DDL

Data Definition Language

Aitrich

INTRODUCTION

DDL stands for Data Definition Language, and it is a subset of SQL

It is used to define and manage the structure of a database.

DDL statements are responsible for creating, altering, and deleting database objects such as tables, views, indexes, and constraints.

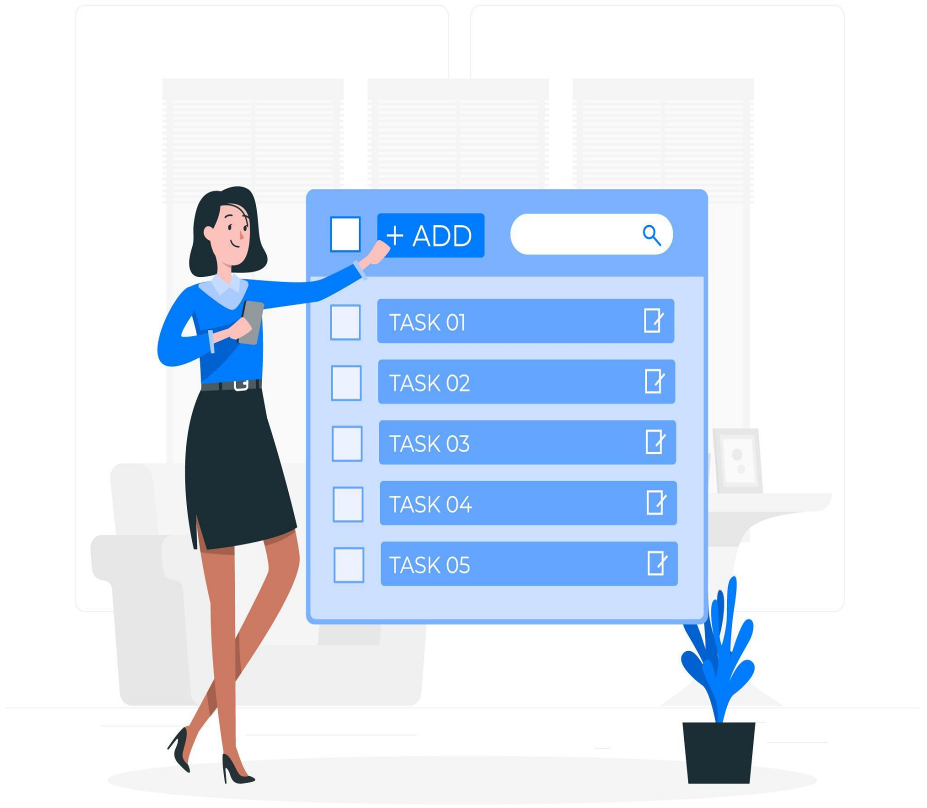
The primary purpose of DDL in SQL is to specify the schema or structure of the database and its objects.

LIST OF DDL COMMANDS

- CREATE
- ALTER
- DROP
- TRUNCATE
- COMMENT
- RENAME

1. CREATE Statement:

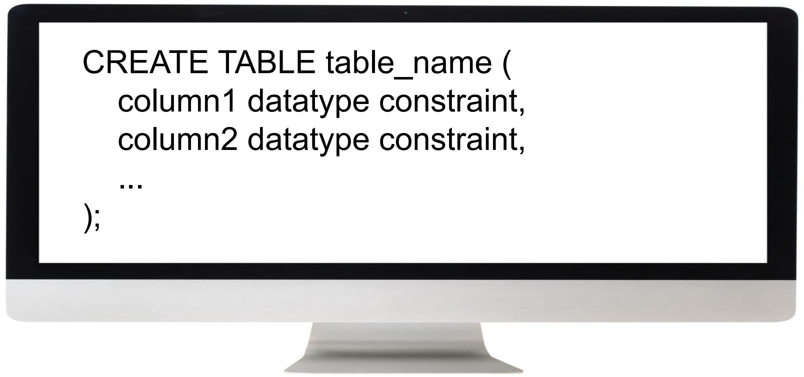
The CREATE statement is used to create new database objects such as tables, views, indexes, and constraints. It defines the structure of the object, including column names, data types, and other properties.



