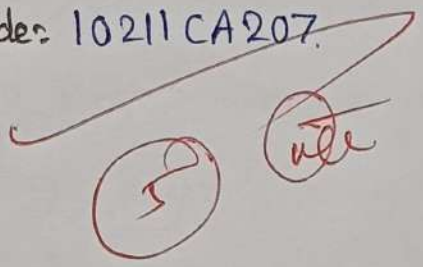
— Data Base Management System —

Assignment

Name: Y. Tarun Kumar Reddy

VTU: 30548

Course Code: 10211CA207



1. Explain in detail about data base system architecture with a neat diagram.

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 user interact with the database:

- Navie users (tellers, angents, web users) → use application interfaces.
- Application Programmers → write application programs.
- Sophisticated users (analysts) → use query tools.
- Database Administrators (DBAS) → use the administration tools.

2. Query Processor.

This is responsible for interpreting and executing the queries.

- Application program object code:
  Generated by compiler and linker
- DML Queries: Data Manipulation language.
  (E.g: SELECT, INSERT, UPDATE, DELETE)
- DDL Interpreter: Interprets schema definition (table, constraints).

- DML Compiler and organizer: Checks syntax and optimizes queries.
- Query Evaluation Engine: Executes optimized queries.

3. Storage Manager:

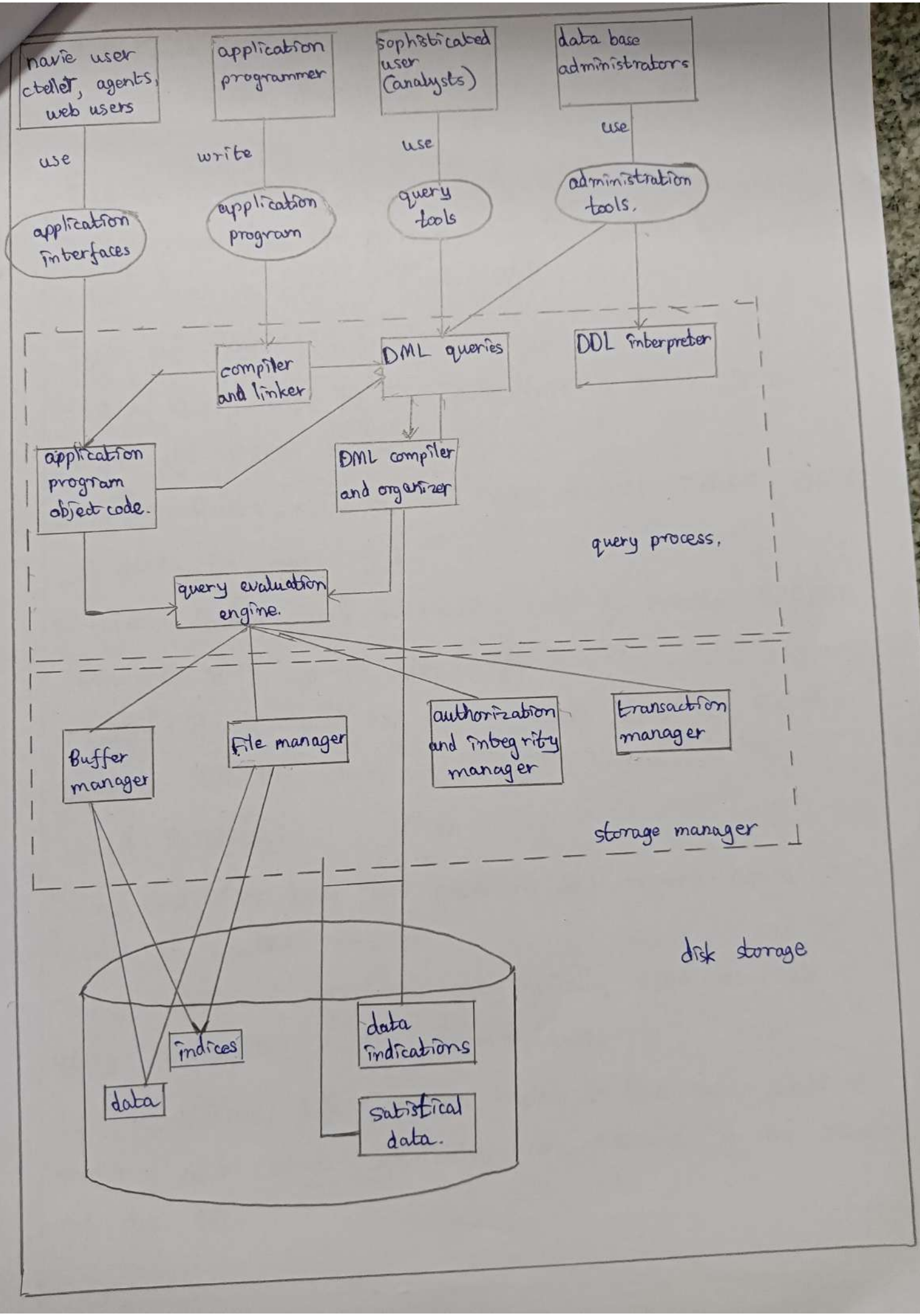
The Storage manager controls how data is stored and Retrieved

