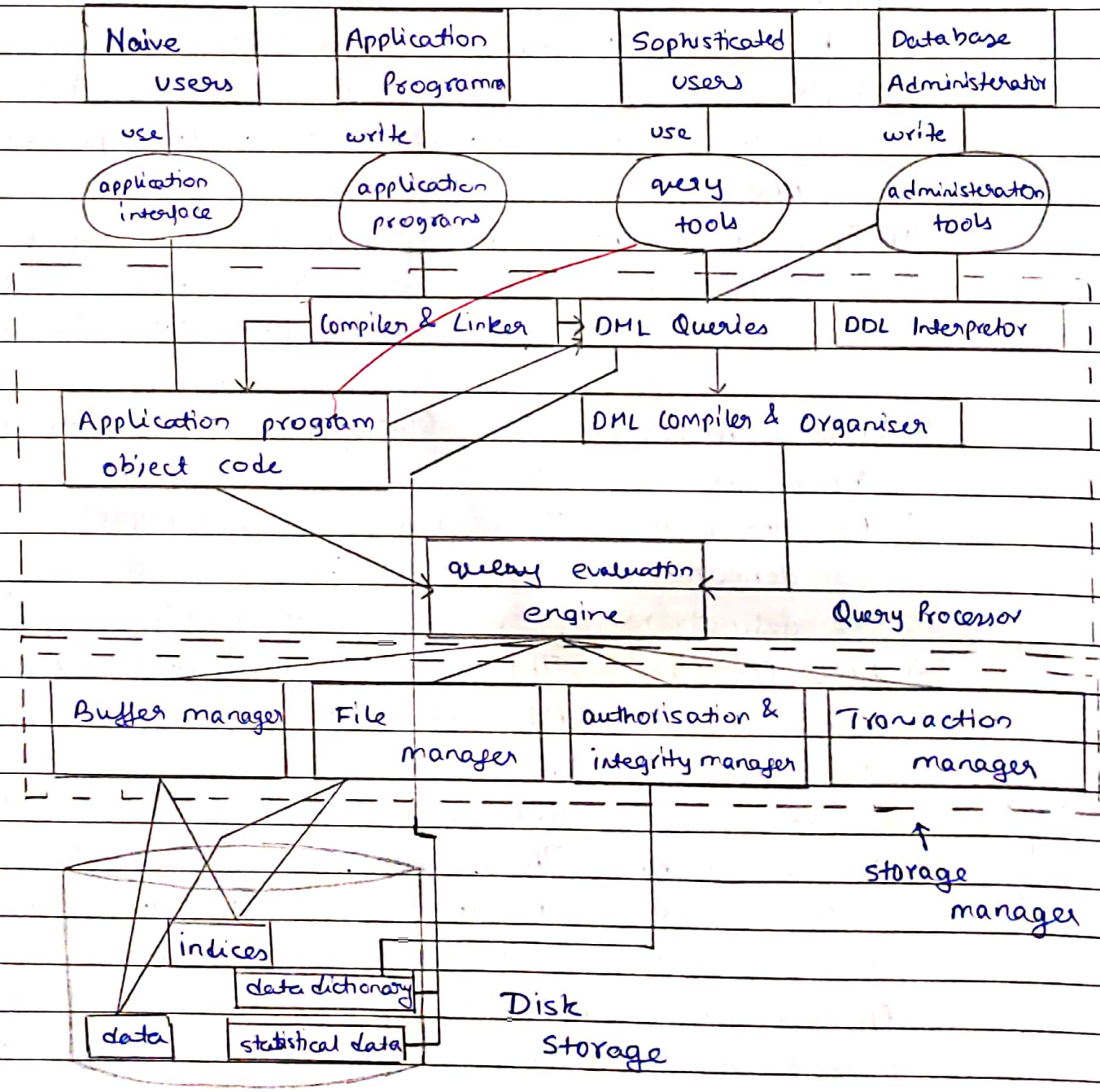
Herbal 23/1/22 (24)

Term Test 1

Shasvat Shah
60004220126

DBMS Assignment

1 DBMS system architecture ..



→ Database Users: There are multiple types of database users that use the DBMS.

1) Naive Users - These are basically unsophisticated users who interact with the system invoking one of the application program. Basically, they use the system only when needed.

2) Application Programmers - These are basically the professionals who write the application programs.

3) Sophisticated Users - These are users who interact with the system without writing the programs, eg. an Analyst.

4) Specialized Users - A specialized user can be a DBA or a database administerator who are sophisticated users who write specialized program applications.

→ Query Processor: The Query processor includes multiple things which include.

1) DDL Interpreter - This interpretor interprets all the data defination language statements and records all the definations in the data dictionary.