a) CREATE Table:

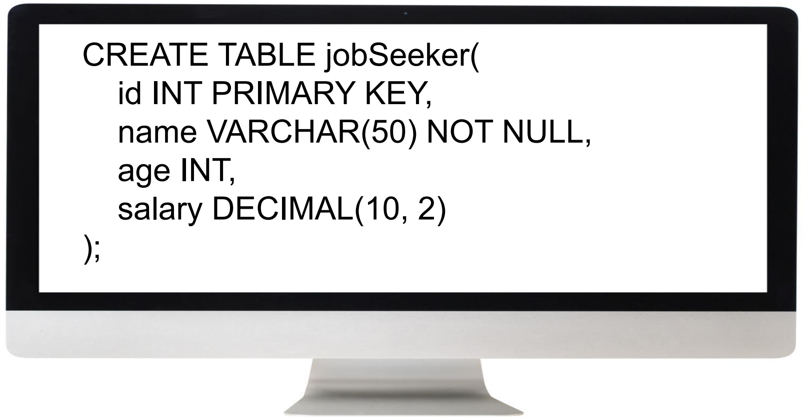
It creates a new table with the specified name and defines the columns, data types, and constraints.

Syntax:



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

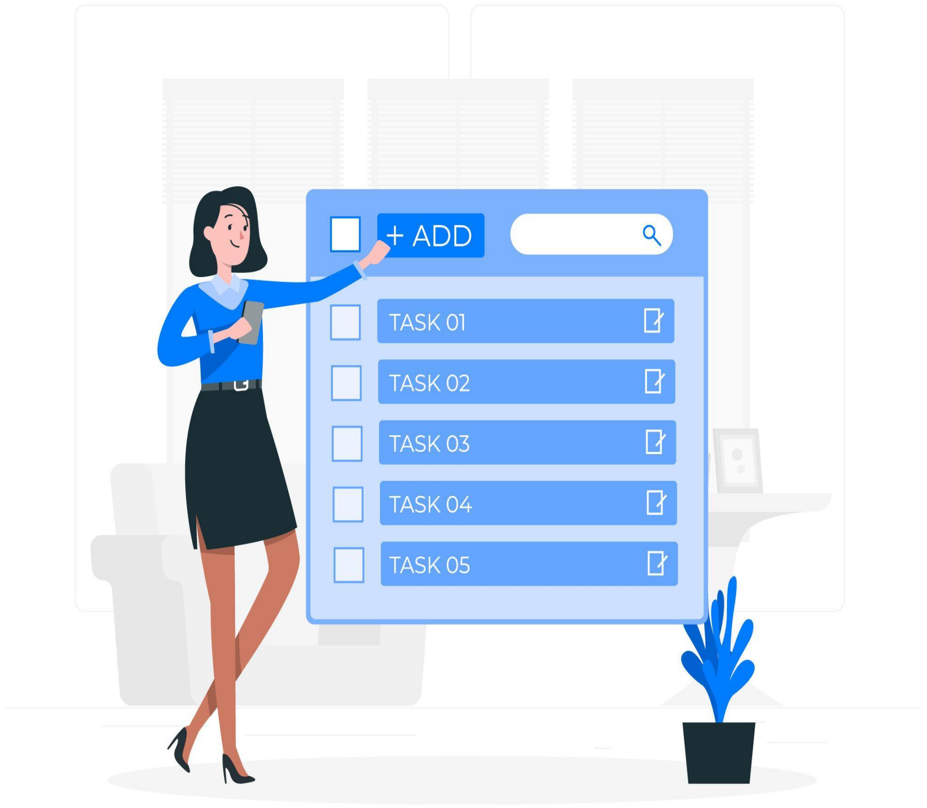
Example:



```
CREATE TABLE jobSeeker(
   id INT PRIMARY KEY,
   name VARCHAR(50) NOT NULL,
   age INT,
   salary DECIMAL(10, 2)
);
```

