

Comprehensive Notes on: Unit - 1: Database Concepts

Ashish Kumar Bhoi
Odisha Adarsha Vidyalaya Bibina

September 8, 2025

Contents

1 Comprehensive Notes	3
1.1 Introduction to Databases	3
1.1.1 What are Data and Information?	3
1.1.2 What is a Database?	3
1.2 Need for a Database	3
1.3 Database Management System (DBMS)	4
1.3.1 Characteristics of the DBMS Approach	4
1.3.2 Advantages of using a DBMS	5
1.4 The Relational Database Model	5
1.4.1 Key Terminologies	5
1.4.2 Keys in a Relational Database	6
1.5 Structured Query Language (SQL)	6
1.5.1 Data Definition Language (DDL)	6
1.5.2 Data Manipulation Language (DML)	7
1.5.3 Clauses and Keywords for Data Retrieval	8
2 Practice Questions	10
2.1 Multiple Choice Questions (MCQs)	10
2.2 One-Word Answer Questions	12
2.3 Differentiate Between Questions	13
2.4 Short Answer Questions	14
2.5 Long Answer Questions	14
2.6 Practical Application Questions	16

1 Comprehensive Notes

1.1 Introduction to Databases

1.1.1 What are Data and Information?

In the world of computers, two words, 'Data' and 'Information', are often used. Although they seem similar, they have different meanings.

- **Data:** Data is a collection of raw, unorganized facts, figures, and symbols. It doesn't have any specific meaning on its own. For example, if we have a list of numbers like 85, 92, 78, 65, it is just data. We don't know what these numbers represent. They could be marks, ages, or weights.
- **Information:** Information is data that has been processed, organized, and structured in a way that gives it meaning and context. When we process data, it becomes useful. For example, if we are told that the numbers 85, 92, 78, and 65 are the marks of a student in four different subjects, this is now information.

The process of converting data into information is called **Data Processing**.

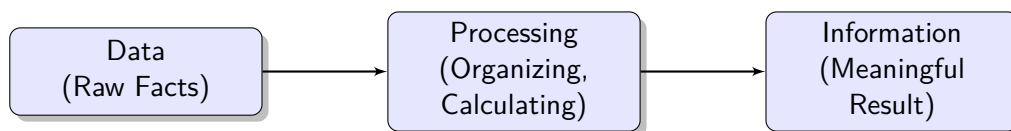


Figure 1: The relationship between Data, Processing, and Information.

1.1.2 What is a Database?

A **database** is an organized collection of related data, stored in a way that it can be easily accessed, managed, and updated. Think of it as a large, electronic filing cabinet. For example, a school's database might contain tables with information about students, teachers, and courses. All this data is related and organized for the school's use.

A database has the following properties:

- It represents some aspect of the real world (also called a *miniworld*).
- It is a logically coherent collection of data with some inherent meaning.
- It is designed, built, and populated with data for a specific purpose.
- It can be of any size and complexity.

1.2 Need for a Database

Before databases, data was stored in separate files in a computer system. This was called the **File Processing System**. For example, the Sales department of a company would have its own files, and the Payroll department would have its own separate files, even if both contained information about the same employees.

This old system had several major problems:

1. **Data Redundancy:** The same piece of information was stored in multiple files. For example, an employee's name and address might be in both the sales file and the payroll file. This wastes a lot of storage space.
2. **Data Inconsistency:** If an employee's information changed (e.g., they moved to a new address), it had to be updated in all files where it was stored. If it was updated in one file but not another, the data would become inconsistent (conflicting).

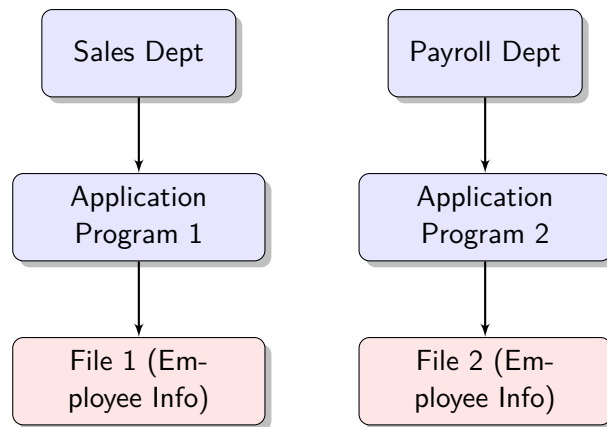


Figure 2: Traditional File Processing System leading to data duplication.