- Buffer Manager: Minizes disk I/O by storing frequently accessed data in memory
- File manager: Manages allocation of space and file structures.
- Authorization and Integrity: Ensures data security and Integrity constraints.
- Transcation Manager: Ensures consistency concurrency control and recovery.

4. Disk storage:

This is the physical layer where actual data resides

- Data: Tables and records.
- Indices: Used for fast searching
- Data Dictionary: Stores meta data (information about tables, schemes, users).
- Statistical Data: Used for query optimization.


navie user
cteller, agents,
web users
application
programmer
sophisticated
user
(analysts)
data base
administrators
use
write
use
use
application
interfaces
application
program
query
tools
administration
tools.
compiler
and linker
DML queries
DDL interpreter
application
program
object code.
DML compiler
and organizer
query process.
query evaluation
engine.
Buffer
manager
file manager
authorization
and integrity
manager
transaction
manager
storage manager
disk storage
indices
data
data
indications
statistical
data.

2. 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 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 result to another relation.

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

Join conditions - defines which tuples in the two relations match, and what attributes are present in the result of the join

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	compt sci
12121	Venkat	Finance
15151	Mozart	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 course

SELECT intructor.ID, name,

Course_id.

FROM instructor
INNER JOIN teaches
ON instructor.ID = teaches.ID.

ID	Name	Course id
10101	Srinivasan	CS-101
12121	Venkat	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;

ID	Name	Course_id
10101	Srinivasan	CS-101
12121	Venkat	Fin-201
15151	Mozart	NULL

.Right outer Join:

Returns all rows from the right table and the matching rows from the left table. If no match NULLs are shown for left table columns.

Example: Show all courses and their instructor including courses with no assigned instructor SELECT instructor. ID, name, course_id.

FROM instructor

RIGHT JOIN Reaches

ON instructor.ID = teaches.ID;

- FULL Outer Join:

Returns all rows from both tables, matching rows where available and filling NULLs where there is no match.

Example: Display all instructor and all courses even if there is no matching entry in either table.

Left join:

SELECT instructor. ID, name, course_id

FROM instructor

LEFT JOIN teaches ON instructor.ID = teaches.ID.

Union:

SELECT instructor. ID, name, course_id.

FROM instructor.

RIGHT JOIN teaches ON instructor. ID = teaches. ID.