2. Alter Statement:

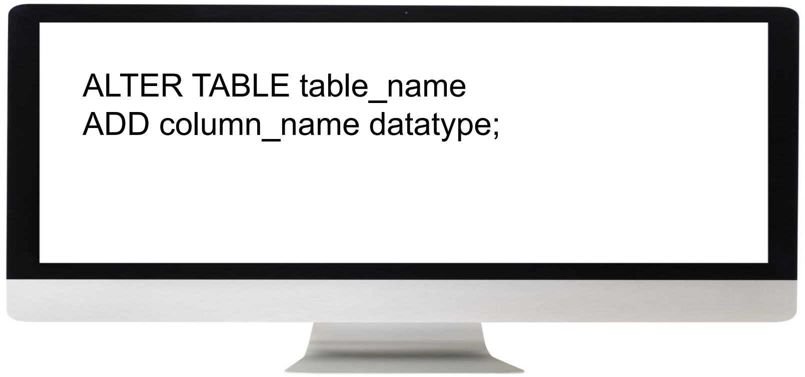
The ALTER statement is used to modify the structure of existing database objects.



a) ALTER Table:

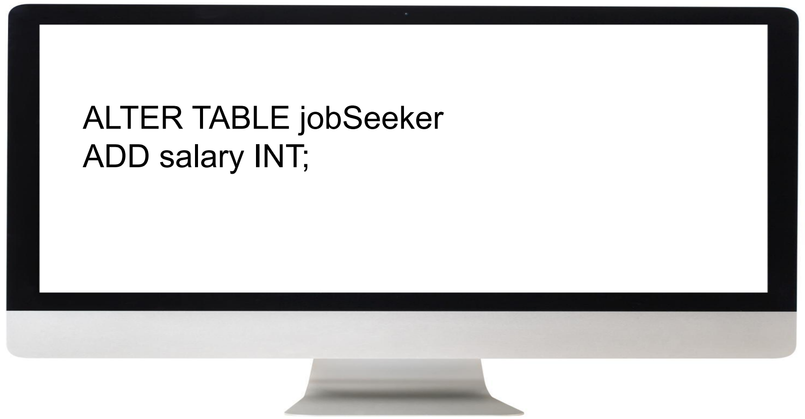
The ALTER statement is used to modify the structure of existing database objects. It can be used to add or drop columns, modify data types, rename objects, or add constraints.

Syntax



```
ALTER TABLE table_name
ADD column_name datatype;
```

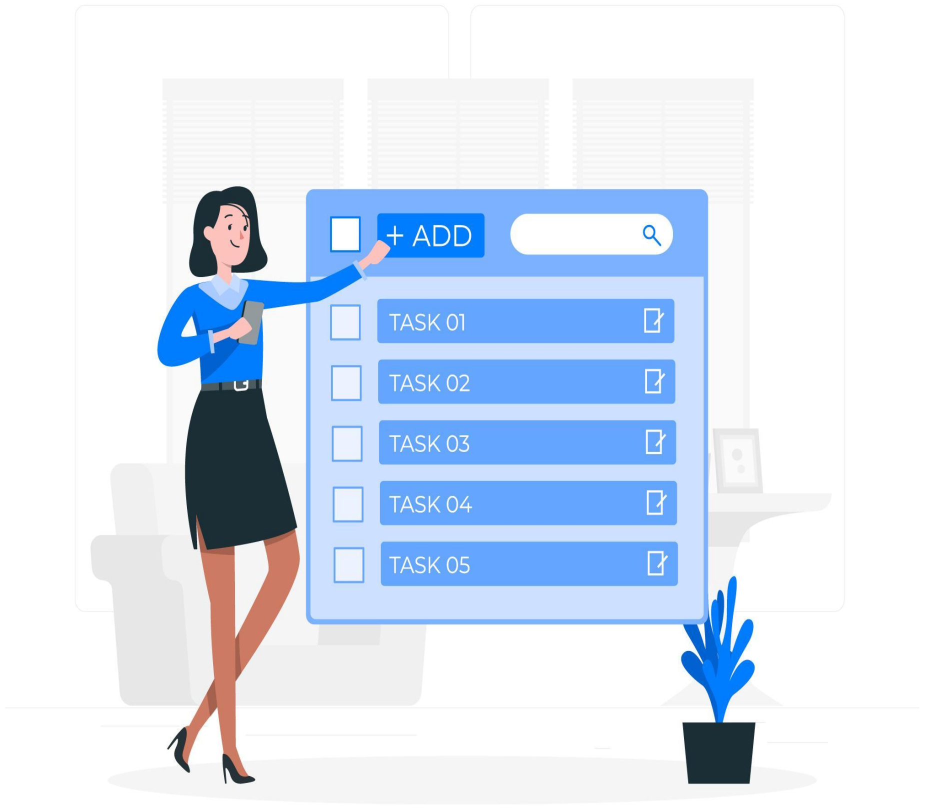
Example:



```
ALTER TABLE jobSeeker
ADD salary INT;
```