2) DML Compiler - This compiler is responsible to translate the DML statements in a query language.

3) Query Evaluation Engine - This engine is responsible to execute the low level instructions generated by DML compiler.

4) Compiler & Linker - The compiler is responsible to compile the instructions given by the user and the linker, links these instructions with the standard libraries.

5) Application Program Object Code: Whenever a source code is executed, an object code of the application program is created

first, and after that we can view the executed application program.

→ Storage Manager : The storage manager as the name suggests help with the process of storing.

1) Authorisation & Integrity Manager - Checks for the authority of the user to access data, it also tests for the satisfaction of integrity constraints.
2) Transaction manager - It manages that the concurrent transactions take place without errors, conflicts and failure
3) File Manager - It helps manage the space allocation on the disk and manages the datastructure used to represent the information present on the disk.
4) Buffer Manager - Mainly responsible to fetch data from disk storage to the main memory. It also decides which data to cache in main memory.

→ Disk Storage : This is where the data of the database is present.

1) Data files - The actual database is stored in the data files.
2) Data Dictionary - It basically stores the meta data. The meta data is basically the structure of the database.
3) Indices - These help in providing fast access to data items.
4) Statistical data - It basically stores the statistical information about the data in the database.

2 Types of Database Users

The types of database users are defined on the basis of their interaction with the database. The users are

1) Database Administerator (DBA)
   → Defines the schema and controls 3 levels of the database
   → Creates ID & password for the new user.
   → provides security to the database
   → Authorises user access in the database.

2) Naive Users
   → Unsophisticated users with no DBMS knowledge.
   → Use the database just for desired results.
   → For eg Railway tickets booking.

3) System Analyst
   → Analyses the requirements of the end user.
   → Keeps a check if all requirements have been satisfied.

4) Sophisticated User
   → These are generally people familiar with the database
   → Capable of developing their own database and run queries in the database.
   → Eg. Engineers, Scientists, business analysts, etc.

5) Database Designers
   → Designs the structure of the database like tables, views, indexes, triggers, constraints, etc.
   → Understands the requirements of the user group and then creates the design accordingly.

6) Application Programmers
   → Also called system analysts or the back end programmers
   → These write the code for application programmers.

7) Casual Users -
   → Use the database occasionaly each time for something different

Features of the EER Diagram { Extend Entity Relationship }

There are multiple features of the EER diagram, Some of those listed below are :-

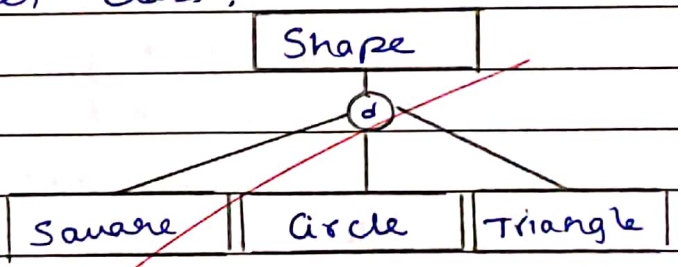
1) Enhanced Entity - Relationships are basically advanced diagrams, which also shows the requirements and complexities of the database. It is basically a diagrammatic representation for displaying.

1) Sub class / Super class

→ An entity can have an additional meaningful subgrouping of its entities that need to be represented explicitely due to their significance

→ The subgroupings are called subclasses of the entity which is called the super class.

→ In the below example, we can se shape is the superclass. The types of shape make the sub-class

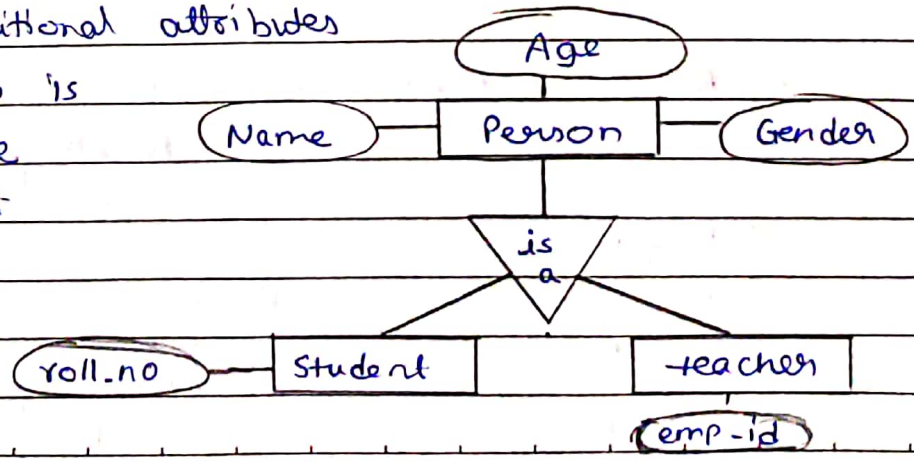


2) Specialization

→ Classification of an entity to a ~~specified~~ specialized entity.

