LAB - Primary Key Constraint

In this lab, you will learn how to use the SQL Server `PRIMARY KEY` constraint to create a primary key for a table.

A primary key is a column or a group of columns that uniquely identifies each row in a table. You create a primary key for a table by using the `PRIMARY KEY` constraint.

If the primary key consists of only one column, you can define use `PRIMARY KEY` constraint as a column constraint:

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

In case the primary key has two or more columns, you must use the `PRIMARY KEY` constraint as a table constraint:

```
CREATE TABLE table_name (
    pk_column_1 data_type,
    pk_column_2 data type,
    ...
    PRIMARY KEY (pk_column_1, pk_column_2)
);
```

Each table can contain only one primary key. All columns that participate in the primary key must be defined as `NOT NULL`. SQL Server automatically sets the `NOT NULL` constraint for all the primary key columns if the `NOT NULL` constraint is not specified for these columns.

SQL Server also automatically creates a unique clustered index (or a non-clustered index if specified as such) when you create a primary key.

Examples

The following example creates a table with a primary key that consists of one column:

```
CREATE TABLE sales.activities (
    activity_id INT PRIMARY KEY IDENTITY,
    activity_name VARCHAR (255) NOT NULL,
    activity_date DATE NOT NULL
);
```

In this `sales.activities` table, the `activity_id` column is the primary key column. It means the `activity_id` column contains unique values.

The `IDENTITY` property is used for the `activity_id` column to automatically generate unique integer values.

The following statement creates a new table named `sales.participants` whose primary key consists of two columns:

```
CREATE TABLE sales.participants(
    activity_id int,
    customer_id int,
    PRIMARY KEY(activity_id, customer_id)
);
```

In this example, the values in either `activity_id` or `customer_id` column can be duplicate, but each combination of values from both columns must be unique.

Typically, a table always has a primary key defined at the time of creation. However, sometimes, an existing table may not have a primary key defined. In this case, you can add a primary key to the table by using the `ALTER TABLE` statement. Consider the following example:

The following statement creates a table without a primary key:

```
CREATE TABLE sales.events(
    event_id INT NOT NULL,
    event_name VARCHAR(255),
    start_date DATE NOT NULL,
    duration DEC(5,2)
);
```

To make the `event_id` column as the primary key, you use the following `ALTER TABLE` statement:

```
ALTER TABLE sales.events
ADD PRIMARY KEY(event_id);
```

Note that if the `sales.events` table already has data, before promoting the `event_id` column as the primary key, you must ensure that the values in the `event_id` are unique.

In this lab, you have learned how to use the SQL Server `PRIMARY KEY` constraint to create a primary key for a table.