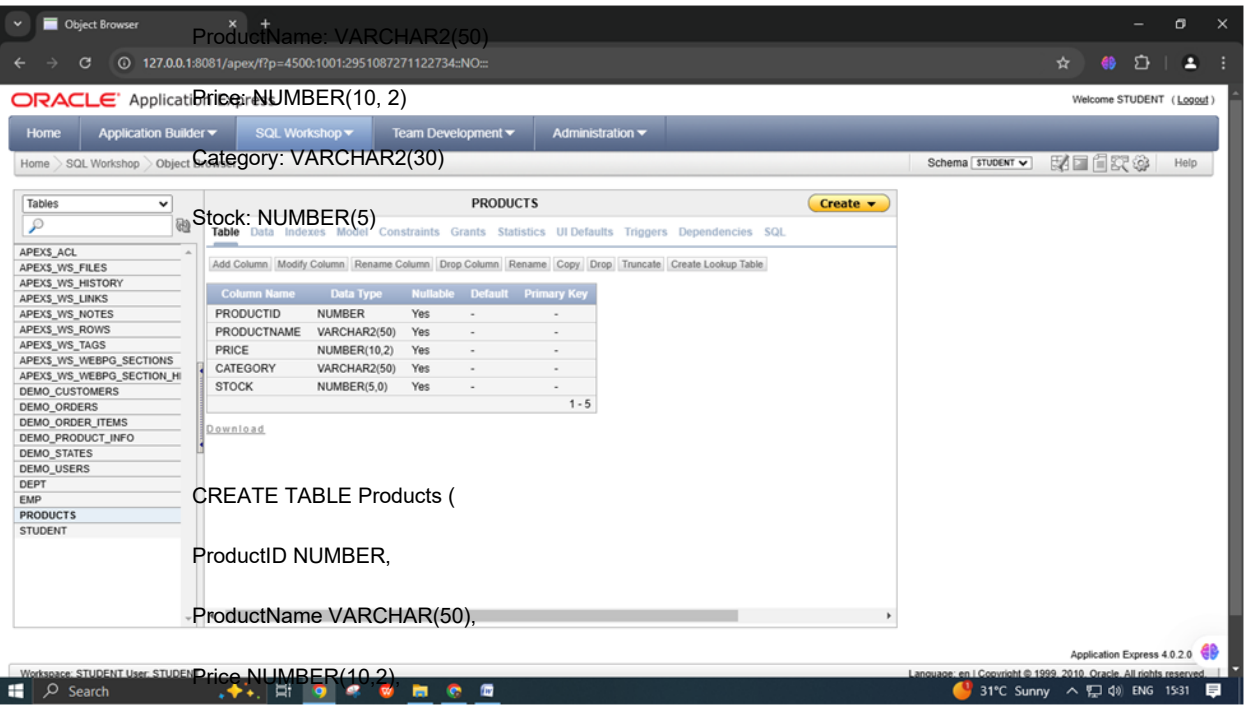
1. Create the Products Table

Create a table named Products with the following columns:

ProductID: NUMBER



ProductName: VARCHAR2(50)

Price: NUMBER(10, 2)

Category: VARCHAR2(30)

Stock: NUMBER(5)

