Unit-1
Introduction to Database Management System

Ms. Sonali Gahlot
Computer Science & Engineering
Ajay Kumar Garg Engineering College, Ghaziabad
sonaligahlot5@gmail.com
9258300179

Outline

- Introduction of DBMS
- Applications of DBMS
- Advantages of DBMS
- File System vs DBMS
- Basic Terms
- Three levels Database system
- Data Abstraction in DBMS
- Mappings and data independence
- Database users and DBA
- Database system architecture
- Keys

Introduction to DBMS

Section - 1

What is Database Management System (DBMS)?

- Data - **Fact** that can be recorded or stored
  - e.g. Person Name, Age, Gender and Weight etc.
- Database - Collection of **logically related data**
  - e.g. Books Database in Library, Student Database in University etc.
- Management - Manipulation, Searching and Security of data
  - e.g. Viewing result in AKTU website, Searching exam papers in AKTU website etc.
- System - **Programs** or **tools** used to manage database
  - e.g. SQL Server Studio Express, Oracle etc.
- DBMS - A Database Management System is a software for creating and managing and securing databases.
- Database Management System (DBMS) is a **software designed to define, construct, manipulate, retrieve , share and secure the data in a database**.
  - e.g. MS SQL Server, Oracle, My SQL, SQLite, MongoDB etc.

Applications of DBMS

Section - 2

Applications of DBMS

- DBMS is a computerized record-keeping system.
- DBMS is required where ever data need to be stored.
  - E-Commerce (**Flikart, Amazon, Shopclues, eBay** etc...)
  - Online Television Streaming (**Hotstar, Amazon Prime** etc...)
  - Social Media (**WhatsApp, Facebook, Twitter, LinkedIn** etc...)
  - Banking & Insurance
  - Airline & Railway
  - Universities and Colleges/Schools
  - Library Management System
  - Human Resource Department
  - Hospitals and Medical Stores
  - Government Organizations

Exercise Write down any five applications of DBMS other than above.

Advantages of DBMS

Section - 3

Reduce data redundancy (Duplication)

Computer

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Civil

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Database management system can remove such data redundancy by storing data centrally.

Same data is stored at four different places.

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Electrical

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Mechanical

Remove data inconsistency

Computer

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	6789	DBMS

Civil

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Database management system can keep data in consistent state.

Same data having different state (values)

Mobile no is changed

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	6789	DBMS

Electrical

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Mechanical

Data isolation

- Data are **scattered** in various files.
- Files may be in **different formats**.
- **Difficult to retrieve** the appropriate data.

DBMS allow us to access (retrieve) appropriate data easily.

Data isolation is a property that determines when and how changes made by one operation become visible to other concurrent users and systems.
This issue occurs in a concurrency situation.

File - 1

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

File - 2

Emp_Name	Post	Salary	Load
Anuj Kumar	AP	1 Lakh	16

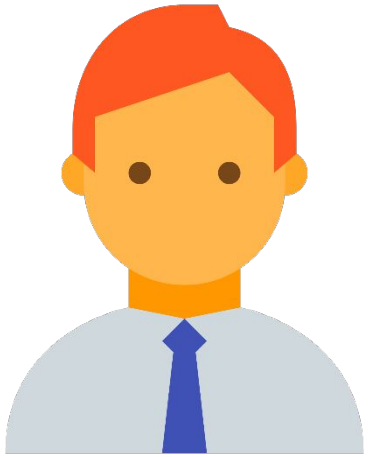
File - 3

Emp_Name	Teaching	Knowledge	Rating
Anuj Kumar	Good	Excellent	9

Guaranteed atomicity

- Atomicity: Either transaction **execute 0% or 100%**.

