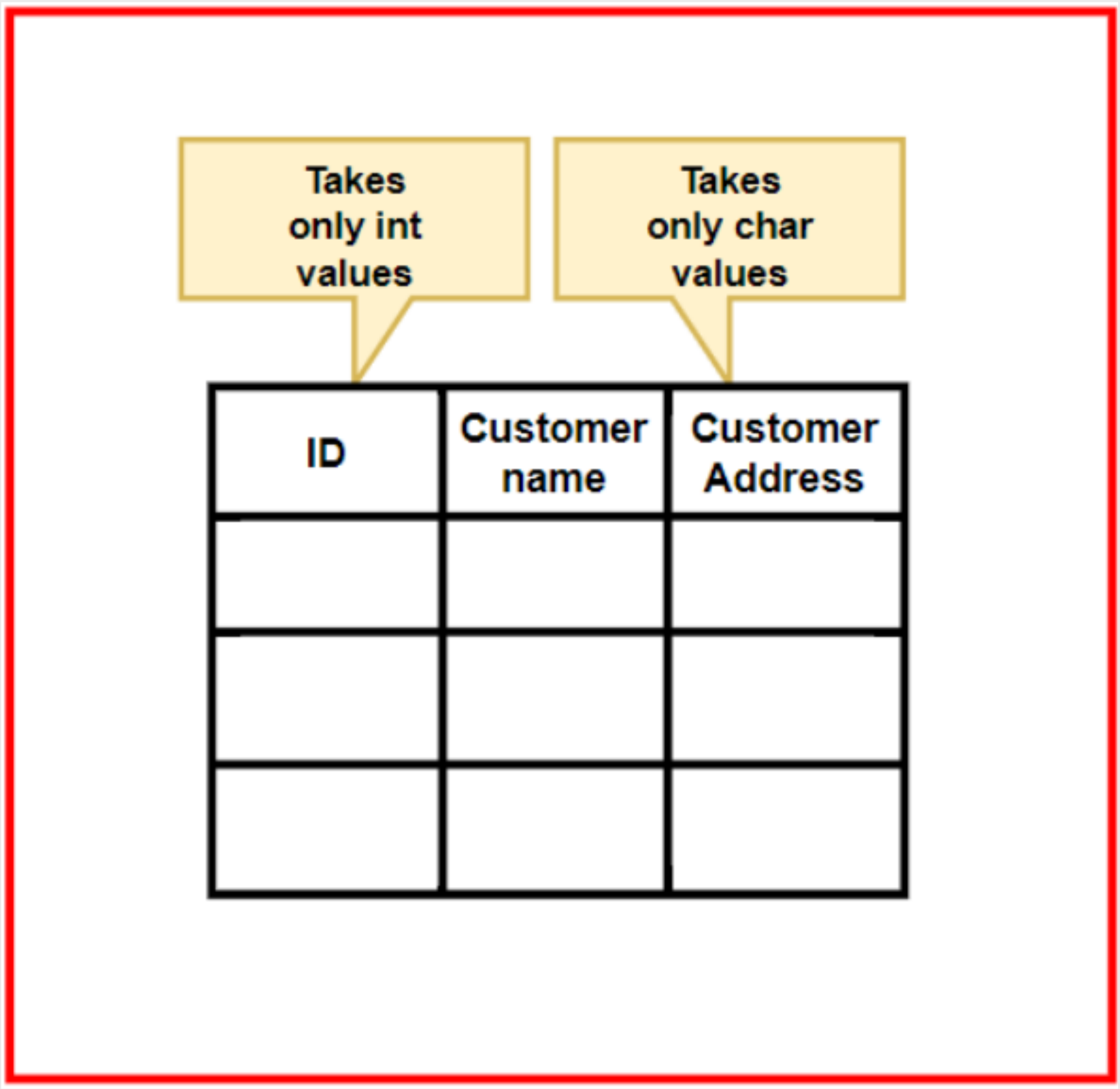
Data Architecture

A schema is a logical container or structure that organises and defines the structure of a database.
It defines how data is organised, what data types are used, what constraints are applied, and the relationships between different pieces of data. A schema acts as a blueprint for the database, ensuring data integrity, consistency, and efficient data retrieval.