3. **Lack of Data Integration:** Since data was stored in separate, independent files, it was very difficult to access information from multiple files at the same time to answer complex questions.

The database approach was developed to solve these problems by storing all related data in a single, centralized repository.

1.3 Database Management System (DBMS)

A **Database Management System (DBMS)** is a software package that allows users to create, maintain, and control access to a database. It acts as an interface between the users and the database.

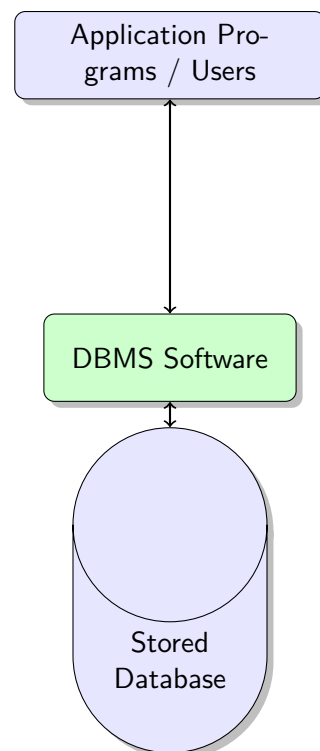


Figure 3: A simplified view of a DBMS environment.

Examples of popular DBMS include MySQL, Oracle, Microsoft SQL Server, and PostgreSQL.

1.3.1 Characteristics of the DBMS Approach

- **Self-describing Nature:** A DBMS stores not only the data but also the description of the data, known as **metadata** (data about data). This metadata is stored in a **data dictionary** and includes information about the structure of the data, data types, and constraints.

- **Insulation between Programs and Data:** Since the data definition (metadata) is stored separately from the application programs, any changes to the structure of the data can be made without changing the programs that access it. This is called **Program-Data Independence**.
- **Sharing of Data and Multi-user Access:** A DBMS allows multiple users to access and update the database simultaneously in a controlled manner, ensuring that the data remains consistent.

1.3.2 Advantages of using a DBMS

- **Controls Redundancy:** By storing data in a central location, it avoids duplication.
- **Enforces Data Consistency:** Ensures that if a data item is changed, the change is reflected everywhere.
- **Improves Data Sharing:** Different users and applications can access the same data.
- **Enforces Security:** The Database Administrator (DBA) can control who can access what data.
- **Provides Backup and Recovery:** DBMS provides facilities to back up the data and recover it in case of a system failure.

1.4 The Relational Database Model

The most popular type of database today is the **Relational Database**, which was proposed by E.F. Codd in 1970. In this model, data is organized into tables.

1.4.1 Key Terminologies

Let's understand the key terms using an example table called **STUDENT**.

Table 1: Example: STUDENT Table (Relation)

RollNo	Name	Class	City
101	Ankit	XII	Delhi
102	Priya	XII	Mumbai
103	Rohan	XI	Delhi

- **Relation:** A table that stores data. In the example, **STUDENT** is a relation.
- **Tuple:** A row in a table. Each tuple represents a single record. The STUDENT table has 3 tuples.
- **Attribute:** A column in a table. Each attribute represents a property or characteristic. The STUDENT table has 4 attributes: RollNo, Name, Class, and City.
- **Domain:** The set of possible values for an attribute. For example, the domain for the 'Class' attribute could be the set {'XI', 'XII'}.
- **Degree:** The total number of attributes (columns) in a relation. The degree of the STUDENT relation is 4.
- **Cardinality:** The total number of tuples (rows) in a relation. The cardinality of the STUDENT relation is 3.
- **Relation Schema:** The structure of a relation, consisting of the relation name and its list of attributes. For our example, the schema is: STUDENT(RollNo, Name, Class, City).