3. Drop Statement:

The DROP statement is used to delete database objects from the database. Here are some common uses:



a) DROP Table:

DROP TABLE: This statement is used to delete an entire table from the database. The syntax is as follows:

Syntax

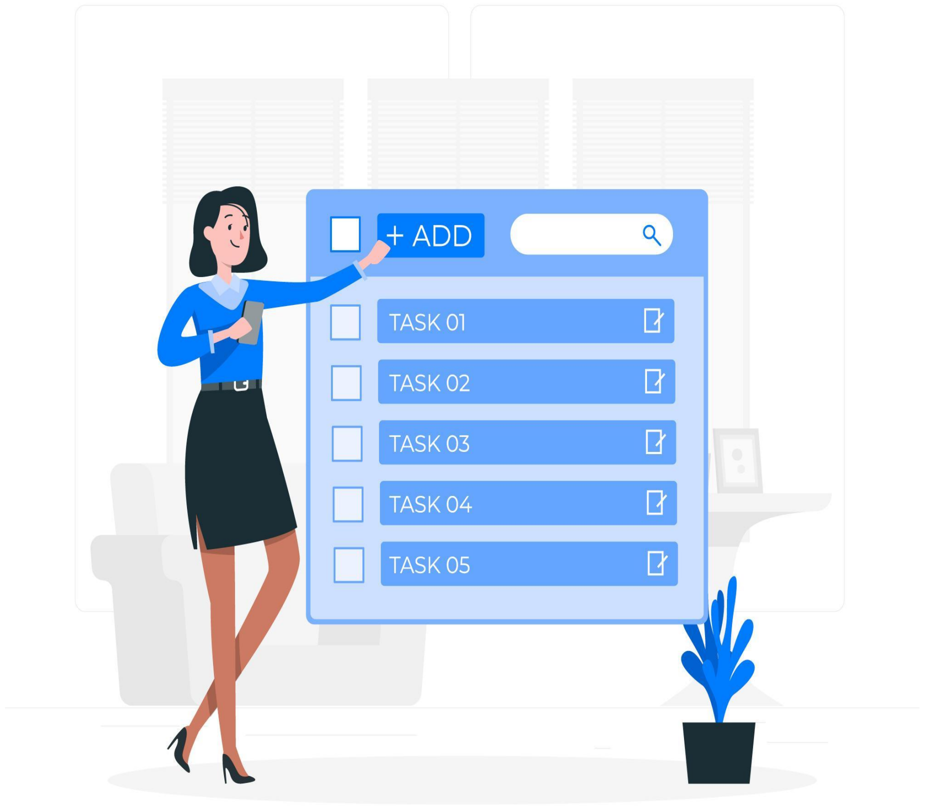
```
DROP TABLE table_name;
```

Example:

```
DROP TABLE jobSeeker;
```

4. RENAME Statement:

The RENAME statement is used to rename database objects.



a) RENAME Table:

This syntax renames an existing table.