Takes only int values

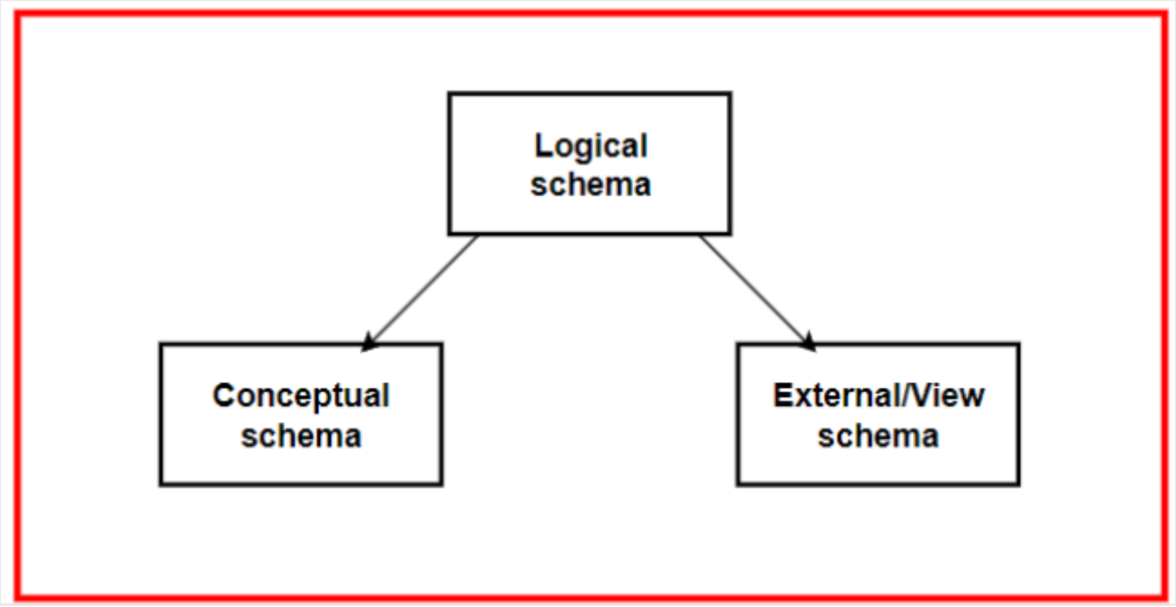
Takes only char values

ID	Customer name	Customer Address

An illustration of the customer database schema is shown in the picture. In this case, we choose to have three rows and three columns. Every row item in the database will have a unique column ID, which can only contain integer values. Only char values are accepted as input for the customer's name and address. This contributes to maintaining data consistency, data integrity (as no other type of value can be kept in the database), and faster data retrieval through the use of unique keys.

Types of Schema

- **Physical Schema**:A physical schema defines how data is stored on the underlying hardware, including details such as storage format, file organisation, indexing methods, and data placement.
  - **Characteristics of Physical Schema:**
    - Its primary focus lies in enhancing the storage and retrieval of data to boost performance.
    - Modifications made to the physical schema demand meticulous planning and can potentially affect the overall performance of the database.
  - **Example:** Deciding to use clustered indexes on specific columns for faster retrieval.
- **Logical Schema**:A logical schema defines the database's structure from a logical or conceptual perspective, without considering how the data is physically stored.



Types of Logical Schema

- **Conceptual Schema:**The conceptual schema represents the overall view of the entire database. It defines the high-level structure and relationships between all data elements.
  For example: Consider a university database with entities such as Student (StudentID, Name, Address, DateOfBirth), Course (CourseID, CourseName, Credits), and Department (DepartmentID, DepartmentName, OfficePhone). Relationships among them are:
  - Students can enrol in multiple Courses.
  - Each Course can be taken by multiple students.
  - Each Course is offered by one Department, but a Department can offer multiple Courses.
- The focus is on what the data represents (students, courses, departments) and how these entities are interrelated through

relationships like enrollment and offerings.