ID	Name	Course_Id.
10101	Srinivasan	CS_101
12121	Venkat	Fin_201
15151	Mozart	NULL
76766	NULL	Bio_101

EQUI Join:

A type of INNER JOIN that uses an equality (=) operator 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	Venkat	Fin_201

Cross Join:

Returns 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	Bro_101
Venkat	CS_101
Venkat	Fin_201
Venkat	Bro_101
Mozart	CS_101
Mozart	Fin_201
Mozart	Bro_101

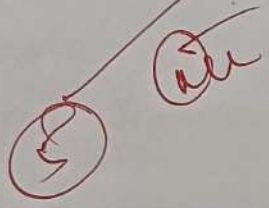
← Data Base Management System —

Assignment.

Name: Y. Tarun Kumar Reddy

VTU: 30548

Course code: 10211CA207



3) Explain about RAID storage and its types?

RAID (Redundant Array of independent disks) is a data storage technology that combines multiple physical hard drives into a single logical unit.

The main goals are.

→ Improved performance.
→ Increased storage capacity
→ Data redundancy (fault tolerance).

RAID is commonly used in servers, data centres, and storage systems to ensure data reliability and faster access.

2. Key features.

Feature	Description.
Redundancy	Data is duplicated or spread to protect against drive failure.
Performance	Multiple disks (can read) write data simultaneously.
Fault Tolerence.	System continues to function even if one drive fails.

Striping	Data is copied identically on two or more disks.
Mirroring	Data is copied identically on two or more details.
Parity	Error checking information is stored to recover data in disk failure.

Types of RAID.

RAID Level: RAID 0 (striping)

Technique used: Data divided into blocks on spread across disks.

Description: Fast performance (no redundancy).

Advantages: High speed

Disadvantages: No fault tolerance - If one disk fails, all data lost.

RAID Level: RAID 1 (mirroring)

Technique used: Same data (copied on two disks)

Description: Provides redundancy.

Advantages: High reliability, simple recovery.

Disadvantages: Storage cost doubles.

RAID level: Bit level stripping with error correction case (ECC).

Technique used: Rarely used

Description: Rarely used

Advantages: Error correction possible.

Disadvantages: Expensive, complex.

RAID Level: RAID 3

Technique used: Byte level stripping with dedicated parity disk.

Description: All drives work together.

Advantages: Good for large sequential data.

Disadvantages: Parity disk may be a bottleneck.

RAID Level: RAID 4

Technique used: Block level stripping with dedicated parity disk.

Description: Parity used for recovery

Advantages: Fast read performance.

Disadvantages: Single parity disk bottle neck.

RAID Level: RAID 5.

Technique used: Block level stripping with distributed parity

Description: Most common RAID Level.

Advantages: Good balance blw performance and redundancy

Disadvantages: Complex rebuild if a disk fails.

RAID Level: RAID 6.

Technique used: Like RAID 5 but with two parity blocks.

Description: Higher fault tolerance.

Advantages: Can bolerence two disk failure

Disadvantages: Slower write performance.

RAID Level: RAID 1

Technique used: Combination of RAID 1 & RAID 0

Description:

Advantages:

Disadvantages:

2. Deadlock and its handling?

A deadlock occurs in a database when two or more transactions are waiting for each other to release locks on resources, preventing further progress

Example:

* Transaction T1 locks resource A and waits for Resource B.
* Transaction T2 locks resource B and waits for Resource A.

→ Both wait forever → deadlock.

Conditions for Deadlock. (Coffman Conditions).

1. Mutual Exclusion:
Only one transaction can use a resource at a time.
2. Hold and wait:
A transaction holds one resource and waits for another.
3. No preemption:
Resources cannot be forcedly taken away.
4. Circular wait:
A circular chain of waiting transactions exists.

Deadlock Handling Techniques

Method	Description
1. Dead Lock Prevention.	Avoids deadlocks by denying at least on coffman condition. Example: Assign resource ordering or use time stamp based methods (wait-die, wound-wait).
2. Deadlock Avoidance.	The system checks before the ranting a lock to ensure no deadlock will occur. Example; Banker's algorithm checks safe stabes.
3. Deadlock detection.	The system allows deadlocks to occurs but periodically checks for them using a wait-for graph (WFG).
4. Deadlock Recovery	Once detected, resolve by rolling back one or more transaction victim selecti

1. Normalization and its various types of normalization.

Normalization and its various types:

Normalization is a process in database management system, DBMS used to organize data in a database to reduce redundancy (duplicate) data and improve data integrity.