→ It converts the High level entities to low level entities by adding additional attributes

→ Here, entity person is classified to more specialized student and teacher entity.

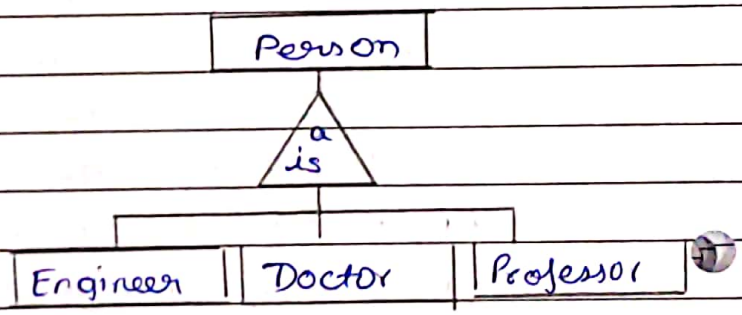


3) Generalisation

→ Opposite of specialization which uses a bottom up approach.

→ Common attributes of more than one entity form a new entity.

→ Lower level entities like doctor, engineer, professor combine to form a higher level attribute.

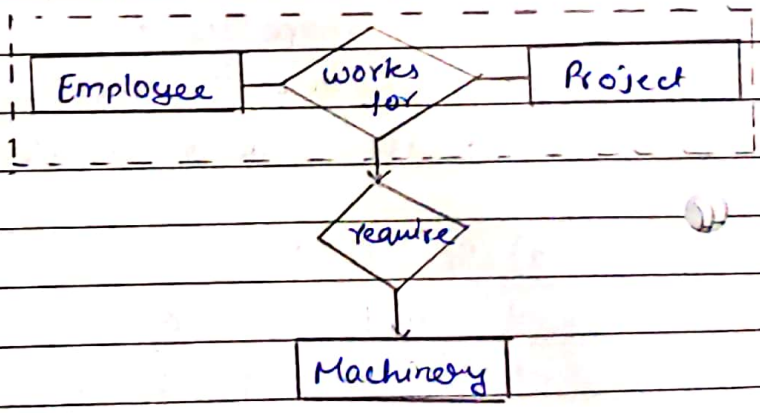


4) Aggregation

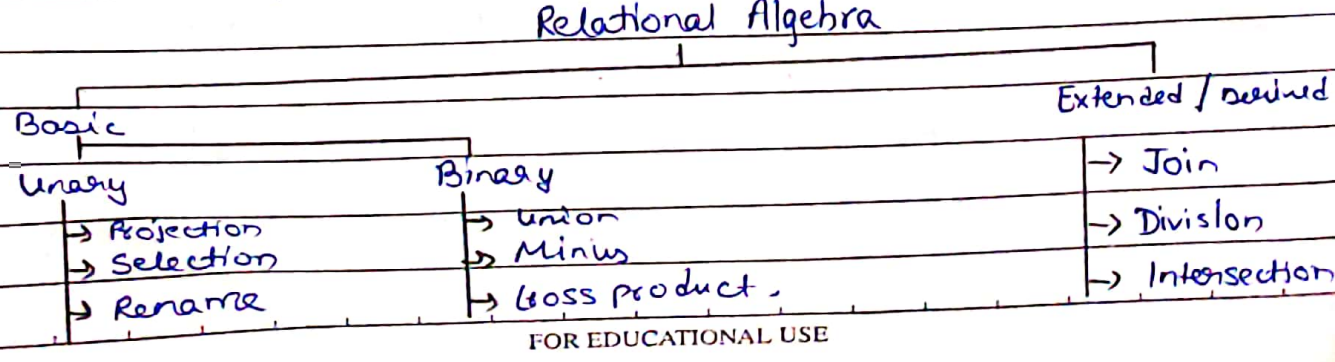
→ A relationship where two entities are treated as a single entity.

→ An abstraction through which relationships are treated as higher level entities.

→ Here we can see employee and project is treated as a single entity.



4 Relational Algebra Operators.



1) Projection (π): Projects required column data from the relation. It uses symbol (π) to choose attributes from a relation. Eg. Student. π Name, Marks (student)

Name	Age	Marks
Joe	19	70
John	18	80
Jack	15	90

Name	Marks
Joe	70
John	80
Jack	90

Syntax : π <attribute list> (relation)

2) Selection (σ): It uses symbol (σ) sigma which is used as an expression to choose tuples that meet the selection ~~conditions~~. condition.

