|  |  |  |
| --- | --- | --- |
| **KEY TYPE** | **USAGES** | **WHEN TO USE** |
| **Primary** | Identifies each record in a table. It ensures that there are no duplicate records, and it’s important for data integrity | When you need a unique identifier for each record in a table. |
| **Secondary** | Provides another way to identify records. Unlike the primary key, it may allow duplicates but is still used for efficient data retrieval. | When you need to retrieve records based on a non-primary identifier. |
| **Foreign** | Links two tables by referencing the primary key of another table. It enforces referential integrity. | When creating relationships between tables. |
| **Composite** | The combination of multiple columns is needed to uniquely identify a record | When a combination of columns is needed for uniqueness. |
| **Super** | Any combination of columns that uniquely identifies a record is a super key | When uniquely identify a record (It is more of a theoretical concept). |
| **Candidate** | Is a minimal super key and is a potential choice for the primary key | When determining the potential primary key columns. |

**When can a key serve multiple roles?**

When the key has characteristics that meet distinct requirements. For instance, a composite key might function as both a primary and secondary key, uniquely identifying records within a table while also facilitating efficient data retrieval. This double functionality often happens in complex scenarios.

|  |  |
| --- | --- |
| **Foreign key** | **Secondary key** |
| Makes a relationship between two tables | Provides another way to identify records and improve query performance |
| Values don’t need to be unique | Values may or may not be unique |
| Enforces referential integrity | Doesn’t enforce relationships between tables |
| References another table’s primary key | Doesn’t reference another table’s primary key |
| Used for creating and maintaining relationships between tables | Used to speed up SELECT queries by providing alternative paths for data retrieval. |

**Conclusion**

Keys play a fundamental role in ensuring data integrity, establishing relationships between tables, and optimizing query performance. Understanding the different types of keys and their usages is important for designing a strong and efficient database schema. Each key type has its specific purpose, and the choice of which key to use depends on the requirements or needs of the database and the relationships between its tables.