Syntax

```
EXEC sp_rename 'old_table_name',
'new_table_name'
```

Example:

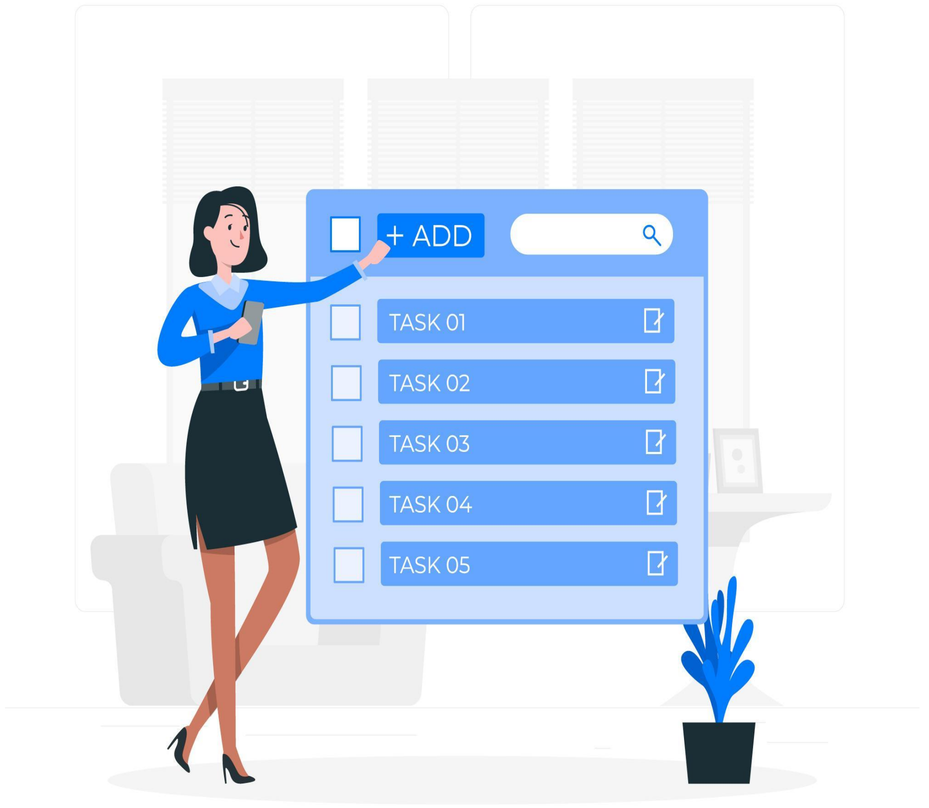
```
EXEC sp_rename 'JobSeeker', 'Jobseekers'
```

CONSTRAINTS

Constraints in SQL are rules or conditions applied to the columns or tables to enforce data integrity and ensure that the data meets certain requirements.

Constraints help maintain the accuracy, consistency, and validity of the data stored in a database.

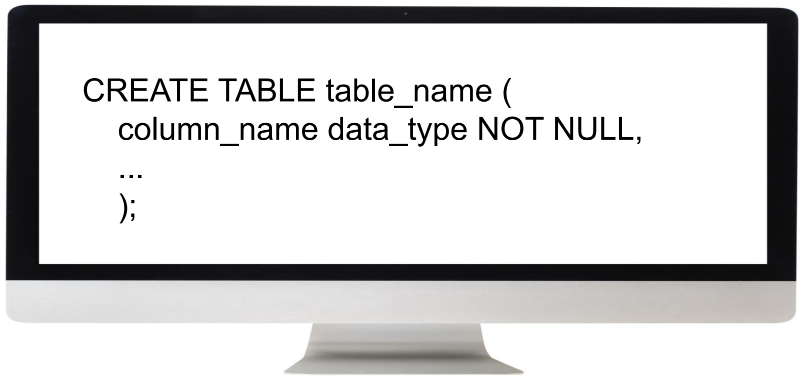
Here are the commonly used constraints in SQL:



