Assignment - I

Database management system:

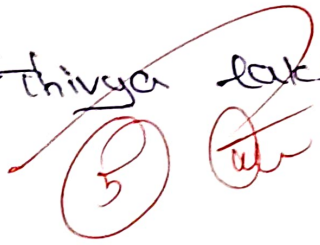
Name: Y. PRATAP

VTU : 31871

SEC : CSE [AI & ML]

course code: 102 11 CA207

faculty name: ~~thivya lakshmi~~



Y. pratap
VTU 31871

Database: system Architecture.

The database system architecture can be divided into three main parts.

users, Query processor, and storage manager, with the disk storage at the bottom.

1. users:

different types of users interact with database.

* native users -> use application interfaces.
* sophisticated users - use query tools.
* Database administrators -> use administrator tools.

2. Query processor:

This is responsible for interpreting and executing queries.

* Application program object code: Granted by compiler and linker
* DML Queries: Data manipulation language.
* DDL: interpreter: interpretes shedule definations

Query Evaluvation engine: executes optimized queries.

3. storage manager:

The storage manager controls how data is stored and retrieved

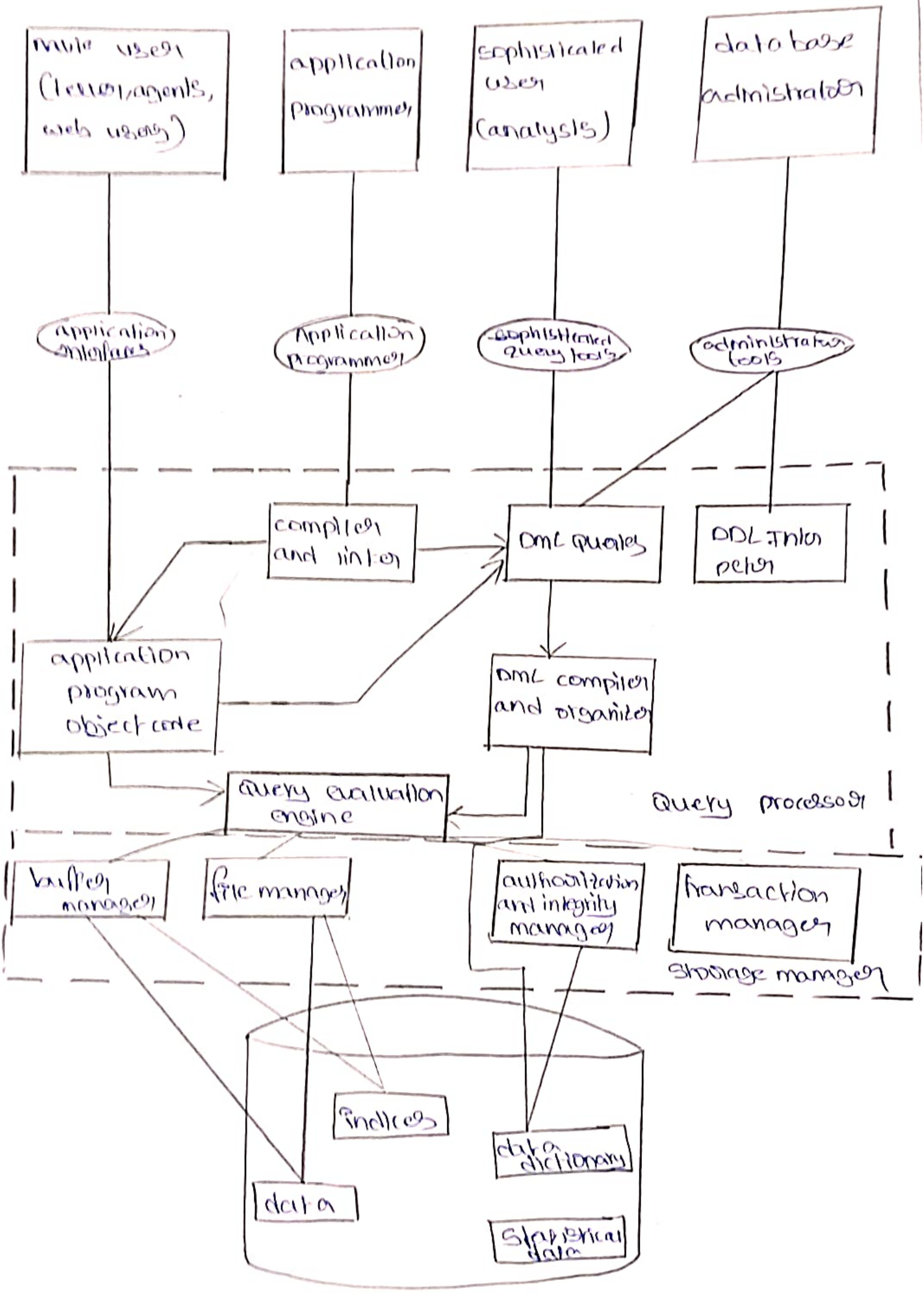
* Buffer manager: minimizes disk I/O by storing. frequently accessed data in memory.
* Transaction manager: ensures consistency, concurrency control, and recovery.
* file manager: manages allocation of space and file storage.

