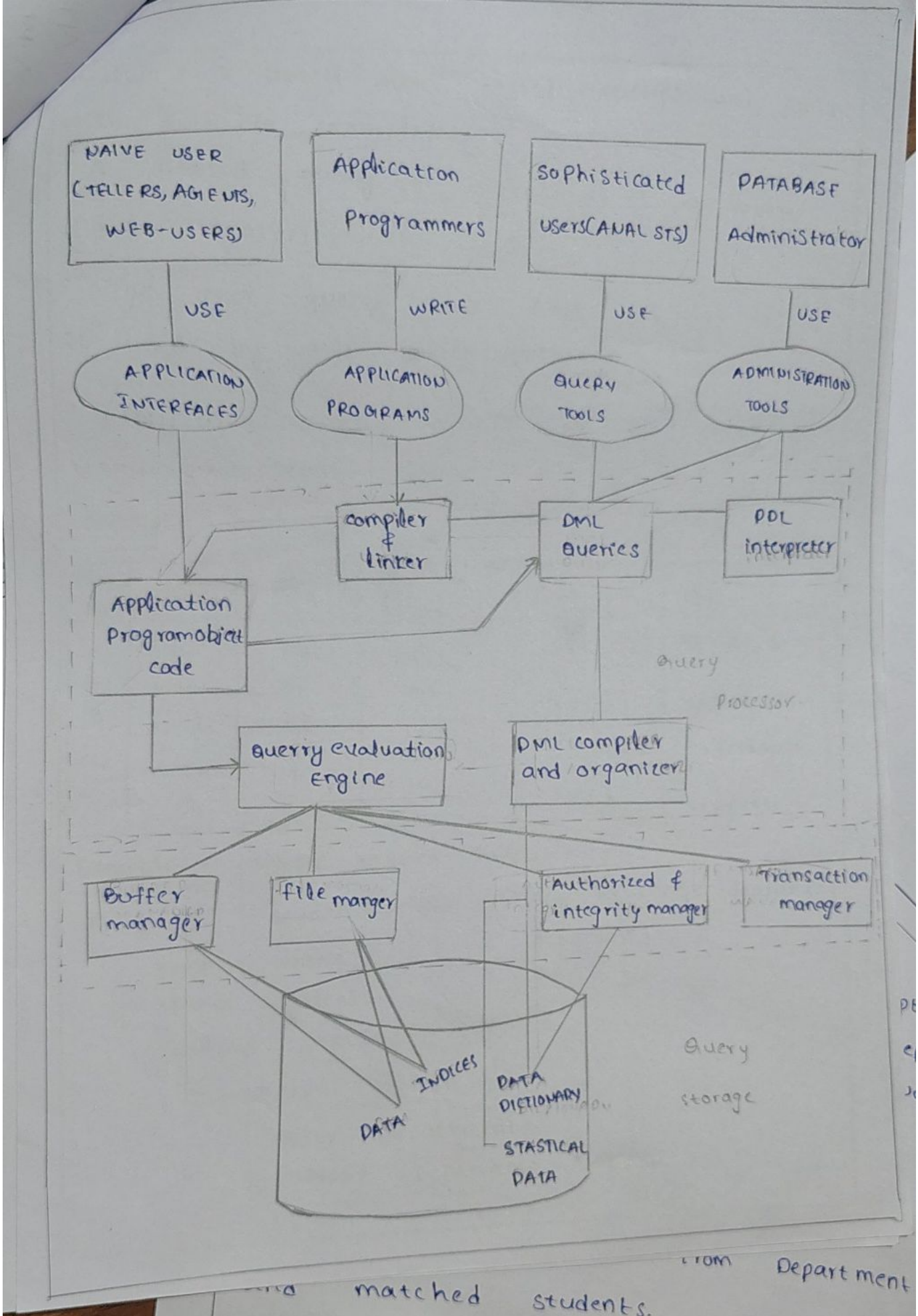
- Middle tier that processes requests.
- Performs authentication, validation, calculations, and sends SQL queries to DBMS.
- Example: Web server or Application server.

(c) Database Layer (DBMS layer)

- Stores actual data.
- DBMS (like MySQL, Oracle, PostgreSQL) manages data, concurrency, backup, and security.
- Advantages.
- High security (users never directly access database).
- Scalability (supports thousands of users).
- Better performance with caching and load balancing.

Why it is important.

