Assignment - I

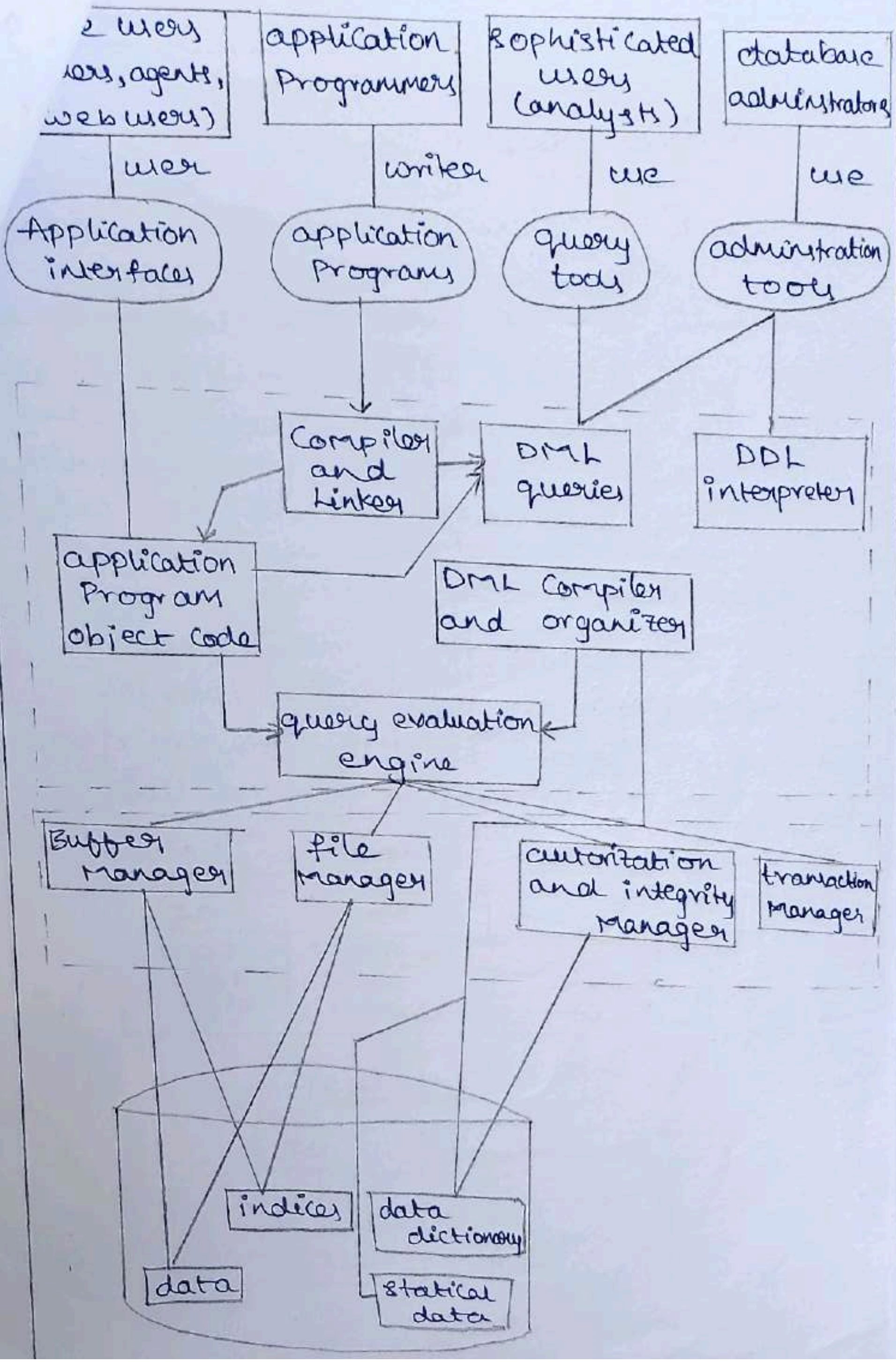
Name : G. Bhargav Ram

VTU : 30538

Subject : DBMS

Section: ~~B~~CSE (AI-ML)

5


e users
ers, agents,
web users)
application Programmers
sophisticated users (analysts)
database administrators
user
writer
use
use
Application interfaces
application Programs
query tools
administration tools
Compiler and Linker
DML queries
DDL interpreter
application Program object code
DML Compiler and organizer
query evaluation engine
Buffer manager
file manager
autorization and integrity manager
transaction manager
indices
data dictionary
data
statical data

ned Relations

Join operations take two relations nd returns as a result another relation.