- **External/View Schema:**An external schema defines the user-specific views of the database. It focuses on the portions of the database that are relevant to specific user roles or applications.
  For example: In a university database, for a Student's Portal, the view will have StudentProfile (StudentID, Name, Address, CoursesEnrolled). This view provides a student with access to their data and their course enrollments, but not to other students' information or course details like credits or department.

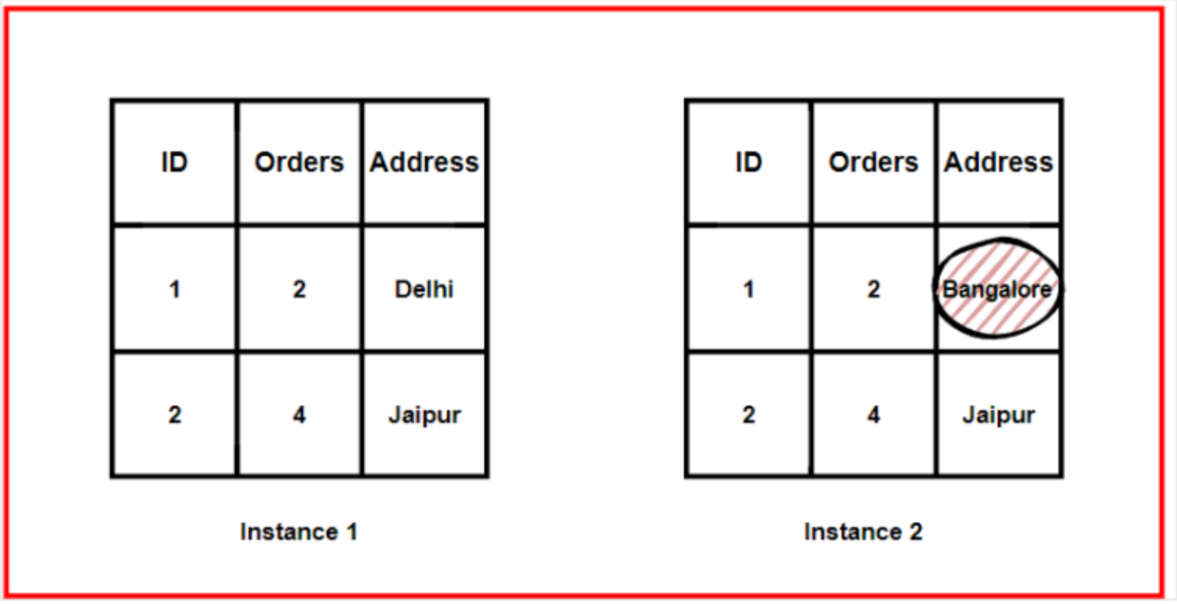
Characteristics of Logical Schema

- It delineates how data is structured into tables, the interconnections between these tables, and the restrictions placed on the data.
- Logical schemas prioritise data modelling and database design over considerations related to hardware or storage specifics.
- **Example:** Defining tables, specifying primary and foreign keys, and creating views for data access.

Instance

The information residing within a database at a specific point in time is referred to as the database’s "instance."

Within a given database schema, the declarations of variables within its tables pertain to that specific database. The term "instance" in this context denotes the current values of these variables at a particular moment in time for that database.



ID	Orders	Address
1	2	Delhi
2	4	Jaipur

Instance 1

ID	Orders	Address
1	2	Bangalore
2	4	Jaipur

Instance 2

The figure depicts the customer database in different instances. In instance 1, the customer with ID 1’s address is Delhi; however, it gets updated to Bangalore. Hence, in instance 2, we get a different value. Instances help in accommodating the updates taking place in the database.