- Ensures data consistency & security.
- Supports multi-user environments.
- Provides better performance
- Makes applications scalable and flexible.
- Separates logic from data storage for easier maintenance.



matched students.

Department

Explain in detail about nested queries and joins with suitable examples.

- A nested query (or subquery) is a query inside another query.
- The inner query runs first and its result is used by the outer query.
- They are usually written in the WHERE, HAVING, or FROM clause.

Example Database:

Student ID	Name	Age	Dept ID	Dept name	Phone no
1	Ravi	20	101	CSE	9876543210
2	Sita	21	102	ECE	9876765454
3	Arjun	22	103	Mechanical	7676768291
4	Meena	23	101	Civil	9182767676

Example 1: Simple Nested Query

Find the students who belong to CSE department

```
SELECT Name
FROM stodents
WHERE DeptID = (
    SELECT Dept ID
    FROM Departments
    WHERE Deptname = 'CSE'
);
```

Explanation:

- Inner query finds DeptID of CSE → 101.
- Outer query selects all students with Dept ID 101.

Result: Ravi, Arjun

Example: 2 Nested query with IN

find students who belong to CSE or ECE.

```
SELECT Name
FROM Students
WHERE DeptID IN(
    SELECT Dept ID
    FROM Departments
    WHERE DeptName IN ('CSE', 'ECE')
);
```

2. Joins.

A join is used to combine data from two or more tables based on a related column.

Types of Joins:

(a) INNER JOIN

- Returns only the matching rows from both tables.

```
SELECT Students.Name,
Departments.DeptName
FROM Students
```

```
SELECT Students.Name,
Departments.Dept Name,
 FROM Students
 RIGHT JOIN Departments
ON Students.Dept ID = Departments.Dept ID;
```

If a department has no students, it still appears with NULL for Name.

(d) FULL OUTER JOIN

- Returns all rows from both tables. (match or no match.)
- In My SQL we simulate it with UNION.

```
SELECT Students.Name,
Departments.Dept Name
 FROM Students
 LEFT JOIN Departments
 ON Students.Dept ID = Departments.Dept ID
 UNION
 SELECT Students.Name,
 Departments.Dept Name
  FROM Students
  RIGHT JOIN Departments
  ON Students.DeptID = Departments.Department ID;
```

• INNER JOIN Departments
ON Students.DeptID = Departments.Dept ID;
Result:

Ravi	CSE
Sita	ECE
Arjun	CSE
Meena	Mechanical

(b) LEFT JOIN(OUTER JOIN)

• Returns all rows from the left table (Students) and matched rows from Departments.

• if no match → NULL

```
SELECT Students.Name,
Departments.Dept Name
FROM Students
LEFT JOIN Departments
ON Students.Dept ID = Departments.Dept ID;
```

if same result here, but if a student's Dept ID didn't Exist, Dept Name would be NULL).

(c) Right JOIN

• opposite of LEFT JOIN

• Returns all rows from Departments, and matched students.

DBMS Assignment-3

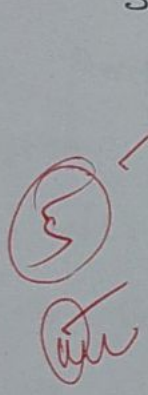
Name : R. Akshitha

VTU. no : 29072

subject
title : database management system.

subject
code : 10211CA207.

Slot : S10L12.



Unit - III :

Normalization and its various types of normalization.

Definition:

Normalization is the process of organizing data. in a database to reduce data redundancy and improve data. Integrity. It divides large Tables into smaller, related tables and defines relationships between them.

objectives of normalization:-

→ Eliminate redundant data
→ Ensure data dependencies make sense.
→ Improve query Performance.

Types of Normalization:

1) first Normal form (1NF):-

→ Each column contains atomic values.
→ Each record is unique.

Ex:-

Student ID	Name	Subject
1	Ravi	math
1	Ravi	science.

2) Second normal form (2NF):-

→ Must be in 1NF.
→ No Partial dependency.

Ex: if a table's primary key is (student ID, course ID), and student name depends only on student ID, move student name to another table.

3) Third normal form (3NF):
→ Must be in 3NF
→ No Transitive dependency.

Ex: If student ID → Department ID → Department name, move Department details to a separate table.

4) Boyce-codd normal form (BCNF):
→ A stronger version of 3NF.
→ for every functional dependency.
→ Ensures highest level of data integrity for most cases.

5) fourth normal form (4NF):
→ must be in BCNF.
→ Removes multi-valued dependencies.