CREATE TABLE Products (

ProductID NUMBER,

ProductName VARCHAR(50),

Price NUMBER(10,2),

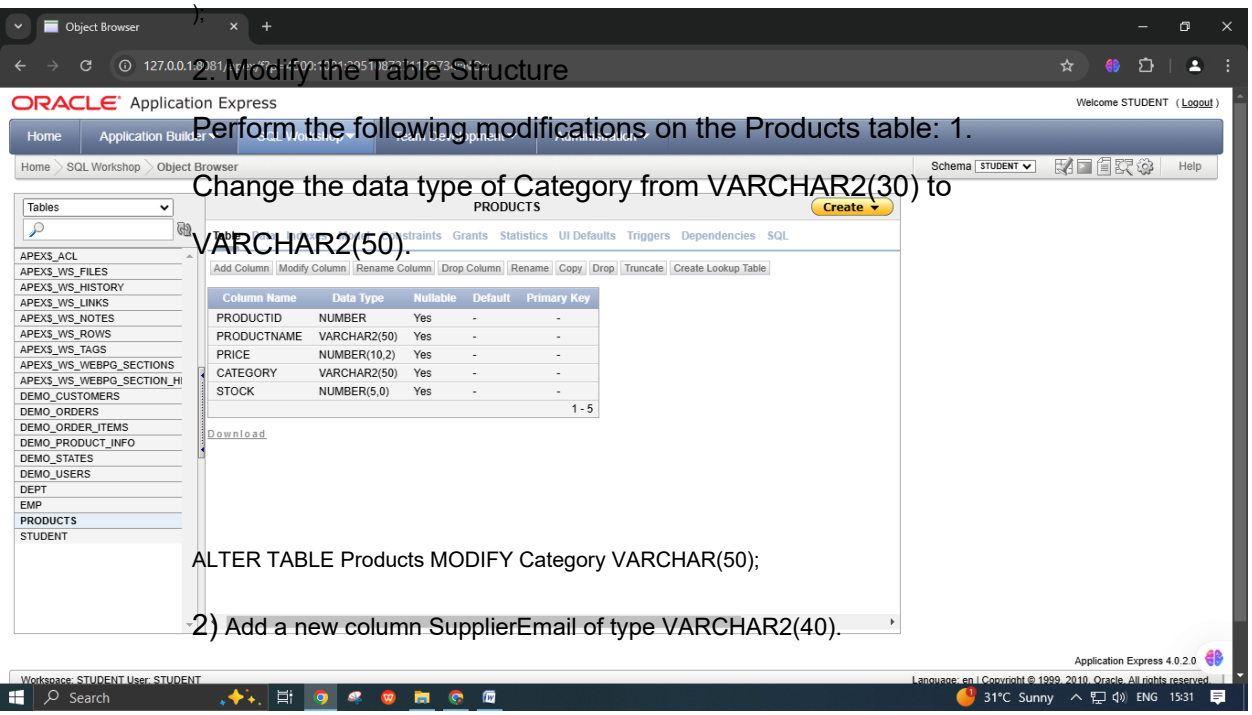
Category VARCHAR(30),

Stock NUMBER(5)

);

2. Modify the Table Structure

Perform the following modifications on the Products table: 1. Change the data type of Category from VARCHAR2(30) to VARCHAR2(50).



ALTER TABLE Products MODIFY Category VARCHAR(50);

2) Add a new column SupplierEmail of type VARCHAR2(40).

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTID	NUMBER	Yes	-	-
PRODUCTNAME	VARCHAR2(50)	Yes	-	-
PRICE	NUMBER(10,2)	Yes	-	-
CATEGORY	VARCHAR2(50)	Yes	-	-
STOCK	NUMBER(5,0)	Yes	-	-
SUPPLIEREMAIL	VARCHAR2(40)	Yes	-	-

1 - 6

ALTER TABLE Products ADD SupplierEmail VARCHAR2(40);

3) Drop the Stock column from the table.

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTID	NUMBER	Yes	-	-
PRODUCTNAME	VARCHAR2(50)	Yes	-	-
PRICE	NUMBER(10,2)	Yes	-	-
CATEGORY	VARCHAR2(50)	Yes	-	-
SUPPLIEREMAIL	VARCHAR2(40)	Yes	-	-

1 - 5

ALTER TABLE Products DROP COLUMN Stock;

4) Add a new column AddedDate with the data type DATE.