There additional operations are typically used as subquery expressions in the form Clause.

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

Join Types
inner Join
left outer join
right outer join
full outer join

Join Conditions
Natural
On <predicates>
using (A1, A2 -- An)

Database Example - Join

ID	Name	dept_name
10101	Srinvasan	Comp.sci
12121	Wu	finace
15151	Mozart	Music

Teachers

ID	Course_id
10101	CS - 101
12121	fin - 201
76766	B10 - 101

: JOIN

an the Cartesian product of two les - every row from the first table ined with every row from second

yntax:

```
SELECT Columns
FROM table 1
CROSS JOIN table2;
```

Ex:

```
SELECT name, Course _id
FROM instructor
CROSS JOIN teacher;
```

Name	Course_id
srinivasan	CS-101
Crinivasan	FIN-201
srinivasan	BIO-101
Wu	CS-101
Wu	FIN-201
Wu	BIO-101
Mozart	CS-101
Mozart	FIN-101
Mozart	BIO-101

Natural JOIN: A type of join that automatically join tables using Column with the same name and Compatible data

Syntax:

```
SELECT
FROM table1
NATURAL JOIN table2;
```

Ex:

```
SELECT *
FROM instructor
NATURAL JOIN teacher;
```

ID	Name	dept	Name	Course_id
10101	srinivasan	Comp-sci		CS-101
12121	Wu	finance		FIN-201

Assignment-3

Name :- G. Bhargav RAM

VTU :- 30538

sub :- DBMS

3

Join

n only the matching rows from both es based on a given Condition.

sic Syntax:

```
.ELECT Columns
FROM table1
INNER JOIN table2
ON table1. Common
FROM table 1
INNER JOIN table2
ON table 1. Common- Column =table2.
 Common _ Column;
```

Example: Display instructors who are teaching at least one Course

```
SELECT instructor. ID, name, Course_i
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.

Basic syntax:

```
SELECT Columns
FROM table 1
LEFT JOIN table2
```

```
able 1. Common - column = table2.
mmon - Column;
```

Example: ~~shows~~

```
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	Wu	FIN-201
76766	NULL	BIO-101

full Outer Join

Returns all rows from both tables, Matching rows where available, and filling NULLs.

Basic syntax:

```
SELECT Column
FROM table 1
~~FULL table~~
FULL OUTER JOIN table 2
 ON table 1. Common_column =
  table2. Common - Column;
```

Example: Display all instructors and all Courses, even if there's no matching entry in either table.

Left join:

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

```
CT instructor. ID, name, course_id
om instructor
GHT JOIN teaches ON instructor.ID
    = teaches.ID;
```

ID	Name	Course _id
10101.	Srinivasan	CS-101
12121	Wu	FN-201
15151	Mozart	NULL
76766	NULL	BIO-101

Equi JOIN: A type of INNER JOIN that uses an equality (=) operator to match rows.

Basic syntax:

```
SELECT Column
FROM table 1, table 2
WHERE table 1.
Common _column = table2. Common _column;
```