6) fifth normal form (5NF):
→ must be in 4NF.
→ Removes join dependency ensuring that data cannot be reconstructed incorrectly by joining multiple tables.

DBMS Assignment - 4

Name : R. Akshitha.

VTU no : 29072

Subject

title : Database management system.

subject

code : 10211CA207.

slot : S10L12

(5)

Unit - IV:

Explain about deadlock and it's handling.

Deadlock in operating systems:

Definition: A deadlock is a situation in a multi-Process system where two or more processes are blocked forever, each waiting for a resource that is held by another process.

Example of deadlock:

→ Process P_1 holds Resource R_1 and needs R_2 to continue.

→ Process P_2 holds Resource R_2 and needs R_1 to continue.

→ Both are waiting for each other indefinitely this is a deadlock.

Necessary conditions for Deadlock:

Deadlock occurs only if all four of these conditions hold simultaneously:

1) Mutual Exclusion:

At least one resource must be held in a non-shareable mode.

2) Hold and wait:

A Process holding at least one resource is waiting to acquire additional resources held by others.

3) No Preemption: Resources cannot be forcibly taken away from a Process : They must be released voluntarily.

Deadlock Handling methods:

1) Deadlock Prevention:

Prevent any one of the four conditions from occuring.

→ mutual Exclusion: make some resources shareable.

→ Hold and wait :- Require a process to request all resources at once.

→ circular wait :- Impose an order of resource requests.

2) deadlock Avoidance:

The system checks every resource request and decides whether granting it may lead to a deadlock.

→ uses algorithm's like the Banker's algorithm.

3) deadlock detection:

→ system allows deadlocks to occur but detects them periodically.

→ A wait-for graph (WFG) is used. if a cycle is found, a deadlock exists.

4) Deadlock Recovery:

→ used After detection to remove deadlock.

→ Transaction Rollback - undo or more transaction.

→ Transaction Termination - Abort one or more transaction to break the cycle.

DBMS Assignment - 5

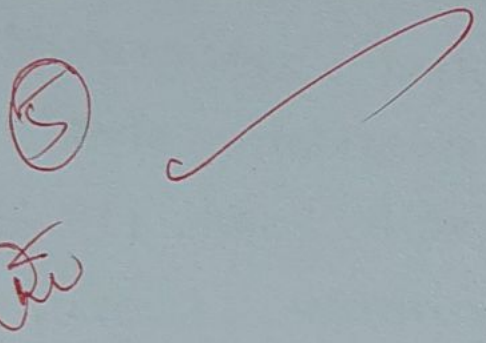
Name : R. Akshitha

VTU. no : 29072

subject title : Database Management System.

subject code : 102U CA207.

slot : S10 212.



Unit-V

Explain about RAID storage and it's types.

Definition: RAID (Redundant Array of independent disks) is a data storage technology that combines multiple physical hard drives into the logical unit to:

→ Increase performance (speed).
→ Provide fault tolerance (data protection).
→ Expand storage capacity.

Types of RAID:

1) RAID - 0 Stripping:-

→ Data is split into blocks and stored across multiple disks.
→ No duplication or parity.
→ Performance high
→ fault tolerance : None

Ex: if one disk fails, all data is lost.

2) RAID 1 - Mirroring:-

→ data is duplicated (intarred) on two or more disks.
→ If one disk fails, data can be retrieved from the other.
→ Performance moderate.
→ fault tolerance high.

Ex: Two disks stored in the same data.

3) RAID 2 :-
→ uses bit-level stripping and Hamming code for error correction.
→ Rarely used today because it's complex and costly.
4) RAID 3 :-
→ uses byte-level striping with on dedicated Parity disk for Error correction.
→ Good for large sequential data Transfers.
→ Drawback :- Parity disk becomes a bottleneck.
5) RAID 4 :-
→ similar to RAID3 but uses block-level striping.
→ one dedicated Parity disk.
→ faster reading but slower writing.
6) RAID 5 : uses block level striping with distributed parity across all disks.
→ can tolerate one disk failure.
→ most commonly used RAID level.
7) RAID 6 : Similar to RAID5 but two Parity blocks.
→ can tolerate one disk failure.
8) RAID 10(1+0) :- combines RAID 1 (mirroring) and RAID 0 (striping).
→ Provides both speed and data Protection.
→ Requires at least 4 disk.
→ Performance very high tolerance very high.