- these components manages communication between query processor and disk storage.

4. Disk storage:

- this is the physical layer where actual data resides.

* Data: tables and records.
* Indices: used for fast searching.
* Data dictionary: stores metadata
* Statistical data: used for query optimization.


naive users (tellers, agents, web users)
application programmers
sophisticated users (analysts)
database administrators
application interfaces
Application programmers
Sophisticated query tools
administration tools
compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
Query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
Storage manager
indices
data dictionary
data
Statistical data

Join Relations

Join operations take two relations and returns as a result

Explain various queries and Joins with suitable example.

Queries in DBMS:

Queries are used to retrieve, insert, update and delete data from a database using SQL.

Types of Queries:

1. Select Query - used to fetch data from a table

```
SELECT name, age FROM students;
```

2. Insert Query - used to add new records.

```
INSERT INTO students (id, name, age)
```

3. Update Query - used to ~~add new records~~ update (modify) existing records.

```
UPDATE Student SET age = 21 WHERE id
```

4. Delete Query - used to remove records

```
DELETE FROM students WHERE id = 1;
```

Joined Relations:-

Join operations take two relations and return as a another relation.

These additional operations are typically used as Query expressions in the from clause.

Join conditions - defines which tuples in the two relo match, and what attributes are present in the

Join type - defines how tuples in each relation that do not match any tuple in the other relation are treated.

Join types
inner join
left outer join
right outer join
full outer join

Join conditions
natural
on <predicate>
using $(A_1, A_2, \ldots A_n)$

Database Example - Join

Instructor

ID	name	dept_name
10101	Srinivasan	comp sci
12121	wu	finance
15151	mazart	music

Teaches

ID	course-ID
10101	CS-101
12121	FIN-201
15151	BIO-101

Inner Join:

Returns only the matching rows from both tables based on a given conditions.

Example: Display instructors who are teaching atleast one co

One course

```
SELECT instructor.ID, name,
course_id
FROM instructor
INNER JOIN teaches
ON instructor.ID = teaches.ID;
```

ID	name	course-id
10101	Srinivasan	CS-101
12121	Wu	Fin-201

Left outer join:

Returns all rows from the left table and the matching rows from the right table. if no match, Nulls are shown for right table columns.

Example: List all instructors including those who are not teaching any courses.

```
SELECT instructor.ID, name, course-id
FROM instructor
LEFT JOIN teaches
ON instructor.ID = teaches.ID;
```

Right

ID	name	course-id
10101	Srinivasan	CS-101
12121	Wu	Fin-201
15151	Mazart	NULL

ID	name	course-id
10101	Srinivasan	CS-101
12121	Wu	FIN-201
15151	mozart	NULL
76766	NULL	BIO-101

Equi JOIN :

A type of INNER JOIN that uses on equality (= operation to match rows

Example : find instructor-course associations using equality condition

```
SELECT instructor-ID, name, course-id
FROM instructor, teaches
WHERE instructor.ID = teaches.ID;
```

ID	name	course-id
10101	Srinivasan	CS-101
12121	Wu	FIN-201

CROSS JOIN :

Return the cartesian product of two tables - every rows from the first table joined with every row from the second table.