1) First Normal Form (1NF).

→ Each cell must contain atomic (indivisible) values.

→ No repeating groups or arrays allowed.

Ex:

St_ID	St_Name	St_Phone	St_Group
123	Rakesh	9843298760	Physics
125	Amrita	9135462819 8763549321	stat
126	Archana	7362543210 8694753210	chemistry

New table:

St_ID	St_Name	St_Phone	St_Group
123	Rakesh	9840735 12	Physics
125	Amrita	9135462818	Stat
125	Amrita	8763549 32	Stat

Second Normal Form (2NF).

→ In the 2NF, first table must be an 1NF.

→ In the 2NF, all non key attributes are fully functionally dependent on the primary key.

→ Every non key attributes are full FHD on key attribute.

If P → A holds, then those should not be any proper subset of Q of P.

Q → A

St_ID	st_Name	Prof_ID	Prof_Name	Grade
101	ABC	2	Sameer	4
102	XYZ	3	Niranjan	6
103	PQR	1	Sushmitha	5.

St_ID → Primary key.

and no multi valued are there so satisfies 1NF.

St_ID	st_Name.
101	ABC
102	XYZ
103	PQR

Prof_ID	Prof_Name
1	Sushmitha
2	Sameer
3	Niranjan

Grades:

St_Id	Prof_ID	Grade.
101	2	4
102	3	6
103	1	5

3. Third Normal Form (3NF)

→ A relation will be in 2NF.

→ It not contains any transitive dependency.

→ The non-key attributes should not have inter dependencies among them, and the non-key attributes should fully functionally depend on key attribute. Then it is called 3 NF principle.

→ By using 3 NF to achieve data integrity and Data duplication.

Transitive dependency.

If A → B (B functionally dependent on A)

B → C

A → C (C is indirectly dependent on A)

It is called T.D.

Ex:

St_ID	St_Name	Dept_Name	Dept_Head.
101	Alice	CSE	Dr. Lalitha
102	Bob	ECE	Dr. Kumar
103	Carol	CSE	Dr. Rao.

Student table.

St_ID	St_Name	Dept_Name.
101	Alice	CSE
102	Bob	ECE
103	Carol	CSE

Dept_table.

Dept_Name	Dept_Head.
CSE	Dr. Lalitha
ECE	Dr. Kumar
CSE	Dr. Rao.

4. BCNF (Boyce-Codd Normal Form).

→ It is advanced version of 3NF.

→ It is in 3NF.

→ For every FD A→B A→Super key

→ A should be super key of a table.

Ex:

Student	Course	Teacher.
Ramesh	Physics	Kishore
Kumar	Chemistry	Ramu
Sruthi	Maths	Sanjeev
Vinay	Physics	Kishore.

keys → {student, course} →.

To eliminate redundant data we move to 2 tables

student	Course
Ramesh	Physics
Kumar	Chemistry
Sruthi	Maths
Vinay	Physics.

Course	Teacher.
Physics	Kishore.
Chemistry	Ramu.
Maths	Sanjeev

Fourth Normal Form [4NF]

→ First relation in BCNF (or) 3.NF

→ It may not contain more than one multivalued attribute.

For a dependency.

$A \rightarrow B$.

→ If for a single value of 'A' multiple values of B exists, then the relation will be multi-valued dependency.

St-ID	Course	Hobbies.
111	Maths	Dancing
111	Computer	Singing
222	Chemistry	Dancing
444	Sanskrit	Cricket
555	Physics	Hockey.

St-ID	Course
111	Maths
111	Computer
222	Chemistry
444	Sanskrit
555	physics

St-ID	Hobbies.
111	Dancing
111	Singing
222	Dancing
444	Cricket
555	Hockey.