1. NOT NULL Constraint:

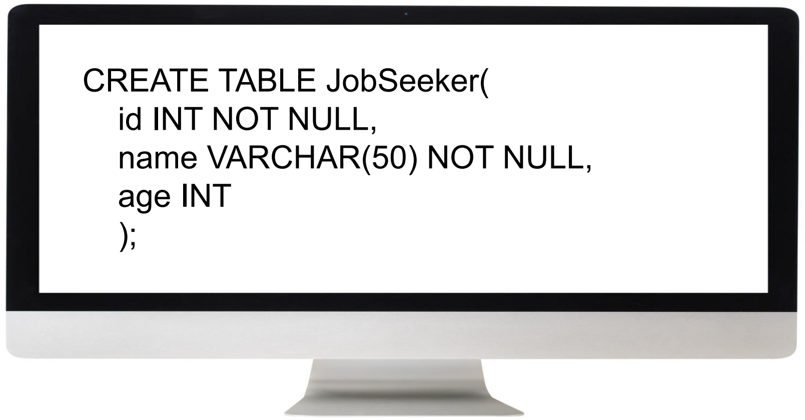
The NOT NULL constraint ensures that a column cannot contain NULL values, meaning it must always have a value.

Syntax



```
CREATE TABLE table_name (
    column_name data_type NOT NULL,
    ...
    );
```

Example:

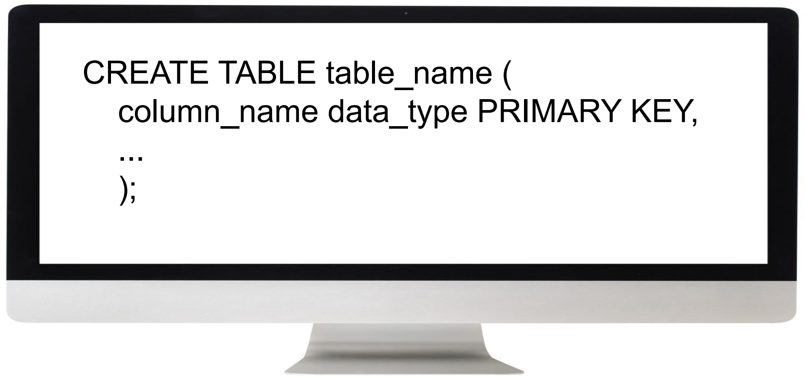


```
CREATE TABLE JobSeeker(
    id INT NOT NULL,
    name VARCHAR(50) NOT NULL,
    age INT
    );
```

2. PRIMARY KEY Constraint:

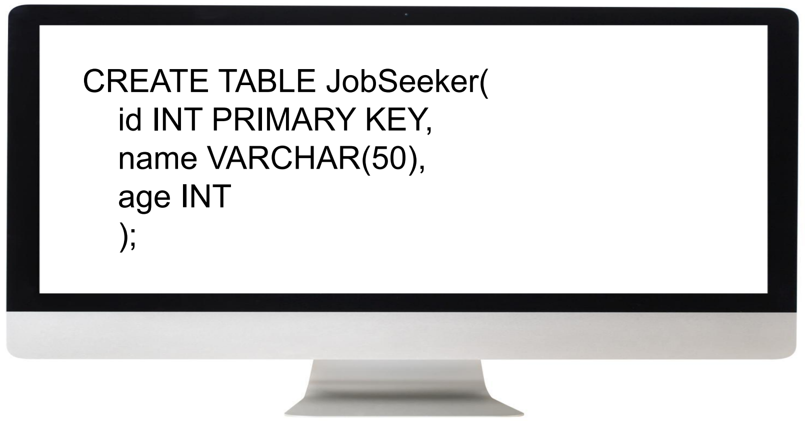
The PRIMARY KEY constraint uniquely identifies each record in a table. It ensures that the specified column or combination of columns has unique values and cannot contain NULL

Syntax



```
CREATE TABLE table_name (
   column_name data_type PRIMARY KEY,
   ...
   );
```

Example:

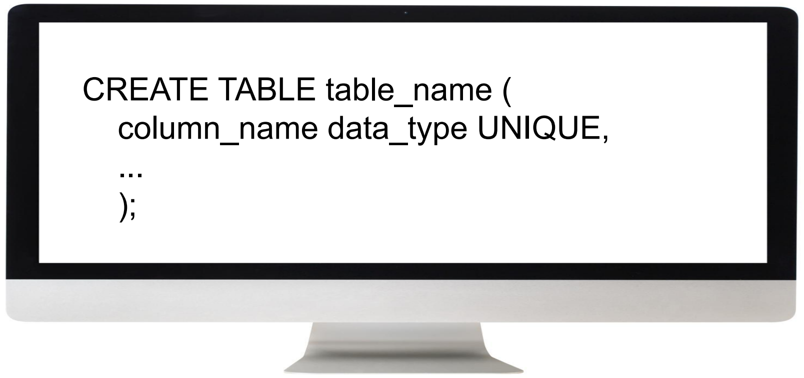


```
CREATE TABLE JobSeeker(
   id INT PRIMARY KEY,
   name VARCHAR(50),
   age INT
   );
```