Sum of both account before transfer is 3000



Person A
Account A
Bal : 2000

Transfer 500

Person B
Account B
Bal : 1000

Step 1 : Debit 500 from Account A
Step 2 : Credit 500 into Account B

Transaction is failed

Sum of both account after transfer is 3000

Sum of both account is 2500 so inconsistent

Allow to implement integrity constraints

Emp_Name	Address	Mobile_No	Subject
Anuj Kumar	Delhi	9557131833	DBMS

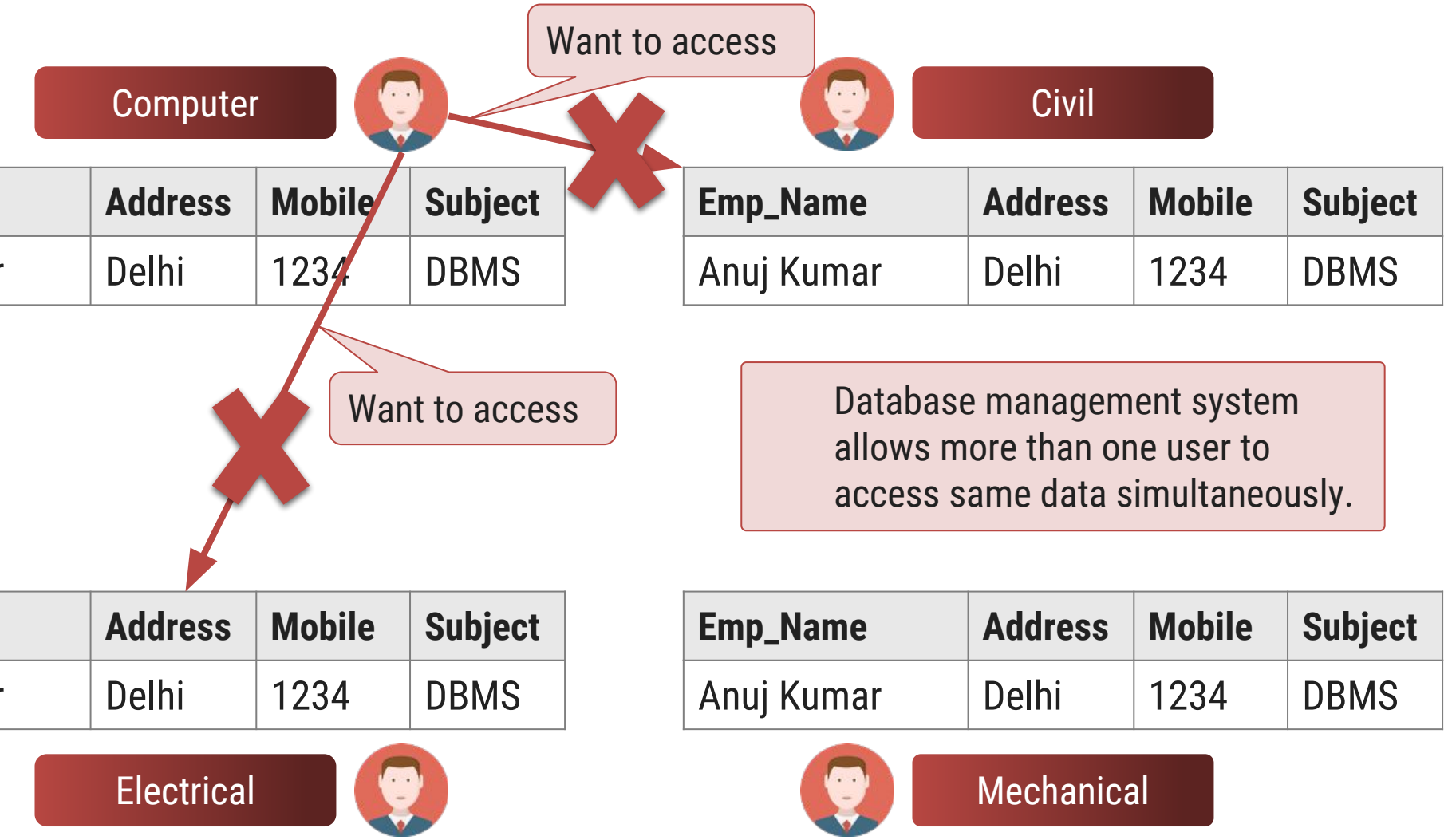
Should contain exact 10 digits

Student_Name	Branch	Backlog	CGPI
Rajat Sharma	Delhi	0	8.5

Should be between 0 to 10

DBMS allows us to implement such business rules in our database..

Sharing of data among multiple users



Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	DBMS

Restricting unauthorized access to data

File - 1

Emp_Name	Address	Mobile	Subject
Anuj Kumar	Delhi	1234	PPS

File - 2

Emp_Name	Post	Salary	Load
Anuj Kumar	AP	1 Lakh	15

File - 3

Emp_Name	Teaching	Knowledge	Rating
Anuj Kumar	Good	Excellent	9

Wants to access

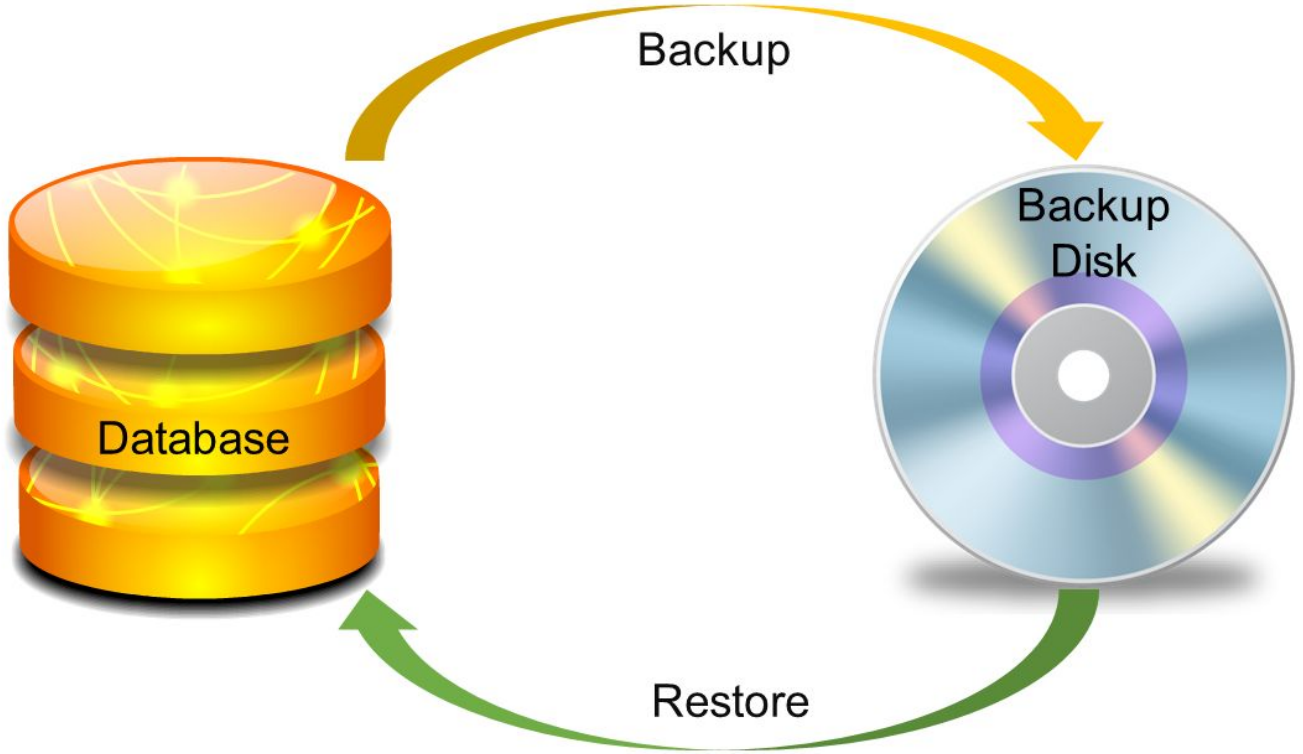
Faculty of other college

Wants to access

AKGEC Faculty

DBMS prevents unauthorized user to access data.

Providing backup and recovery services



Provides facilities to backup and restore the database in case of failure.