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTID	NUMBER	Yes	-	-
PRODUCTNAME	VARCHAR2(50)	Yes	-	-
PRICE	NUMBER(10,2)	Yes	-	-
CATEGORY	VARCHAR2(50)	Yes	-	-
SUPPLIEREMAIL	VARCHAR2(40)	Yes	-	-
ADDEDDATE	DATE	Yes	-	-

1 - 6

ALTER TABLE Products ADD AddedDate DATE;

3. Add Constraints

Add constraints to ensure data integrity:

1. Add a primary key constraint on the ProductID column (if not already added).

Column Name	Data Type	Nullable	Default	Primary Key
PRODUCTID	NUMBER	No	-	1
PRODUCTNAME	VARCHAR2(50)	Yes	-	-
PRICE	NUMBER(10,2)	Yes	-	-
CATEGORY	VARCHAR2(50)	Yes	-	-
SUPPLIEREMAIL	VARCHAR2(40)	Yes	-	-
ADDEDDATE	DATE	Yes	-	-

1 - 6

ALTER TABLE Products ADD CONSTRAINT PK_Products PRIMARY KEY (ProductID);

2. Add a unique constraint to the SupplierEmail column.

Constraint	Type	Search Condition	Related Constraint	Columns	Delete Rule	Status	Last Change	Index	Invalid
PK_PRODUCTS	Primary	-	-	PRODUCTID	-	ENABLED	12/16/2024 03:52:01 PM	PK_PRODUCTS	-
UQ_SUPPLIEREMAIL	Unique	-	-	SUPPLIEREMAIL	-	ENABLED	12/16/2024 03:55:31 PM	UQ_SUPPLIEREMAIL	-

1 - 2

ALTER TABLE Products ADD CONSTRAINT UQ_SupplierEmail UNIQUE (SupplierEmail);

4. Populate and Explore Deleting/Truncating Tables

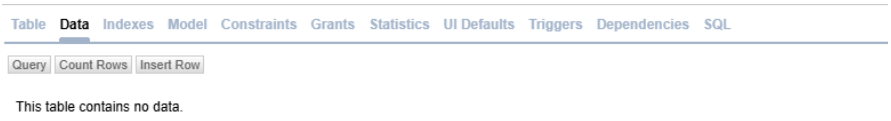
Perform the following actions and observe the differences:

1. Insert a few rows into the Products table to test the table structure and constraints.

EDIT	PRODUCTID	PRODUCTNAME	PRICE	CATEGORY	SUPPLIEREMAIL	ADDEDDATE
	1	soap	30	washing	piya@gmail.com	12/16/2024
	2	detergent	40	washing	jalmi@gmail.com	12/16/2024
	3	spoon	3	crockery	laxmi@gmail.com	12/16/2024

row(s) 1 - 3 of 3

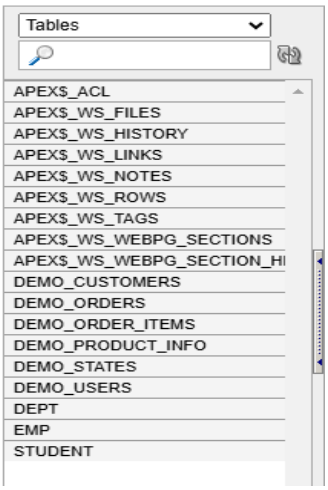
2. Use the TRUNCATE command to remove all rows from the table without deleting its structure.



TRUNCATE TABLE Products;

3. Use the DROP command to delete the Products table completely.

DROP TABLE Products;



5. Insert 10 Rows into the Products Table

EDIT	PRODUCTID	PRODUCTNAME	PRICE	CATEGORY	STOCK
	1	soap	30	washing	300
	2	detergent	35	washing	300
	3	spoon	3	crockery	350
	4	plate	10	crockery	340
	5	glass	10	crockery	330
	6	bowl	20	crockery	300
	7	blanket	100	furnishing	100
	8	pillow	79	furnishing	110
	9	bed	12000	furnishing	100
	10	quilt	10000	furnishing	120

row(s) 1 - 10 of 10

6. Perform Basic DML Commands

1. SELECT Command (Retrieve Data):

SELECT * FROM Products;

PRODUCTID	PRODUCTNAME	PRICE	CATEGORY	STOCK
1	soap	30	washing	300
2	detergent	35	washing	300
3	spoon	3	crockery	350
4	plate	10	crockery	340
5	glass	10	crockery	330
6	bowl	20	crockery	300
7	blanket	100	furnishing	100
8	pillow	79	furnishing	110
9	bed	12000	furnishing	100
10	quilt	10000	furnishing	120