3. UNIQUE Constraint:

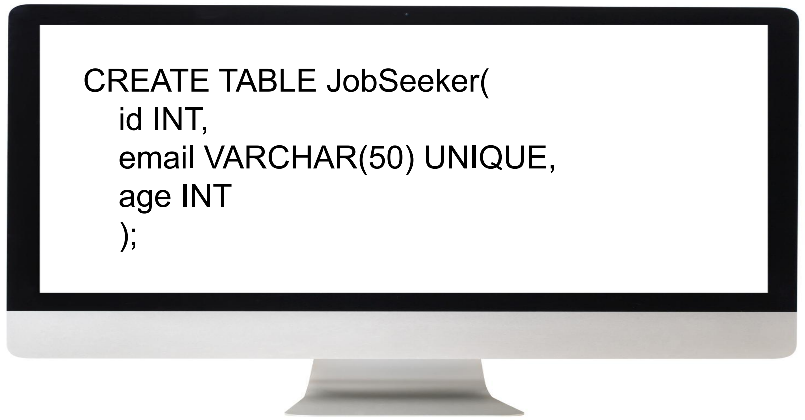
The UNIQUE constraint ensures that the values in a column or a combination of columns are unique, meaning no duplicate values are allowed.

Syntax



```
CREATE TABLE table_name (
    column_name data_type UNIQUE,
    ...
    );
```

Example:

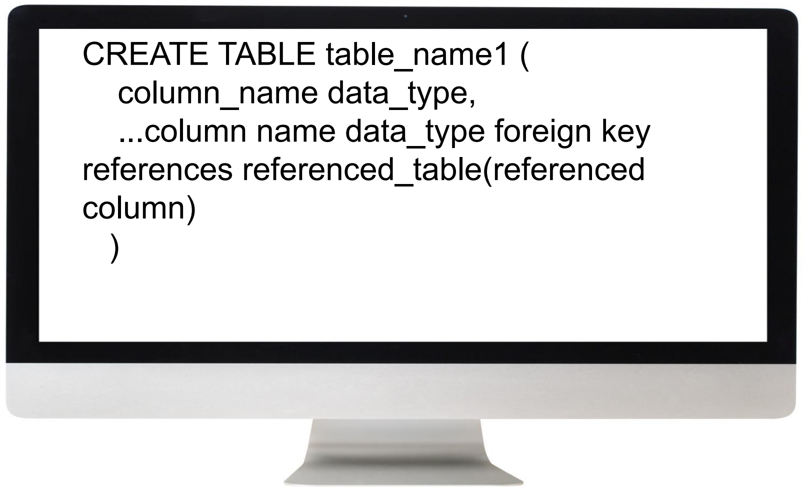


```
CREATE TABLE JobSeeker(
    id INT,
    email VARCHAR(50) UNIQUE,
    age INT
    );
```