Advantages of DBMS (Summary)

- Reduce data redundancy (duplication)
  - **Avoids unnecessary duplication** of data by storing data centrally.
- Remove data inconsistency
  - By **eliminating redundancy**, data **inconsistency can be removed**.
- Data isolation
  - A user can **easily retrieve proper data** as per his/her requirement.
- Guaranteed atomicity
  - Either transaction **executes 0% or 100%.**

Advantages of DBMS (Summary)

- Allow implementing integrity constraints
  - **Business rules can be implemented** such as do not allow to store amount less than Rs. 0 in balance.
- Sharing of data among multiple users
  - **More than one users can access** same data at the same time.
- Restricting unauthorized access to data
  - A user can **only access data which is authorized** to him/her.
- Providing backup and recovery services
  - Can **take a regular auto or manual backup** and **use it to restore** the database if it corrupts.

Basic Terms

Section - 4

Basic terms

- Data
  - Data is **raw, unorganized facts** that need to be processed.
  - Example: Marks of students
  - Student_1 = 50/100, Student_2 = 25/100.
- Information
  - When data is **processed, organized, structured** or presented in a given context so as to make it useful, it is called information.
  - Example: Result of students (Pass or Fail)
  - Student_1 = Pass, Student_2 = Fail.

Basic terms (cont…)

- Metadata
  - Metadata is **data about data**.
  - Data such as table name, column name, data type, authorized user and user access privileges for any table is called metadata for that table.

Faculty

Emp_Name	Address	Mobile_No	Subject
Anuj Kumar	Delhi	9557131833	DBMS

  - Metadata of above table is:
    - Table name such as Faculty
    - Column name such as Emp_Name, Address, Mobile_No, Subject
    - Datatype such as Varchar, Decimal
    - Access privileges such as Read, Write (Update)

Basic terms (cont…)

- Data dictionary
  - A data dictionary is an information repository which **contains metadata**.
    - Table Name – Faculty
    - Column Name – EmpName, Address, Mob, Subject, Salary
    - Datatype – Varchar, Decimal
    - Access Privileges – Read, Write (Update)

- Data warehouse
  - A data warehouse is an information repository which **stores database**.

Faculty

Emp_Name	Address	Mobile_No	Subject
Anuj Kumar	Delhi	9557131833	DBMS
Sandeep Kumar	Gzd	9876543210	DAA

Exercise Why data dictionary and data warehouse are stored in the different places?

Basic terms (cont…)

- Field
  - A field is a **character or group of characters** that have a specific meaning.
  - E.g, the value of Emp_Name, Address, Mobile_No etc are all fields of Faculty table.

Faculty

Emp_Name	Address	Mobile_No	Subject
Anuj Kumar	Delhi	9557131833	DBMS
Sandeep Kumar	Gzd	9876543210	DAA

- Record / Tuple
  - A record is a **collection of logically related fields**.
  - E.g, the collection of fields (Emp_Name, Address, Mobile_No, Subject) forms a record for the Faculty.

Anuj Kumar	Delhi	9557131833	DBMS
Sandeep Kumar	Gzd	9876543210	DAA

Record / Tuple

Basic terms (cont..)

- Schema : A schema is a **description** of a particular collection of data, using a given data model.
- Schema is the **overall design of database**.
- A database schema is the **skeleton structure** that represents the logical view of the entire database.
- A database schema can be divided broadly into two categories –
  - **Physical Database Schema** – This schema pertains to the **actual storage of data** and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
  - **Logical Database Schema** – This schema defines all the **logical constraints** that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Basic terms (cont..)

- Instance- It is a state of **operational database** with data at **any given time**.
- It contains a snapshot of the database.
- Database instances tend to change with time
  - Example- Entry of data in any table at particular time.

File System vs DBMS

Section - 5

File System vs DBMS

Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.

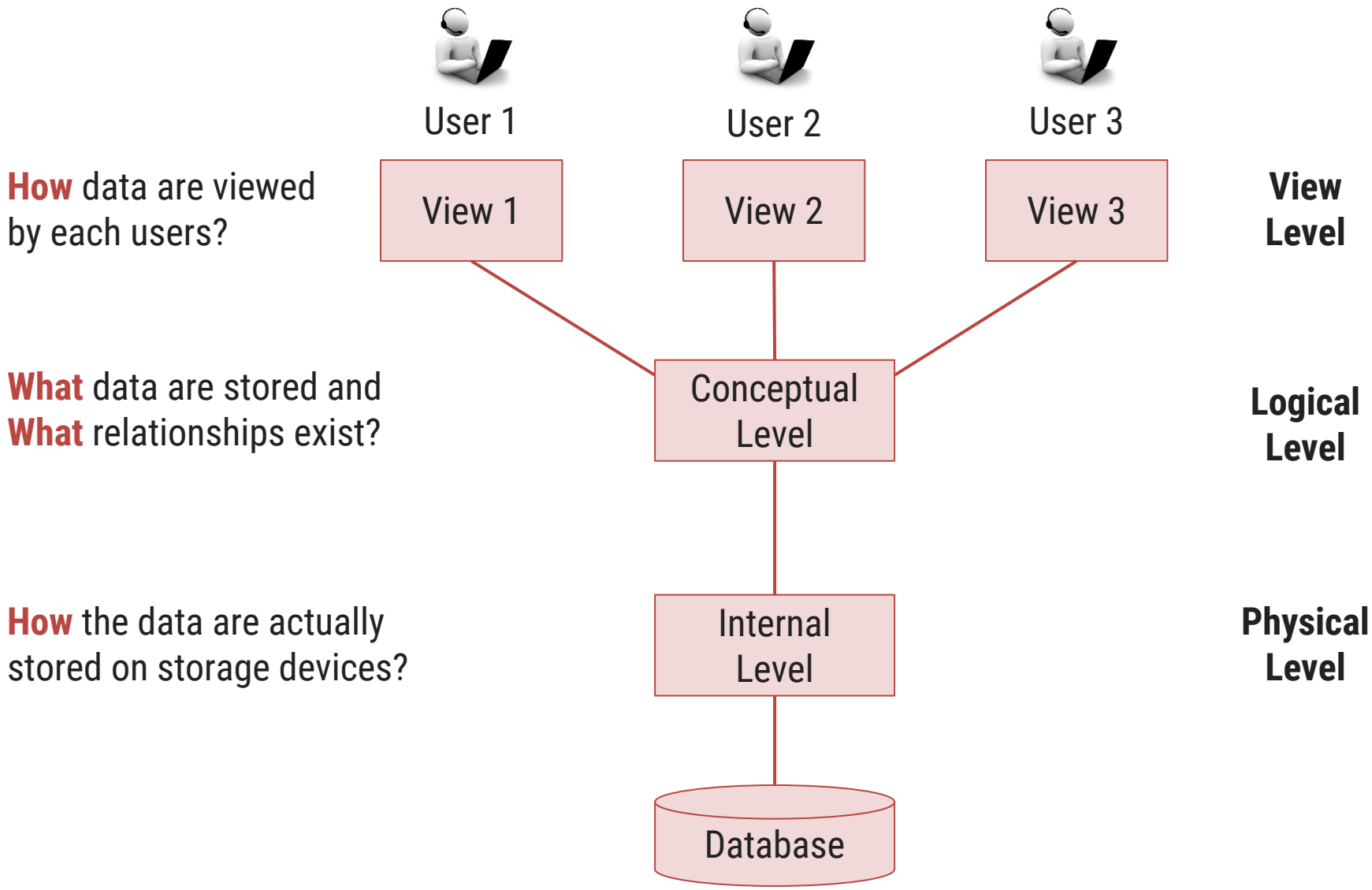
File System vs DBMS (Cont..)