Syntax : σ Salary >10000 (staff)
σ <selection condition> (relation)

Eg. σ Salary > 10000 (staff)

∴ Result that we achieve includes:-

EmpID	Emp_name	Salary
1	Farhan	100000
4	Shasvat	20000

3) Rename (ρ): A unary operation used for renaming the attributes (columns) of a relation.

Syntax : ρ newname | oldname (relation)

Eg. ρ (ShashwatShah | ShasvatShah (Shashwat)

∴ ShasvatShah column will be changed to name ShashwatShah in the table Shashwat.

Union (U) : This is basically the same as the uni operation in set theory, though here the constraint is tha both the relations must have the same attributes, I we consider a union of two relations.

Syntax : Relation 1 U Relation 2

Eg.

A

Rollno	Subject
1	C
2	C++
3	Java

B

Rollno	Subject
5	Python
6	Kotlin.
1	C

→ A U B

Rollno	Subject
1	C
2	C++
3	Java
5	Python
6	Kotlin.

5 Cross Product (X) : It is basically the same as cartesian products. 'X' is the symbol.

Syntax : Relation A X Relation B.

Eg.

Student

Sid	Name
1	Sharvat
2	Shah

Course.

Cid	Cname
1	C++
2	Java

→ Student X Course.

Sid	Name	Cid	Cname
1	Sharvat	1	C++
2	Shah	1	C++
1	Sharvat	2	Java
2	Shah	2	Java

6) Join - Used to join two types of relations. Multiple types of joins are available.

→ Natural Join → Equi Join

→ Outer Join

→ Left outer join

→ Right outer join

→ Full outer join.

Syntax : Relation A ⋈ Relation B

Eg.

Employee		Salary	
Emp_code	Emp-name	Emp-code	Salary
101	Stephan	101	50000
102	Jack	102	30000

Employee ⋈ Salary

Emp_code	Emp_name	Salary
101	Stephan	50000
102	Jack	30000

7) Intersection - It is basically the same as that of set theory, though the constraint is that, the column attributes must be same for both the columns.

Syntax : Relation A ∩ Relation B

∴ Referring previous table of unions.

Eg.

A ∩ B

A	B
C-10	C-Subject
01	C++

5 Integrity Constraints; These are mainly of 3 types.

- Domain → Not Null & Check
- Entity → Unique & Primary key
- Referential → Foreign Key

Explanation of these above constraints :-

1) Notnull - It basically avoids the absense of values in a row. for a given attribute.
2) Check - It basically defines the condition that must be staisfied by each row.
3) Unique - Helps prevent duplications of values within rows of the same column.
4) Primary Key - Helps identify records uniquely. The primary key, is a combination of unique and not null constraints.
5) Foreign key - It is basically a field in one tables that refers to the primary key of the other ~~tho~~ table.

6 Aggregate functions in SQL

These aggregate functions are used to perform the calculations on multiple rows of a single column of a table. It returns a single value. Multiple egs.

1) Count (column_name)
   Counts the number of values in a column.
   eg. select count (name) from Emp;
   select count (*) from Emp;
   → returns all rows from Emp, only count.
2) MIN (column_name)
   Returns the lowest value for a column.
   eg. select Min (salary) from EMP.

3) MAX (column_name)

It returns the highest value from a column.

eg. Select max (salary) from emp.

4) Sum (column_name)

Helps calculate the sum of all values of the column

eg. select sum (salary) from emp.

5) Avg (column_name)

Helps calculate the average of values in a column

Eg. Select avg (salary) from emp.

table

Product	Company	Price	Rate
P_1	Emp 1	20	10
P_2	Emp 2	20	20
P_3	Emp 1	30	10
P_4	Emp 3	150	30
P_5	Emp 4	20	20

→ select count * from table ;

→ 5

→ select avg (price) from table;

→ 48.00

→ select sum (price) from table ;

→ 240

→ select min(price) from table;

→ 20

→ select max (price) from table;

→ 150

Integrity constraints

Example

Syntax:

```
Create table  table_name (
        column 1   datatype   constraint,
        column 2   datatype   constraint,
        column 3   datatype   constraint,
        ....
    );
```

example

```
Create  table  employee(
    emp_ID   int   primary key,
    email   varchar(20)   unique, not null,
    Foreign key (emp-code) References Employee detal
                                    (emp-code)
    );
```

example

```
Create table  persons (
    Age int,
    check (age >= 18)
    city varchar(20)  default 'Mumbai'
    );
```