1.4.2 Keys in a Relational Database

Keys are attributes or a set of attributes that help to uniquely identify a tuple (row) in a relation (table).

- **Superkey:** A set of one or more attributes that, taken collectively, can uniquely identify a tuple in a relation. For example, in the STUDENT table, {RollNo} is a superkey. {RollNo, Name} is also a superkey.
- **Candidate Key:** A minimal superkey. This means it is a superkey from which no attribute can be removed without it losing its unique identification property. A relation can have multiple candidate keys. In our example, {RollNo} is a candidate key. If students could also be uniquely identified by their email, then {Email} would be another candidate key.
- **Primary Key:** One of the candidate keys is chosen by the database designer to be the primary means of uniquely identifying tuples. The primary key's value cannot be NULL (empty). In the STUDENT table, **RollNo** is the best choice for the primary key. The primary key is usually underlined in the schema: STUDENT(RollNo, Name, Class, City).
- **Foreign Key:** An attribute (or set of attributes) in one relation that refers to the primary key of another relation. It is used to link two tables together. For example, if we have another table MARKS(ExamID, RollNo, Subject, Score), the 'RollNo' attribute in the MARKS table would be a foreign key that refers to the 'RollNo' primary key in the STUDENT table. This creates a relationship between the two tables.

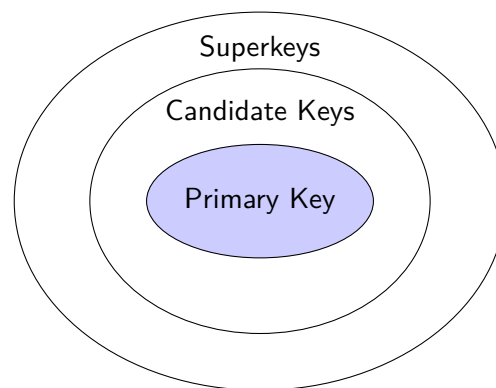


Figure 4: Relationship between different types of Keys.

1.5 Structured Query Language (SQL)

Structured Query Language (SQL) is the standard language used to communicate with relational databases. It is used to perform tasks such as retrieving data, inserting data, updating data, and creating or modifying the database structure. SQL commands are broadly divided into the following categories:

1.5.1 Data Definition Language (DDL)

DDL commands are used to define and manage the database structure (schema).

- **CREATE:** Used to create new databases, tables, etc.
- **ALTER:** Used to modify the structure of an existing database object, like adding a column to a table.
- **DROP:** Used to delete entire databases or tables.

Example: CREATE TABLE This command creates a new table. You must specify the table name, column names, and data types for each column.

```
CREATE TABLE Teacher (
    Teacher_ID INTEGER PRIMARY KEY,
    First_Name VARCHAR(20) NOT NULL,
    Last_Name VARCHAR(20),
    Salary DECIMAL(10, 2) DEFAULT 40000,
    Dept_No INTEGER
);
```

In this example:

- INTEGER, VARCHAR(20), and DECIMAL(10, 2) are data types.
- PRIMARY KEY is a constraint that makes Teacher_ID the primary key.
- NOT NULL is a constraint that ensures a value must be entered for First_Name.
- DEFAULT 40000 sets a default value for Salary if no value is provided.

1.5.2 Data Manipulation Language (DML)

DML commands are used to manage the data within the tables.

- **SELECT:** Used to retrieve data from a database.
- **INSERT:** Used to insert new data (rows) into a table.
- **UPDATE:** Used to modify existing data in a table.
- **DELETE:** Used to remove data (rows) from a table.

Example: INSERT This command adds a new row to the 'Teacher' table.

```
INSERT INTO Teacher (Teacher_ID, First_Name, Last_Name, Salary, Dept_No)
VALUES (101, 'Surbhi', 'Bansal', 52000, 4);
```

Example: UPDATE This command changes the salary of the teacher with ID 101.

```
UPDATE Teacher
SET Salary = 55000
WHERE Teacher_ID = 101;
```

Example: DELETE This command removes the record of the teacher with ID 101.

```
DELETE FROM Teacher
WHERE Teacher_ID = 101;
```

Example: SELECT This is the most commonly used SQL command. It retrieves data from one or more tables.