Basis	File System	DBMS
Data Independence	There is no data independence.	In DBMS data independence exists.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The user has to write procedures for managing databases	The user not required to write procedures.
Sharing	Data is distributed in many files. So, not easy to share data	Due to centralized nature sharing is easy
Data Abstraction	It give details of storage and representation of data	It hides the internal details of Database
Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement
Example	NTFS, Cobol, C++	Oracle, SQL Server,MySql

3 Levels ANSI Database System

Section - 6

3 Levels ANSI Database System



3 Levels ANSI SPARC Database System

- Internal level (Physical level)
  - It describes **how a data is stored** on the storage device.
  - Deals with physical storage of data.
    - Structure of records on disk - files, pages, blocks and indexes and ordering of records
  - Internal view is described by the internal schema.
- Conceptual level (Logical level)
  - **What data are stored and what relationships exist** among those data?
  - It hides low level complexities of physical storage.
  - For Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware about their storage.
  - Database administrator works at this level to determine what data to keep in the database.
- External level (View level)
  - It describes only part of the entire database that an end user concern or **how data are viewed** by each user.
  - Different user needs different views of the database, so there can be many views in a view level abstraction of the database. Used by end users and application programmers.
  - End users need to access only part of the database rather than the entire database.

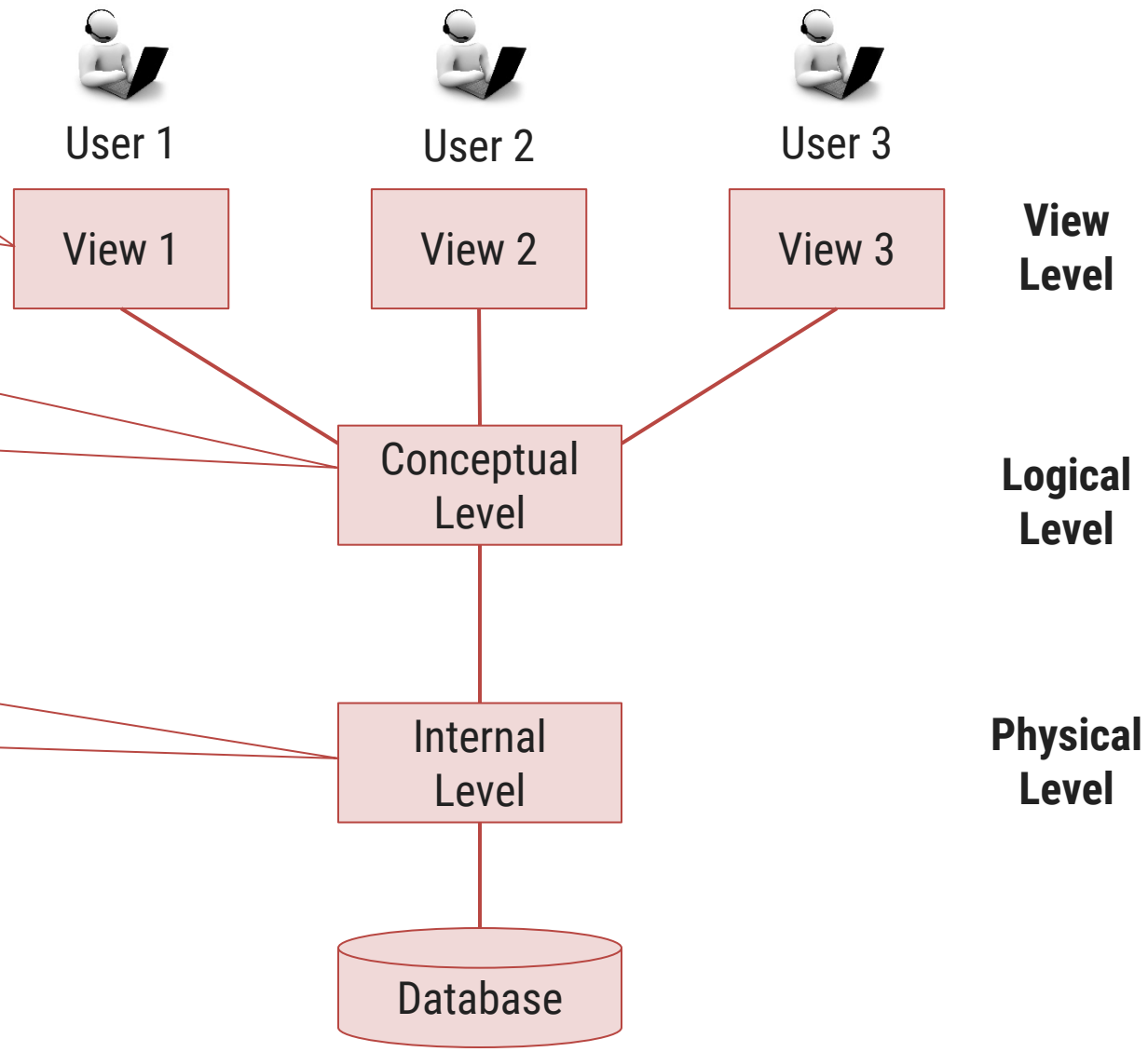
3 Levels ANSI SPARC Database System: Example

- We are storing student information in a student table

User just interact with system with the help of GUI.
Users are not aware of how and what the data is stored.

