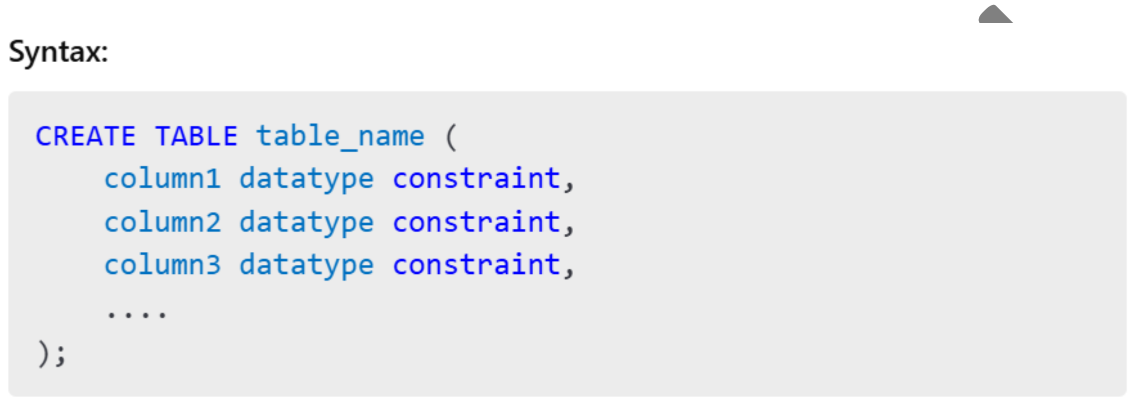
SQL Constraints

SQL constraints are used to specify rules for data in a table.

SQL Create Constraints

Constraints can be specified when the table is created using the `CREATE TABLE` statement or added later using the `ALTER TABLE` statement.



Syntax:

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

SQL Constraints Overview

SQL constraints define rules for the data stored in a table. These constraints help maintain data accuracy and integrity. If a data action violates a constraint, the action is aborted.

Constraints can be applied at two levels:

- **Column Level**: Applies to a specific column.
- **Table Level**: Applies to the entire table.

The most commonly used constraints in SQL are:

- **NOT NULL**: Ensures that a column cannot have a `NULL` value.
- **UNIQUE**: Ensures that all values in a column are distinct.
- **PRIMARY KEY**: A combination of `NOT NULL` and `UNIQUE`. Uniquely identifies each row in a table.
- **FOREIGN KEY**: Prevents actions that would destroy links between tables by enforcing referential integrity.
- **CHECK**: Ensures that the values in a column satisfy a specific condition.
- **DEFAULT**: Assigns a default value to a column when no value is specified.
- **CREATE INDEX**: Speeds up data retrieval by creating an index on one or more columns.

By implementing these constraints, database systems ensure data consistency, reliability, and enforce business rules effectively.