- To get all information about all teachers:

```
SELECT * FROM Teacher;
```

- To get the names of teachers whose salary is more than 50000:

```
SELECT First_Name, Last_Name FROM Teacher
WHERE Salary > 50000;
```

1.5.3 Clauses and Keywords for Data Retrieval

The SELECT statement is very powerful and can be customized using various clauses and keywords to retrieve precisely the data you need.

- **The WHERE Clause:** This is the most fundamental clause for filtering. It is used to extract only those records that fulfill a specified condition. You can use comparison operators (=, <>, <, >, <=, >=) and logical operators (AND, OR, NOT).

```
-- Selects teachers from department 4 who earn more than 50000
SELECT First_Name, Salary FROM Teacher
WHERE Dept_No = 4 AND Salary > 50000;
```

- **The ORDER BY Clause:** This clause is used to sort the result-set in ascending (ASC) or descending (DESC) order. By default, it sorts in ascending order.

```
SELECT * FROM Teacher ORDER BY First_Name ASC;
```

- **Aggregate Functions:** These functions perform a calculation on a set of rows and return a single, summary value.

- COUNT(*) or COUNT(column): Counts the number of rows.
- SUM(column): Calculates the sum of values in a numeric column.
- AVG(column): Calculates the average of values in a numeric column.
- MAX(column): Returns the largest value in a column.
- MIN(column): Returns the smallest value in a column.

```
-- Find the number of teachers and their average salary
SELECT COUNT(*), AVG(Salary) FROM Teacher;
```

- **The GROUP BY Clause:** This clause is used with aggregate functions to group the result-set by one or more columns. It collects data from multiple records and groups them so that aggregate functions can be applied to each group.

```
-- Count the number of teachers in each department
SELECT Dept_No, COUNT(*) AS NumberOfTeachers
FROM Teacher
GROUP BY Dept_No;
```

- **The HAVING Clause:** The WHERE clause cannot be used with aggregate functions. The HAVING clause was added to SQL to filter groups based on the results of aggregate functions.

```
-- Find departments that have more than 5 teachers
SELECT Dept_No, COUNT(*)
FROM Teacher
GROUP BY Dept_No
HAVING COUNT(*) > 5;
```

- **Joining Tables:** Most of the time, data is stored in multiple related tables. To query this data, you must join the tables. A join combines rows from two or more tables based on a related column between them (a Primary Key-Foreign Key link).

```
-- Get the name of each teacher and their department's name
SELECT T.First_Name, D.Dept_Name
FROM Teacher AS T, Department AS D
WHERE T.Dept_No = D.Dept_ID;
```

Here, aliases ('T' and 'D') are used for table names to simplify the query. The WHERE clause specifies the join condition.

- **Other Useful Keywords:**

- **DISTINCT:** Eliminates duplicate rows from the query result.

```
SELECT DISTINCT Dept_No FROM Teacher;
```

- **LIKE:** Used in a 'WHERE' clause for pattern matching in text. Wildcards % (matches any sequence of characters) and _ (matches any single character) are used.

```
-- Finds teachers whose first name starts with 'A'  
SELECT First_Name FROM Teacher WHERE First_Name LIKE 'A%';
```

- **IS NULL / IS NOT NULL:** Used to check for or exclude records that have NULL (empty) values.

```
-- Finds teachers where the last name is not specified  
SELECT * FROM Teacher WHERE Last_Name IS NULL;
```

2 Practice Questions

2.1 Multiple Choice Questions (MCQs)

1. Raw, unorganized facts are known as:

- | | |
|-----------------|--------------|
| (A) Information | (C) Database |
| (B) Data | (D) Schema |

Answer: (B)

2. A software package used to manage databases is called a:

- | | |
|----------------------|-----------------|
| (A) Operating System | (C) Spreadsheet |
| (B) DBMS | (D) File System |

Answer: (B)