Records can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented.
Programmers generally work at this level.

Records can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory.
These details are often hidden from the programmers.



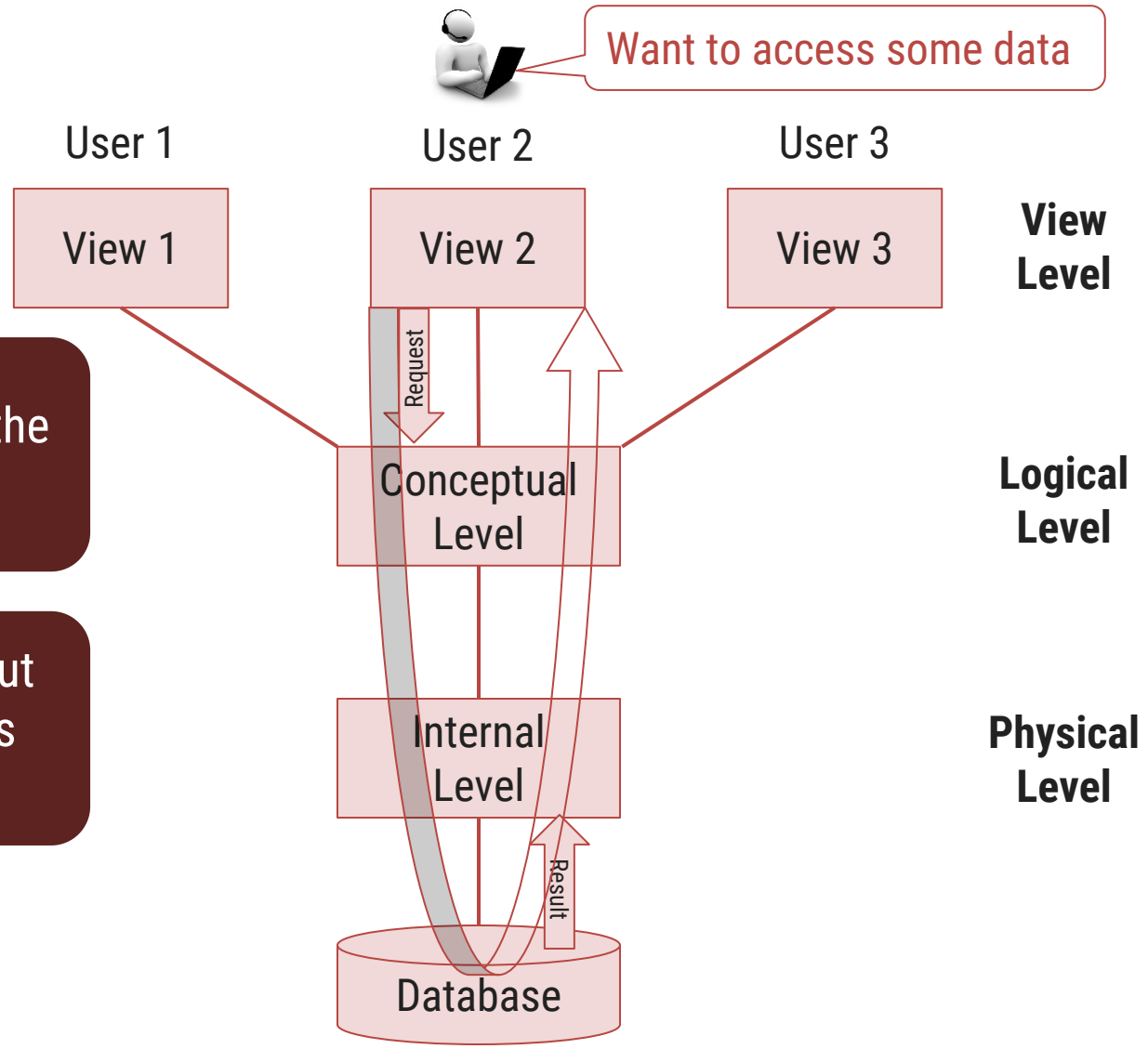
Data Abstraction in DBMS

- Database systems are made-up of complex data structures.
- To ease the user interaction with database, the developers hide internal irrelevant details from users.
- This **process of hiding irrelevant details** from user is called data abstraction.

Mapping and Data Independence

Process of transforming requests and results between the three levels is called mapping.

Ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called Data Independence



Types of Data Independence

- Physical Data Independence
  - Physical Data Independence is the ability to modify the physical schema without requiring any change in logical (conceptual) schema and application programs.
  - Modifications at the internal levels are occasionally necessary to improve performance.
  - Possible modifications at internal levels are changes in file structures, compression techniques, hashing algorithms, storage devices, etc.
- Logical Data Independence
  - Logical data independence is the ability to modify the conceptual schema without requiring any change in application programs.
  - Modification at the logical levels is necessary whenever the logical structure of the database is changed.
  - Application programs are heavily dependent on logical structures of the data they access. So any change in logical structure also requires programs to change.

Types of Database Users

Section - 7

Types of Database Users

- Naive Users (End Users)
  - **Unsophisticated users** who have zero knowledge of database system
  - End user interacts to database via sophisticated software or tools
  - e.g. Clerk in bank
- Application Programmers
  - **Programmers** who write software using tools such as Java, .Net, PHP etc…
  - e.g. Software developers
- Sophisticated Users
  - **Interact with database system** without using an application program
  - Use query tools like SQL
  - e.g. Analyst
- Specialized Users (DBA)
  - User **write specialized** database applications program
  - Use administration tools
  - e.g. Database Administrator

Role of DBA (Database Administrator)

Section - 8

Role of DBA

- Schema Definition
  - DBA **defines the logical schema** of the database.
- Storage Structure and Access Method Definition
  - DBA **decides how the data is to be represented** in the database & how to access it.
- Defining Security and Integrity Constraints
  - DBA **decides on various security and integrity constraints**.
- Granting of Authorization for Data Access
  - DBA **determines which user needs access to which part** of the database.
- Liaison with Users
  - DBA **provide necessary data** to the user.