Example: Generate all possible combinations of instructor and courses.

```
SELECT name, course-id,
FROM Instructor,
CROSS JOIN teaches
```

Name	course-id
srinivasan	CS-101
Srinivasan	FIN-201
srinivasan	BIO-101
wu	CS-101
wu	FIN-201
wu	BIO-101
mozart	CS-101
mozart	FIN-201
mozart	BIO-101

Assignment - II

Database management system:

Name: Y. PRATAP

VTU : 31871

SEC : CSE [AI & ML]

Course code: 10211CA207

Faculty name: Thivya Lakshmi

Normalization and its various Types

Normalization is a database design technique that organizes tables to reduce data redundancy and improve data integrity. The process involves dividing large tables into smaller ones and defending relationships between them.

Objectives of Normalization :-

- Eliminate redundant data
- Ensure data dependencies make sense
- make the database more efficient and easier to maintain

Types of Normalization:

1. First Normal form (1NF) :

- Removes repeating groups
- Ensures that all columns contain atomic (indivisible) values
- Each record must be unique

2. Second Normal form (2NF) :

- must be in 1NF
- Removes partial dependencies

3. Third Normal form (3NF) :-

- Must be in 2nf
- Removes transitive dependencies
- Non-Prime attributes are not dependent on other non-Prime attributes

4. Boyce - codd Normal form (BCNF):

- Stricter version of 3NF
- Every determinant must be a candidate key

5. Fourth Normal form (4NF)

- must be in BCNF
- Removes multi-valued dependencies

6. Fifth Normal form (5NF):

- Also called Project - Join Normal form
- Deals with cases where information can be reconstructed from smaller pieces of data.

Deadlock and its Handling

Dead Lock is a situation is an operating system where two or more processes are unable to proceed because each is waiting for the other to release a resource.

conditions for Deadlock :

1. mutual exclusion - only one Process can use a resource at a time
2. Hold and wait - A Process is holding one resource and waiting for another
3. No Preemption - A resource cannot be forcibly taken away
4. circular wait - A set of processes are waiting for each other in a circular chain

Deadlock Handling methods:

1. Dead Lock Prevention:
   - Ensures that at least one of the neccessary condition cannot occur
   - Example : Disallow Hold and wait by requiring all resources to be at once

2. Deadlock avoidance :-
   - The system makes careful resource allocation decisions
   - Banker's Algorithm is popular avoidance technique
   - It ensures the system remains in a safe state

3. Deadlock Detection and Recovery;
   - Allow deadlocks to occurs but detect and cover from them
   - Use algorithms to detect cycles in resource allocation graphs
   - Recover by ;
     - Terminating Process
     - preempting resources

3. RAID 5 - Block-level striping with Parity:

- Data and Parity are distributed across all disks
- Can tolerate on disk failure
- Good balance of Performance and redundancy

4. RAID 6 - Double Parity:

- Similar to RAID 5, but stores two Parity blocks
- Can tolerate two disk failures

5. RAID 10 (1+0) - mirrored sets in a striped set

- Combines RAID 1 and RAID 0
- High Performance and high redundancy
- Requires a minimum of 4 disks

6. RAID 2, 3, 4, 50, 60:

- less common
- Used for specialized needs with trade-offs in costs, Performance, and redundancy.

4. Deadlock Ignorance:

- The system assumes deadlock is rare ignores the problem.
- Used in many operating systems like windows and UNIX

5 RAID Storage and its Types

RAID (Redundant Array of Independent Disks) is a storage virtualization technology that combines multiple physical disk drives into one logical unit to improve performance and provide redundancy

Advantages of RAID:-

- Increased data reliability
- Improved read/write Performance
- Fault tolerance

Types of RAID:

1. RAID 0 - Striping:

- Data is split across multiple disks
- No redundancy, high Performance.
- If one disk fails, all data is lost.

2. RAID 1 - mirroring:

- Duplicates data across two or more disks
- Provides high redundancy, but doubles Storage cost.