3. In a relational database, a table is also known as a:

- | | |
|---------------|--------------|
| (A) Tuple | (C) Relation |
| (B) Attribute | (D) Degree |

Answer: (C)

4. The total number of columns in a table is called its:

- | | |
|-----------------|------------|
| (A) Cardinality | (C) Domain |
| (B) Degree | (D) Schema |

Answer: (B)

5. Which key is used to uniquely identify each row in a table?

- | | |
|-----------------|-------------------|
| (A) Foreign Key | (C) Primary Key |
| (B) Superkey | (D) Candidate Key |

Answer: (C)

6. Which SQL command is used to retrieve data from a database?

- | | |
|------------|------------|
| (A) INSERT | (C) GET |
| (B) UPDATE | (D) SELECT |

Answer: (D)

7. 'CREATE TABLE' is an example of which type of SQL command?

- | | |
|---------|---------|
| (A) DML | (C) DCL |
| (B) DDL | (D) TCL |

Answer: (B)

8. Which clause is used to sort the result of a query?

- (A) GROUP BY
- (B) SORT BY
- (C) ORDER BY
- (D) ARRANGE BY

Answer: (C)

9. A minimal superkey is called a:

- (A) Primary Key
- (B) Candidate Key
- (C) Foreign Key
- (D) Alternate Key

Answer: (B)

10. A key that establishes a relationship between two tables is a:

- (A) Primary Key
- (B) Superkey
- (C) Composite Key
- (D) Foreign Key

Answer: (D)

11. Which problem arises from storing the same data in multiple locations?

- (A) Data Security
- (B) Data Integration
- (C) Data Redundancy
- (D) Data Access

Answer: (C)

12. In SQL, which command is used to add new rows to a table?

- (A) ADD
- (B) INSERT
- (C) UPDATE
- (D) CREATE

Answer: (B)

13. The total number of rows in a table is called its:

- (A) Degree
- (B) Domain
- (C) Cardinality
- (D) Attribute

Answer: (C)

14. The 'WHERE' clause in a SELECT statement is used for:

- (A) Sorting rows
- (B) Grouping rows
- (C) Filtering rows
- (D) Joining tables

Answer: (C)

15. Which aggregate function returns the number of rows?

- (A) SUM()
- (B) AVG()
- (C) MAX()
- (D) COUNT()

Answer: (D)

16. To delete a table structure completely, which command is used?

- (A) DELETE TABLE
- (B) REMOVE TABLE

- (C) DROP TABLE
- (D) ERASE TABLE

Answer: (C)

17. A row in a table is also known as a:

- (A) Field
- (B) Attribute

- (C) Tuple
- (D) Relation

Answer: (C)

18. Which clause filters the results of a GROUP BY query?

- (A) WHERE
- (B) HAVING

- (C) FILTER
- (D) WITH

Answer: (B)

19. The 'NOT NULL' keyword in a 'CREATE TABLE' statement is a:

- (A) Data Type
- (B) Constraint

- (C) Clause
- (D) Function

Answer: (B)

20. Data about data is known as:

- (A) Hyperdata
- (B) Information

- (C) Metadata
- (D) A database

Answer: (C)

2.2 One-Word Answer Questions

1. What is an organized collection of related data called? (Database)
2. The set of possible values for an attribute is called its _____. (Domain)
3. A column in a table is also known as an _____. (Attribute)
4. Which language is used to interact with relational databases? (SQL)
5. The structure of a relation, including its name and attributes, is its _____. (Schema)
6. What command is used to modify the structure of a table? (ALTER)
7. What command is used to modify the data within a table? (UPDATE)
8. In SQL, what symbol is used to select all columns from a table? (*)
9. The property that allows changes to the data structure without changing application programs is _____. (Program-Data Independence)
10. What kind of key is a combination of two or more columns used to identify a record? (Composite Key)
11. What command is used to remove all rows from a table but keep the structure? (DELETE FROM table_name)
12. The problem of data becoming conflicting in different files is called data _____. (Inconsistency)