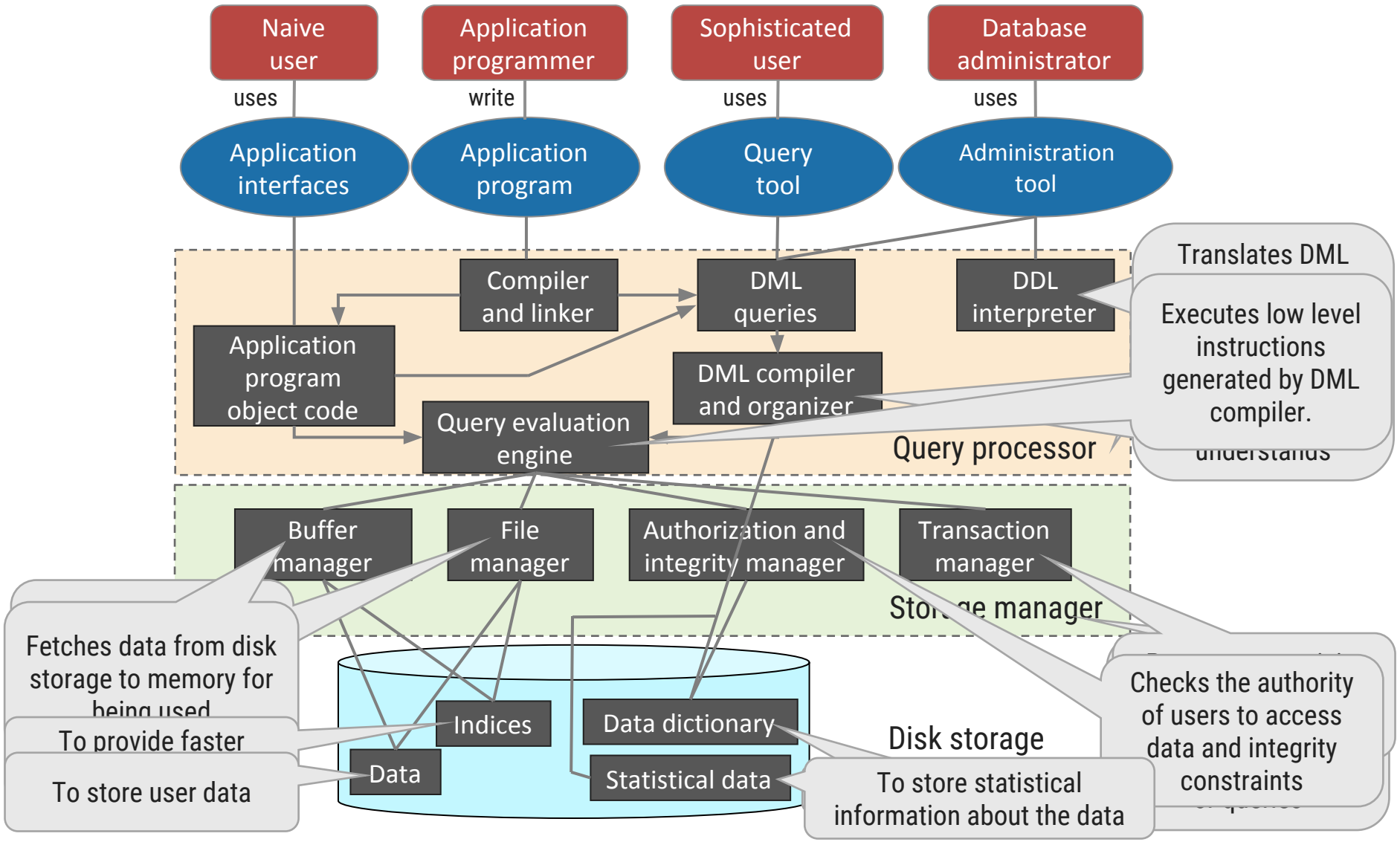
Role of DBA

- Assisting Application Programmer
  - DBA **provides assistance to application programmers** to develop application programs.
- Monitoring Performance
  - DBA **ensures that better performance is maintained** by making a change in the physical or logical schema if required.
- Backup and Recovery
  - DBA **backing up the database** on some storage devices such as DVD, CD or magnetic tape or remote servers and **recover the system in case of failures**, such as flood or virus attack from this backup.

Database System Architecture

Section - 9

Database System Architecture



Keys

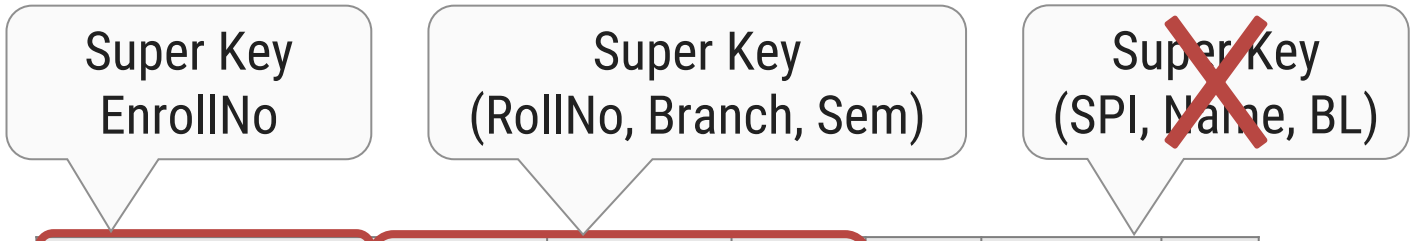
Section - 10

Keys in DBMS

- Key is a attribute or set of attributes that is used to identify data in entity sets.
- Key is defined for unique identification of rows in table.
- Key allow us to find the relation between two tables.
- Types of Keys are :-
  - **Super Key**
  - **Candidate Key**
  - **Primary Key**
  - **Alternate Key**
  - **Foreign Key**

Super Key

- A super key is a set of one or more **attributes whose values uniquely identifies each record** within a relation (table).



EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
190540107002	102	CE	3	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
190540106002	102	CI	3	9	Nilesh	0
180540107001	101	CE	5	7	Hitesh	1
180540106001	101	CI	5	8	Tarun	0
180540106002	102	CI	5	9	Suresh	0

Candidate Key

- A candidate key is a **subset of a super key**.
- A candidate key is a single attribute or the least combination of attributes that uniquely identifies each record in the table.
- A candidate key is a **super key for which no proper subset is a super key**.
- **Every candidate key is a super key** but **every super key is not a candidate key**.

Candidate Key
EnrollNo

Candidate Key
(RollNo, Branch, Sem)

EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
180540107002	101	CE	5	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
180540106002	101	CI	5	9	Nilesh	0

Primary Key

- A primary key is a **candidate key that is chosen by database designer** to identify tuples uniquely in a relation (table).

Primary Key

Candidate Key
EnrollNo

Candidate Key
(RollNo, Branch, Sem)

EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
180540107002	101	CE	5	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
180540106002	101	CI	5	9	Nilesh	0

Alternate Key

- An alternate key is a **candidate key that is not chosen by database designer** to identify tuples uniquely in a relation.

Primary Key

Candidate Key
EnrollNo

Candidate Key
(RollNo, Branch, Sem)

Alternate Key

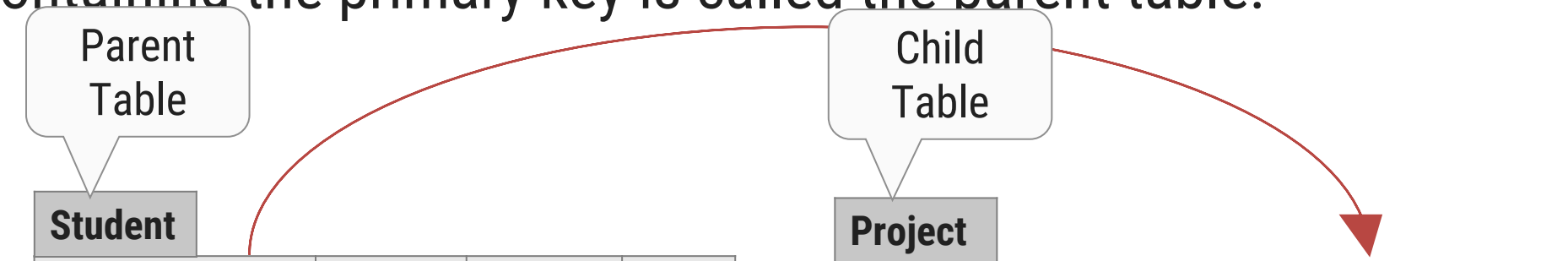
EnrollNo	RollNo	Branch	Sem	SPI	Name	BL
190540107001	101	CE	3	8	Raju	0
180540107002	101	CE	5	7	Mitesh	1
190540106001	101	CI	3	6	Mayur	2
180540106002	101	CI	5	9	Nilesh	0

Primary Key rules

- A primary key **may have one or more attributes**.
- There is **only one primary key** in the relation (table).
- A primary key **attribute value cannot be NULL**.
- Generally, the **value of a primary key attribute does not change**.

Foreign Key

- A foreign key is **used to link two relations** (tables).
- A foreign key is an **attribute** or collection of attributes in one table that **refers to the primary key in another table**.
- A table containing the foreign key is called the child table, and the table containing the primary key is called the parent table.



Student

EnrollNo	Name	Branch	Sem
190540107001	Raju	CE	3
190540107002	Mitesh	CE	3
190540107003	Nilesh	CE	3
190540107004	Meet	CE	3

Project

ProjectID	Title	EnrollNo
101	Bank	190540107001
102	College	190540107002
103	School	190540107003
104	Hospital	190540107001

Introduction to SQL

Section - 11

SQL- Structured Query Language

- SQL stands for **Structured Query Language**. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a **user to create, read, update and delete** relational databases and tables.
- All the RDBMS like **MySQL, Informix, Oracle, MS Access and SQL Server use SQL** as their standard database language.

