CREATE AND DROP TABLE

Reading: Examples to CREATE and DROP tables (5 mins)

Objective(s)

At the end of this lab you will be able to:

- Create and Drop tables in the database

Effort: 5 min

Here we will look at some examples to create and drop tables. In the previous video we saw the general syntax to create a table: create table TABLENAME (COLUMN1 datatype, COLUMN2 datatype, COLUMN3 datatype, ...);

Therefore to create a table called TEST with two columns - ID of type integer, and Name of type varchar, we could create it using the following SQL statement:

```
create table TEST (
    ID integer,
    NAME varchar(30)
    );
```

Now let's create a table called COUNTRY with an ID column, a two letter country code column, and a variable length country name column:

```
create table COUNTRY (
    ID int,
    CCODE char(2),
    NAME varchar(60)
    );
```

Sometimes you may see additional keywords in a create table statement:

```
create table COUNTRY (
    ID int NOT NULL,
    CCODE char(2),
    NAME varchar(60),
    PRIMARY KEY (ID)
    );
```

In the above example the ID column has the **NOT NULL** constraint added after the datatype - meaning that it cannot contain a NULL or an empty value. If you look at the last row in the create table statement above you will note that we are using ID as a Primary Key and the database does not allow Primary Keys to have NULL values. A Primary Key is a unique identifier in a table, and using Primary Keys can help speed up your queries significantly. If the table you are trying to create already exists in the database, you will get an error indicating **table XXX.YYY already exists**. To circumvent this error, either create a table with a different name or first DROP the existing table. It is quite common to issue a DROP before doing a CREATE in test and development scenarios. Here is an example:

```
1  drop table COUNTRY;
2  create table COUNTRY (
3      ID integer PRIMARY KEY NOT NULL,
4      CCODE char(2),
5      NAME varchar(60)
6      );
```

WARNING: before dropping a table ensure that it doesn't contain important data that can't be recovered easily. Note that if the table does not already exist and you try to drop it, you will see an error like **XXX.YYY is an undefined name**. You can ignore this error as long as the subsequent CREATE statement executed successfully.

In the lab later in this module you will practice creating tables and other SQL statements hands-on.