13. Who is responsible for the overall administration of a database? (DBA)
14. The most popular database model is the _____ model. (Relational)
15. Which clause in a SELECT statement specifies the table to retrieve data from? (FROM)
16. What is the default sort order for the ORDER BY clause? (Ascending/ASC)
17. Which SQL command category includes INSERT, UPDATE, and DELETE? (DML)
18. A candidate key that is not chosen as the primary key is called an _____ key. (Alternate)
19. What special value is used to represent a missing or unknown value in a cell? (NULL)
20. The process of converting data into meaningful information is known as data _____. (Processing)

2.3 Differentiate Between Questions

1. Data and Information.

Answer: Data is a collection of raw, unorganized facts without any context (e.g., the number '50'). Information is data that has been processed and given context, making it meaningful (e.g., 'Student's marks are 50').

2. DDL and DML.

Answer: DDL (Data Definition Language) commands like CREATE and ALTER are used to define and manage the structure of the database. DML (Data Manipulation Language) commands like INSERT, UPDATE, and SELECT are used to manage the data within that structure.

3. Primary Key and Foreign Key.

Answer: A Primary Key is an attribute that uniquely identifies each row within a single table. A Foreign Key is an attribute in one table that refers to the Primary Key of another table, used to link the two tables.

4. Degree and Cardinality.

Answer: Degree is the number of attributes (columns) in a table. Cardinality is the number of tuples (rows) in a table.

5. DELETE and DROP.

Answer: The 'DELETE' command is a DML command used to remove one or more rows from a table. The 'DROP' command is a DDL command used to permanently delete the entire table, including its structure and data.

6. Superkey and Candidate Key.

Answer: A Superkey is any set of attributes that can uniquely identify a row. A Candidate Key is a minimal superkey, meaning no attribute can be removed from it without it losing its uniqueness property.

7. WHERE and HAVING clause.

Answer: The 'WHERE' clause is used to filter rows before any grouping is done. The 'HAVING' clause is used to filter groups after the 'GROUP BY' clause has been applied.

8. CHAR and VARCHAR.

Answer: 'CHAR' is a fixed-length data type, so it always uses the same amount of storage space regardless of the string's length. 'VARCHAR' is a variable-length data type, so it only uses storage space for the characters you actually store.

9. File Processing System and DBMS.

Answer: A File Processing System stores data in separate, independent files, leading to redundancy and inconsistency. A DBMS stores data in a centralized, integrated database, which controls redundancy and improves data consistency and security.

10. UPDATE and ALTER.

Answer: The 'UPDATE' command is a DML command used to modify the actual data (records) within a table. The 'ALTER' command is a DDL command used to modify the structure of the table itself, such as adding or deleting a column.

2.4 Short Answer Questions

1. **What is a database? Give a real-world example.**

Answer: A database is an organized collection of related data. A real-world example is a library database, which stores information about books (title, author), members (name, address), and loans (which member has which book).

2. **List two major problems of a traditional file processing system.**

Answer: Two major problems are (1) Data Redundancy, where the same data is duplicated across multiple files, wasting space, and (2) Data Inconsistency, where updating data in one file but not another leads to conflicting information.

3. **What is SQL? Name its two main sub-languages.**

Answer: SQL (Structured Query Language) is the standard language for managing relational databases. Its two main sub-languages are DDL (Data Definition Language) for defining structure and DML (Data Manipulation Language) for managing data.

4. **Explain the purpose of the primary key in a table.**

Answer: The purpose of a primary key is to uniquely identify every record (row) in a table. It ensures that no two rows are identical and that every row can be located precisely. It cannot contain NULL values.

5. **What are aggregate functions in SQL? List three of them.**

Answer: Aggregate functions perform a calculation on a set of values and return a single summary value. Three common aggregate functions are 'COUNT()' (counts the number of rows), 'SUM()' (calculates the sum of values), and 'AVG()' (calculates the average of values).

6. **What is metadata?**

Answer: Metadata is "data about data." In a DBMS, it is the data that describes the structure of the database itself, such as table names, column names, data types, and constraints. This information is stored in the data dictionary.

7. **Why is the 'NOT NULL' constraint used?**

Answer: The 'NOT NULL' constraint is used to ensure that a column cannot have a NULL (empty) value. This is important for fields that must contain data for a record to be meaningful, such as a student's name.