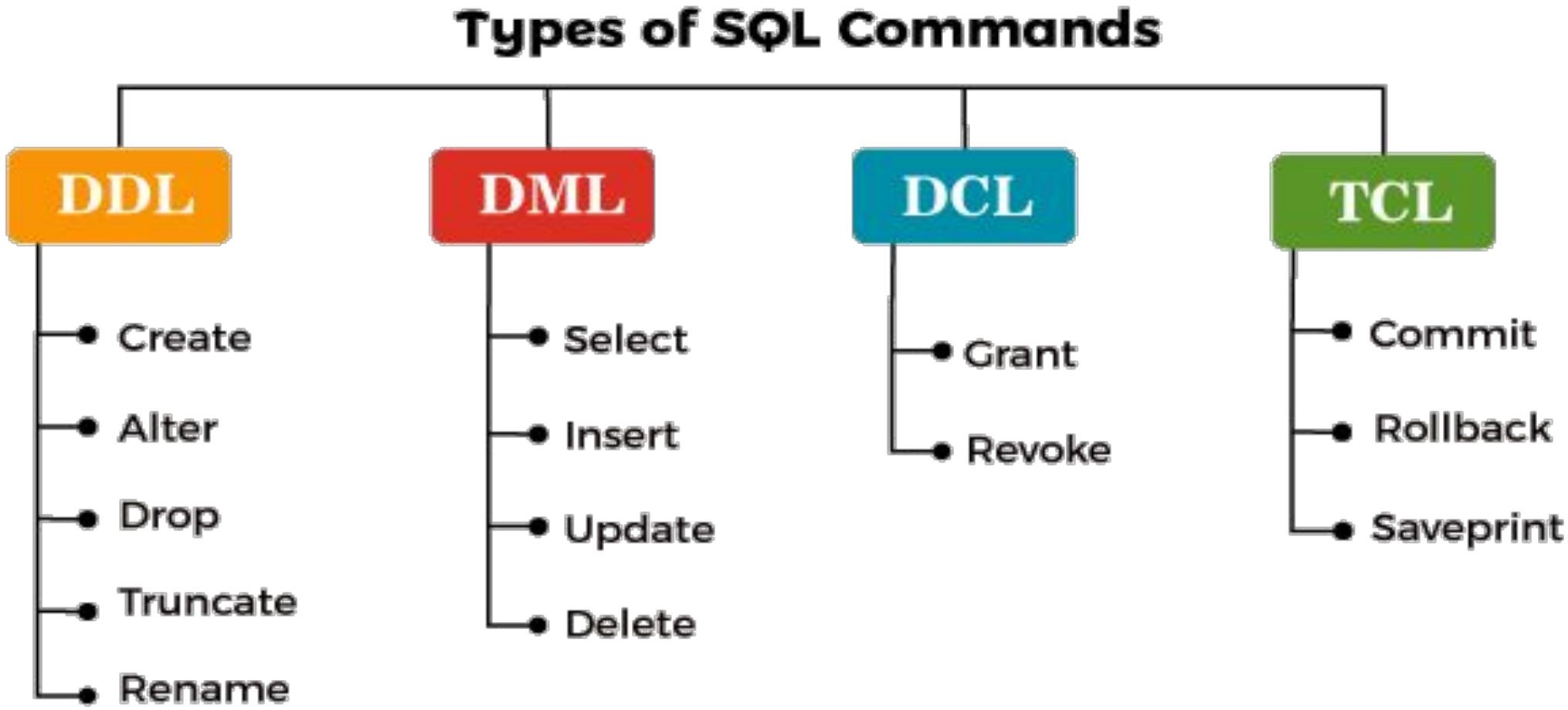
Characteristics of SQL

- SQL is **easy to learn**.
- SQL is used to **access data** from relational database management systems.
- SQL can **execute queries** against the database.
- SQL is used to **describe the data**.
- SQL is used to **define the data** in the database and manipulate it when needed.
- SQL is used to **create and drop** the database and table.
- SQL is used to **create a view, stored procedure, function** in a database.
- SQL allows users to **set permissions** on tables, procedures, and views.

SQL Commands

- SQL commands are **instructions**.
- It is used to **communicate** with the database.
- It is also used to perform **specific tasks, functions, and queries** of data.
- SQL can perform various tasks like **create a table, add data to tables, drop the table, modify the table, set permission** for users.

Types of SQL Commands



DDL- Data Definition Language

- DDL **changes the structure** of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are **auto-committed** that means it permanently save all the changes in the database.
- Here are some commands that come under DDL:

  - **CREATE -** It is used to create a new table in the database.
  - **DROP -** It is used to delete both the structure and record stored in the table.
  - **ALTER -** It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.
  - **TRUNCATE -** It is used to delete all the rows from the table and free the space containing the table.

DML- Data Manipulation Language

- DML commands are used to **modify the database**. It is responsible for all form of changes in the database.
- The command of DML is **not auto-committed** that means it can't permanently save all the changes in the database. They can be rollback.
- Commands that come under DML:
  - INSERT- It is used to insert data into the row of a table.
  - UPDATE- It is used to update or modify the value of a column in the table.
  - DELETE - It is used to remove one or more row from a table.

DCL- Data Control Language

- DCL commands are used to **grant and take back authority** from any database user.
- Whenever we want to **control the access** to the data present in SQL tables, we will use DCL commands in SQL. Only the **authorized users** can access the data stored in the tables.
- Commands that come under DCL:
  - **GRANT-** Access privileges can be assigned to a user for the databases and tables using the GRANT command.
  - **REVOKE -** All the access privileges which are already assigned to the user can be revoked by using the REVOKE command.

TCL- Transaction Control Language

- TCL commands are generally used in transactions.
- Using TCL commands in SQL, we can save our transactions to the database and roll them back to a specific point in our transaction.
- We can also save a particular portion of our transaction using the SAVEPOINT command.
- Here are some commands that come under TCL:
  - **COMMIT-** Commit command is used to save all the transactions to the database.
  - **ROLLBACK -** Rollback command is used to undo transactions that have not already been saved to the database.
  - **SAVEPOINT-** It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Data Types in SQL

- Data types are used to **represent the nature of the data** that can be stored in the database table.
- For example, in a particular column of a table, if we want to store a string type of data then we will have to declare a string data type of this column.
- Data types mainly classified into three categories for every database.
  - **String Data types**
  - **Numeric Data types**
  - **Date and time Data types**

String Data Types

- **CHAR(Size)-** It is used to specify a fixed length string that can contain numbers, letters, and special characters. Its size can be 0 to 255 characters. Default is 1.
- **VARCHAR(Size)-** It is used to specify a variable length string that can contain numbers, letters, and special characters. Its size can be from 0 to 65535 characters.
- **BINARY(Size)-** It is equal to CHAR() but stores binary byte strings. Its size parameter specifies the column length in the bytes. Default is 1.

Numeric Data Types

- **BIT(Size)-** It is used for a bit-value type. The number of bits per value is specified in size. Its size can be 1 to 64. The default value is 1.
- **INT(size)-** It is used for the integer value. Its signed range varies from -2147483648 to 2147483647 and unsigned range varies from 0 to 4294967295. The size parameter specifies the max display width that is 255.
- **FLOAT(size, d)-** It is used to specify a floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal point is specified by d parameter.
- **Numeric (p, d) -** A fixed point number with user defined precision. It consists of p digits and d of the p digits are to the right of the decimal point.

Date and time Data Types

- **DATE -** It is used to specify date format YYYY-MM-DD. Its supported range is from '1000-01-01' to '9999-12-31'.
- **DATETIME(fsp)-** It is used to specify date and time combination. Its format is YYYY-MM-DD hh:mm:ss. Its supported range is from '1000-01-01 00:00:00' to 9999-12-31 23:59:59'

Questions asked in AKTU University Question paper

1. List and explain the advantages of DBMS over file based system. **OR** Explain disadvantages of files based system.
2. Draw and explain 3 level architecture of DBMS.
3. List and explain different categories/types of database users.
4. List and explain different tasks/roles/functions/duties of DBA (Database Administrator).
5. Explain DBMS architecture with block diagram. **OR** Explain Database System architecture with block diagram.
6. Explain all types of Keys with examples ?
7. Explain all types of SQL commands with syntax and example?

Thank You

Ms. Sonali Gahlot
Computer Engineering Department
Ajay Kumar Garg Engineering College, Ghaziabad
sonaligahlot5@gmail.com
9258300179