Example:

```
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

Assignment - 5

Namer G. Bhargav RAm
VTUr 30538
Subr DBMs

5

n about RAID storage
its types:

an acronym for Redundant
y of indepent Disks) is a
- storage virtualization
nology that Combines
tiple physical hard disk
ves (HDDs) or solid-state
ves (SSDs) into a single
cal unit.

- primary goals of using RAID
e to increase data redundanc
st performance, or a Combina
n of both Core RAID
echniques.

here are three techinques.

. striping (RAID 0): Data is split into blocks and written sequentially across multiple disks. This significantly increases performance but offers no fault tolerance.

+. Mirroring (RAID 1): Data is simultaneously and identically written to two or more disks. This provides excellent fault tolerance but halves the effective storage Capacity.

plain about Deadlock and Its handling:

A deadlock in an operating System is a situation where a set of processes are permanently blocked because each process is waiting for a resource that is being held by another process in the set, leading to a circular chain of dependencies. It's essentially a standstill that halts the progress of the involved process.

*. **Mutual exclusive**: At least one resource must be non-sharable, meaning only one Process Can use it at a time.

*. **Hold and wait**: A process is Currenty holding at least one resource and is waiting to acquire additional resources. held by other processes.

*. **No preemption**: A Resource Cannot be forcibly taken from the process holding it; it Can only be realeased voluntarily by that process.

cular wait: A set of ocess (P_0, P_1, P_2, \dots, P_n) ists such that P_0 is waiting for a resource held by P_1, P_1 is waiting for a resource hold by P_2, --, and P_n is waiteing for a resource held by P_0.

Dead lock handing strategies: operating systems use three main strategies to deal with dead locks:

1. Dead lock Prevention:
This strategy involves ensuring that at least one of the four necessary Conditions can never hold. hold and wait.

2. Dead lock Avoidance:
This approaches requires prior knowledge of resource needs. The OS then grants a resource request only if the resulting system.

3. Deadlock Detection & Recovery:
If prevention or avoidance is not used, deadlocks can occur.

4. The ostrich Approach: Many non-critical operating systems choose to ignore the problem.

ormalization and its various types:

Normalization is the process of organizing the data in a data base, primarily a relational data base, to reduce data redudency and improve data integrity.

This is achieved by breaking down a large table into smaller, simpler, and related tables, which follow a set of rules called normal forms are 1NF, 2NF, 3NF and BCNF.

Types of Normalization (Normal forms) Normalization is a Progressive process, meaning a higher Normal form Cannot be achieved unless the precending normal form has been satisfied.

1. first Normal form (1NF):

A table is in 1NF if:

Atomic values: every Column Contains atomic values. There are no multi-valued attributes in a single cell no Repeating group. There are no repeating groups of Columns.

ond Normal form ... it is ...

table is in 2NF if it is ... No partial dependent on the entir primary key. This is rule is significant only when the primary key is a composite key.

3. Third Normal form (3NF)

A table is in 3NF if it is in 2NF. No transitive dependency of a non-key attribute on the primary key. This means a non-key attribute cannot depend on another non-key attribute.

4. Boyce - Codd Normal form (BCNF)

BCNF is considered a stricter version of 3NF. for every non-trivial functional dependency, x must be superkey (a set of attributes that uniquely identifies a tuple).

5. fourth Normal form (4NF):

Deals with multi valued dependencies. the table must be in BCNF.

6. fifth Normal form (5NF):

Deals with join dependencies and aims to decompose a table into the smallest possible tables without loss of information. The Table must be in 4NF.

tabase system Architecture

The database system architectur can be divided into three main part users. Query processor, and storage Manager, with the disk storage at the bottom.

1. Users:

Different types of users interact with the database:

- Naive users (tellers, agents, web user → use application interfaces.
- Application programmers → write application programs.
- sophisticated users → use query tools.
- Database Administrates (DBAs) → use administration tools.

2. Query processor:

This is responsible for interpreting and executing queries.

- Application program object Code: Generated by Compiler and linker.
- DML Queries: Data Manipulation language (eg: select, Insert, update).
- DDL interpreter: Interprets schema definitions (tables, constraints).
- DML Compiler and organizer: Chec syntax and optimizes queries.
- Query Evaluation Engine: Execute optimized queries.

Storage Manager:

The storage Manager Controls how data is stored and retrieved.

- Buffer manager: minimizes disk '1/o by storing frequently accessed data in memory.
- file manager: mangers allocation of space and file structures.
- Authorization and integrity manager: Ensures data security and integrity constraints.
- Transaction 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, schemas, users).
- statistical Data: used for query optimization.