8. **What does the 'GROUP BY' clause do in an SQL query?**

Answer: The 'GROUP BY' clause groups rows from a result set that have the same values in specified columns into summary rows. It is typically used with aggregate functions to perform calculations on each group.

9. **What is a foreign key, and what is its purpose?**

Answer: A foreign key is a column in one table that uniquely identifies a row of another table. Its purpose is to link the two tables together and enforce referential integrity, ensuring that a value in the foreign key column corresponds to an existing primary key value in the other table.

10. **What will the command 'DELETE FROM Student;' do?**

Answer: This command will delete all the rows (records) from the 'Student' table. The table structure, including its columns and constraints, will remain, but the table will be empty of data.

2.5 Long Answer Questions

1. **Describe the advantages of using a DBMS over a traditional file processing system. Explain at least four advantages in detail.**

Answer: Using a DBMS offers several significant advantages over traditional file systems.

- (a) **Control of Data Redundancy:** In a file system, the same information might be stored in different files, leading to wastage of space. A DBMS integrates all data into a single database, so data is stored only once, controlling this redundancy.
- (b) **Data Consistency:** By controlling redundancy, a DBMS also improves data consistency. When a data item needs to be updated, it only has to be changed in one place, which ensures that there are no conflicting versions of the same data.

- (c) **Improved Data Sharing:** A DBMS allows data to be shared among multiple users and applications. It manages concurrent access to the data, ensuring that the actions of one user do not negatively affect others.
- (d) **Enforcement of Security:** A DBMS provides a robust security system. The Database Administrator (DBA) can create user accounts and specify access rights, preventing unauthorized users from viewing or modifying sensitive data.

2. **What are keys in a database? Explain Superkey, Candidate Key, and Primary Key with a suitable example.**

Answer: Keys are attributes or sets of attributes used to uniquely identify records in a table. Consider a table: EMPLOYEE(EMPID, AadharNo, Name, Department).

- (a) **Superkey:** A superkey is a set of one or more attributes that can uniquely identify a tuple. In the EMPLOYEE table, {EMPID} is a superkey. {AadharNo} is also a superkey. Furthermore, combinations like {EMPID, Name} or {AadharNo, Department} are also superkeys because their components are sufficient for unique identification.
- (b) **Candidate Key:** A candidate key is a minimal superkey, meaning it's a superkey with no redundant attributes. In our example, both {EMPID} and {AadharNo} are candidate keys because they can uniquely identify an employee on their own, and we cannot remove any attribute from them.
- (c) **Primary Key:** A primary key is the candidate key that is chosen by the database designer to be the main identifier for the table. Its value must be unique and cannot be NULL. In the EMPLOYEE table, the designer would likely choose **EMPID** as the primary key.

3. **Consider the tables below and write SQL queries for the following questions.**

STUDENT(ROLLNo, Name, Class, City)

MARKS(ReportID, ROLLNo, Subject, Marks)

- (a) Display the names of all students from 'Delhi'.
- (b) Display the name and marks of students in the 'Maths' subject.
- (c) Display the names of students who scored more than 90 marks.
- (d) Display the names of all students in descending alphabetical order.
- (e) Display the total number of students in the 'STUDENT' table.

Answer:

- (a) `SELECT Name FROM STUDENT WHERE City = 'Delhi';`
- (b) `SELECT S.Name, M.Marks FROM STUDENT S, MARKS M WHERE S.ROLLNo = M.ROLLNo AND M.Subject = 'Maths';`
- (c) `SELECT S.Name FROM STUDENT S, MARKS M WHERE S.ROLLNo = M.ROLLNo AND M.Marks > 90;`
- (d) `SELECT Name FROM STUDENT ORDER BY Name DESC;`
- (e) `SELECT COUNT(*) FROM STUDENT;`

4. **Explain the purpose of the 'CREATE', 'ALTER', and 'DROP' commands in SQL with syntax and an example for each.**

Answer: These are DDL (Data Definition Language) commands used to manage the database structure.

