



# EAST TENNESSEE STATE UNIVERSITY

## CSCI 3020 – Database Advanced Concepts

### Project 2 – Extract-Transform-Load

#### Overview

This project allows you to create a process for extracting, transforming, and loading information into a relational database using Database Programming. You have an existing database schema, called BIKE\_SHOP (Relevant ERD provided). Your goal is to extract information, transform it into a secondary design (an archive), and load it into that archive without losing information.

#### Setup Tasks

1. Locate the SQL files for Project 2 in the Project 2 folder of the content section of D2L.
2. Run the file Project2\_Setup.sql to create a couple of tables within your schema that you will use during this Project Assignment.

#### Requirements

You must produce a working database from these specifications. You will do this in two phases.

#### Package BIKESHOP

Create a PL/SQL Package and Package Body called BIKESHOP. **Save the script to a file called Project2\_BIKESHOP.sql** (a skeleton file is provided). This package will extract information from the BIKE\_SHOP schema on the CSDB database, and insert the data into a secondary database format. Following this, you will have the following PROCEDURES (and one FUNCTION) defined in your BIKESHOP package:

1. EXTRACT\_BICYCLES – A procedure that will pull bicycle information from the database.
2. EXTRACT\_CUSTOMERS – A procedure that will pull customer information from the database.
3. CUSTOMER\_BIKES – A function that returns all bikes purchased by an individual customer.
4. ARCHIVE\_CUSTOMER\_BIKES – A procedure that archives all bike purchases into the CUSTOMER\_BIKE table.

#### PROCEDURE EXTRACT\_BICYCLES

Within the BIKESHOP package, create the EXTRACT\_BICYCLES procedure. The procedure must accept a parameter, OUTPUT\_TYPE, that holds the value 'D' if the procedure should output data to the database, or 'S' if the procedure should output a report to the console screen.

1. The procedure should pull the following information from the BIKE\_SHOP.BICYCLE table: SERIALNUMBER, MODELTYPE, PAINTID, FRAMESIZE, ORDERDATE, STARTDATE, SHIPDATE, CONSTRUCTION, LISTPRICE, SALEPRICE, SALESTAX, SALESTATE.
2. If the OUTPUT\_TYPE is 'D', insert the data into the BICYCLES table within your own schema. This table was created by the Project2\_Setup.sql script. Be sure to COMMIT the transaction within the procedure. Output the number of records inserted to the console screen.
3. If the OUTPUT\_TYPE is 'S', display the information to the screen in a well-formatted way. This means you should organize the information and present it to the user in a way that is easy to understand. Make sure you sort the records by ORDERDATE (ascending).

### Procedure `EXTRACT_CUSTOMERS`

Within the `BIKESHOP` package, create the `EXTRACT_CUSTOMERS` procedure. The procedure must accept a parameter, `OUTPUT_TYPE`, that holds the value 'D' if the procedure should output data to the database, or 'S' if the procedure should output a report to the console screen.

1. The procedure should pull the following information from the `BIKE_SHOP.CUSTOMER` table, and associated tables: `CUSTOMERID`, `LASTNAME`, `FIRSTNAME`, `PHONE`, `ADDRESS`, `CITY`, `STATE`, `ZIPCODE`.
2. If the `OUTPUT_TYPE` is 'D', insert the data into the `CUSTOMERS` table within your own schema. Be sure to `COMMIT` the transaction within the procedure. This table was created by the `Project2_Setup.sql` script. Output the number of records inserted to the console screen.
3. If the `OUTPUT_TYPE` is 'S', display the information to the screen in a well-formatted way. This means you should organize the information and present it to the user in a way that is easy to understand. Make sure you sort the records by `LASTNAME`, `FIRSTNAME` (ascending).

### Function `CUSTOMER_BIKES`

Within the `BIKESHOP` package, create the `CUSTOMER_BIKES` function. The function accepts a single parameter, `CUSTOMERID`, and will return a `TABLE` that contains `OBJECT`s representing the `BICYCLES` purchased by that particular customer. You will need to add an `OBJECT` definition that contains the customer's ID and the bicycle's serial number. In addition, you will need to create a `TABLE` of those `OBJECT`s, to be returned by the function.

### Procedure `ARCHIVE_CUSTOMER_BIKES`

Within the `BIKESHOP` package, create the `ARCHIVE_CUSTOMER_BIKES` procedure.

1. The procedure must call the `CUSTOMER_BIKES` function to accomplish its task.
2. For all customers, archive their bicycle purchases in the `CUSTOMER_BIKE` table of your own schema, which was created by the `Project2_Setup.sql` script.
3. Be sure to `COMMIT` the transaction within the procedure.

### Testing

Write a second script, named `Project2_Testing.sql`, that tests this code.

1. Ensure that your `BIKESHOP` package works as expected. This means you should run all of the modules within the package, and check the behavior.
2. You should check to see whether your transformed data is accurate. Pay special attention to the data in the `CUSTOMER_BIKE` table. If your `ARCHIVE_CUSTOMER_BIKES` procedure worked properly, you should be able to compare the `CUSTOMER_BIKE` table to the actual tables in the `BIKE_SHOP` schema.

### Terms

- You may complete this project in **teams of two**.
- I will grade this project as if each person contributed equally. This will be my assumption unless you tell me otherwise. If contributions are uneven, I will speak with both parties.
- Only one person on the team should submit this project.
- Clearly list the names of both team members in all files.

## Submission

Submit materials the Project 2 Drop Box on D2L.

- ❑ **Project 2 Package Script**, that contains the PL/SQL code that creates the BIKESHOP package. Name this file Project2\_BIKESHOP.sql.
- ❑ **Project 2 Testing Script**, contains all of the necessary SQL statements for testing the BIKESHOP database. Name this file Project2\_Testing.sql.

This project is due at or before **Thursday, April 1<sup>st</sup>, 2021, 11:59 PM**.

## Grading

The following grading rubric will be used to check your work. Each checkbox will be used to determine a portion of your grade, so make sure you have completed each item.

	Project Grade Rubric
BIKESHOP Package (80%)	<ul style="list-style-type: none"><li>❑ EXTRACT_BICYCLES (20 points)</li><li>❑ EXTRACT_CUSTOMERS (20 points)</li><li>❑ CUSTOMER_BIKES (25 points)</li><li>❑ ARCHIVE_CUSTOMER_BIKES (15 points)</li></ul>
Testing Script (15%)	<ul style="list-style-type: none"><li>❑ The script includes tests each component (10 points)</li><li>❑ The script investigates the contents of the CUSTOMER_BIKE table (5 points)</li></ul>
The Script Runs without errors (5%)	<ul style="list-style-type: none"><li>❑ The creation and testing scripts run top to bottom without errors (except errors that result from a table not existing) (20 points)</li></ul>

For the purposes of this assignment, the following levels will be used:

		Point Values				
<b>A</b>	All requirements are met and the procedure or function works without any errors.	25	20	15	10	5
<b>B</b>	All or most requirements are met or the procedure has minor errors.	20	16	12	8	4
<b>C</b>	Some of the requirements are met and others are not; or, procedure or function has significant errors.	15	12	9	6	3
<b>D</b>	Few of the requirements are met; or, the procedure has major errors.	10	8	6	4	2
<b>F</b>	The procedure or function was attempted, but no requirements were met.	5	4	3	2	1
<b>Zero</b>	This component was not submitted.	0	0	0	0	0