4. FOREIGN KEY Constraint

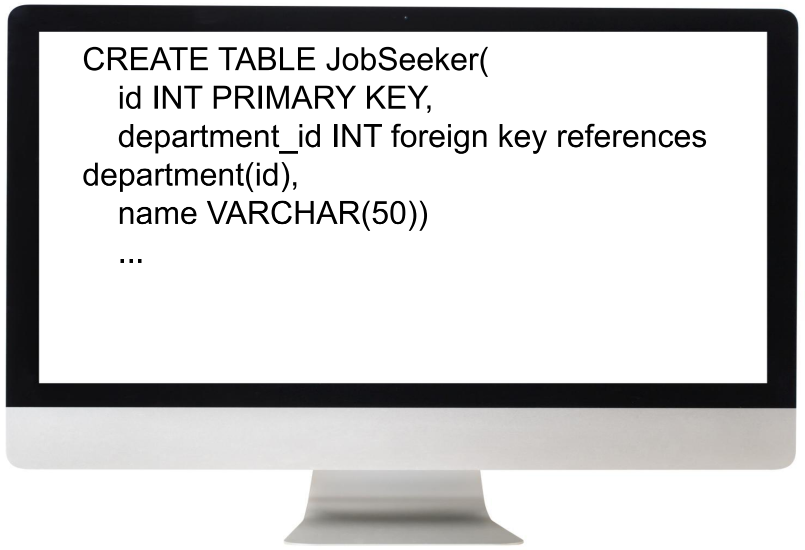
The FOREIGN KEY constraint establishes a relationship between two tables by referencing the primary key of one table in another table. It ensures referential integrity, meaning the values in the foreign key column must match the values in the referenced primary key column.

Syntax



```
CREATE TABLE table_name1 (
   column_name data_type,
   ...column name data_type foreign key
references referenced_table(referenced
column)
  )
```

Example:

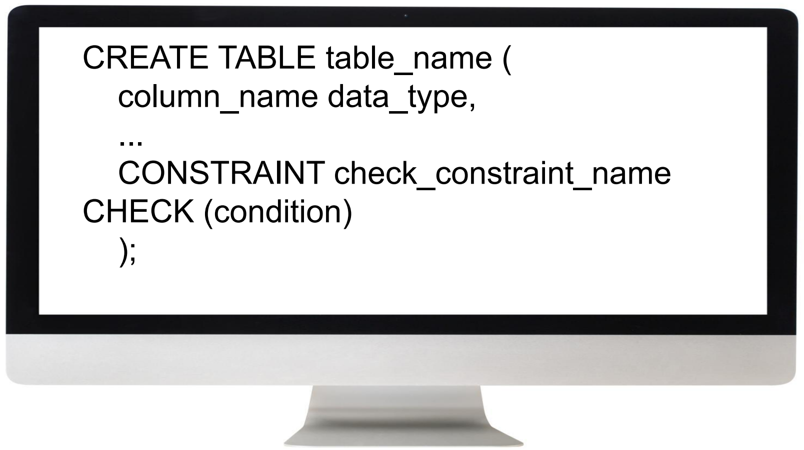


```
CREATE TABLE JobSeeker(
   id INT PRIMARY KEY,
   department_id INT foreign key references
department(id),
   name VARCHAR(50))
   ...
```

5. CHECK Constraint

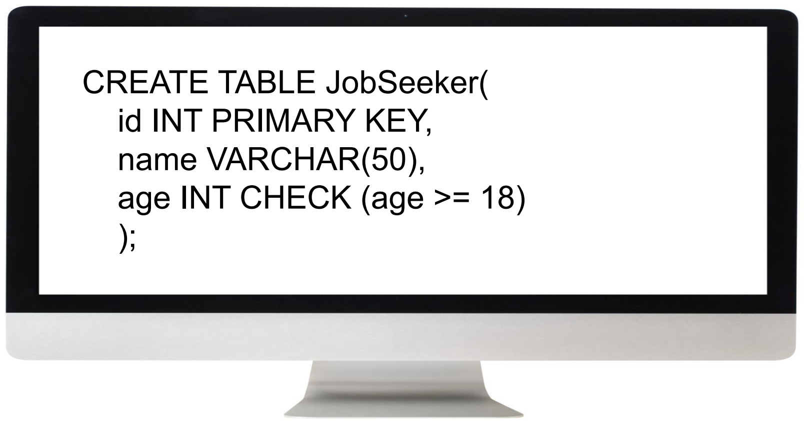
The CHECK constraint defines a condition that must be satisfied by the values in a column. It allows you to specify a logical expression to restrict the range of valid values.

Syntax



```
CREATE TABLE table_name (
   column_name data_type,
   ...
   CONSTRAINT check_constraint_name
CHECK (condition)
   );
```

Example:



```
CREATE TABLE JobSeeker(
   id INT PRIMARY KEY,
   name VARCHAR(50),
   age INT CHECK (age >= 18)
   );
```

thank you