- (a) **CREATE:** This command is used to create a new database object, most commonly a table. **Syntax:** `CREATE TABLE table_name (column1 datatype, ...);` **Example:** `CREATE TABLE Products (ProductID INT, ProductName VARCHAR(50));`
- (b) **ALTER:** This command is used to modify the structure of an existing table, such as adding, deleting, or modifying columns. **Syntax:** `ALTER TABLE table_name ADD column_name datatype;` **Example:** `ALTER TABLE Products ADD Price DECIMAL(10,2);` (This adds a 'Price' column).
- (c) **DROP:** This command is used to permanently delete an entire database object, like a table, including its structure and all its data. **Syntax:** `DROP TABLE table_name;` **Example:** `DROP TABLE Products;`

5. **What is referential integrity? How is it implemented in a relational database? Explain with an example.**

Answer: Referential integrity is a rule that ensures that relationships between tables remain consistent. When one table has a foreign key to another table, the concept of referential integrity states that you cannot add a record to the table with the foreign key unless there is a corresponding primary key record in the linked table. It also restricts actions like deleting or changing a primary key record if dependent records exist. It is implemented using **foreign key constraints**. **Example:** Consider two tables, DEPARTMENT(DeptID, DeptName) and TEACHER(TeacherID, Name, DeptID). Here, DeptID is the primary key in DEPARTMENT and a foreign key in TEACHER. The foreign key constraint on TEACHER.DeptID ensures that:

- (a) You cannot insert a new teacher with a DeptID that does not already exist in the DEPARTMENT table.
- (b) You cannot delete a department from the DEPARTMENT table if there are still teachers assigned to that department in the TEACHER table.

This prevents "orphan" records (a teacher assigned to a non-existent department) and maintains consistency.

2.6 Practical Application Questions

1. **You need to create a table to store employee details. The table should be named 'Employee' and have the following columns: 'E_ID' (integer, primary key), 'E_Name' (text, cannot be empty), 'Hire_Date' (date), and 'Salary' (decimal). Write the SQL 'CREATE TABLE' command for this.**

Answer:

```
CREATE TABLE Employee (  
    E_ID INTEGER PRIMARY KEY,  
    E_Name VARCHAR(50) NOT NULL,  
    Hire_Date DATE,  
    Salary DECIMAL(10, 2)  
);
```

2. **After creating the 'Employee' table, you realize you need to add a column for the employee's department, called 'Dept_ID' (integer). Write the SQL command to add this new column to the existing table.**

Answer:

```
ALTER TABLE Employee  
ADD Dept_ID INTEGER;
```

3. **List the steps you would follow to install the MySQL server on a Windows computer to practice SQL commands.**

Answer:

- (a) Download the MySQL Installer from the official MySQL website.
- (b) Run the installer executable file.
- (c) Choose the 'Server only' setup type to install just the database server.
- (d) Follow the on-screen prompts to check for requirements and execute the installation.
- (e) During configuration, set the root password for the server. This is a very important step.
- (f) Configure the Windows Service to allow MySQL to run automatically.
- (g) Complete the installation. You can then access the server using the MySQL Command Line Client.

4. **You have inserted several records into your 'Employee' table. Now, your company has decided to give a 10% salary increase to all employees. Write the SQL 'UPDATE' command to reflect this change in the database.**

Answer:

```
UPDATE Employee  
SET Salary = Salary * 1.10;
```

5. **Describe the steps to create a new database named 'Company' and then make it the active database to start creating tables within it using MySQL Command Line.**

Answer:

- (a) **Step 1: Open the Command Line Client.** Start the MySQL 5.6 Command Line Client from the Start Menu and enter your root password when prompted.
- (b) **Step 2: Create the Database.** At the 'mysql>' prompt, type the command to create the new database and press Enter. The command is: `CREATE DATABASE Company;`
- (c) **Step 3: Make the Database Active.** To start working inside this new database, you must select it. Type the command: `USE Company;`
- (d) **Step 4: Verify (Optional).** The command line will show 'Database changed'. You can now start using 'CREATE TABLE' and other commands, which will apply to the 'Company' database.