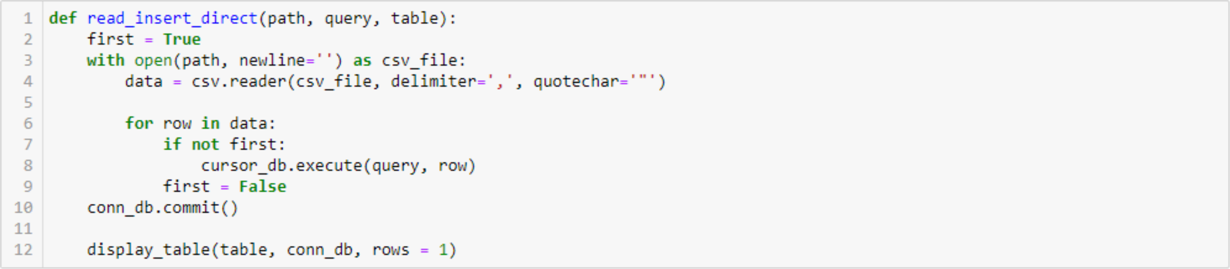
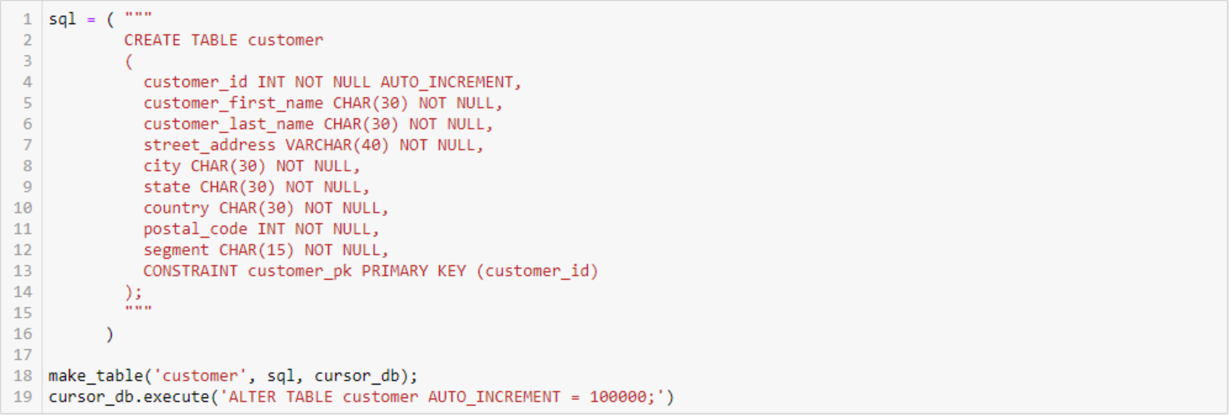
Created a function read_insert_direct to insert data from csv file

Extracting the data from the CSV file and loading the data into a dataframe



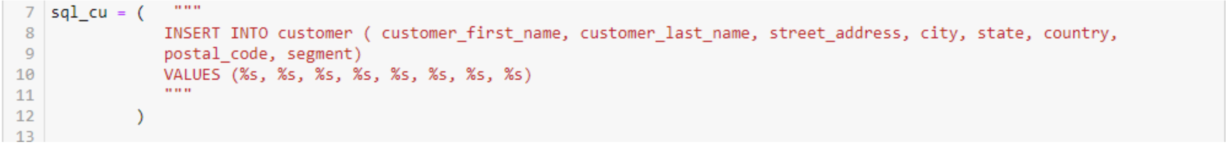
```python
def read_insert_direct(path, query, table):
    first = True
    with open(path, newline='') as csv_file:
        data = csv.reader(csv_file, delimiter=',', quotechar='"')

        for row in data:
            if not first:
                cursor_db.execute(query, row)
            first = False
    conn_db.commit()

    display_table(table, conn_db, rows = 1)
```

1. Customer Table Creation

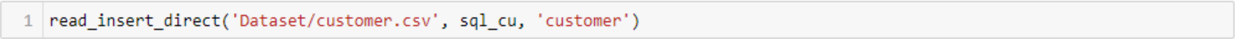


```python
sql = ( """
        CREATE TABLE customer
        (
          customer_id INT NOT NULL AUTO_INCREMENT,
          customer_first_name CHAR(30) NOT NULL,
          customer_last_name CHAR(30) NOT NULL,
          street_address VARCHAR(40) NOT NULL,
          city CHAR(30) NOT NULL,
          state CHAR(30) NOT NULL,
          country CHAR(30) NOT NULL,
          postal_code INT NOT NULL,
          segment CHAR(15) NOT NULL,
          CONSTRAINT customer_pk PRIMARY KEY (customer_id)
        );
        """
      )

make_table('customer', sql, cursor_db);
cursor_db.execute('ALTER TABLE customer AUTO_INCREMENT = 100000;')
```

Data Insertion in Customer Table

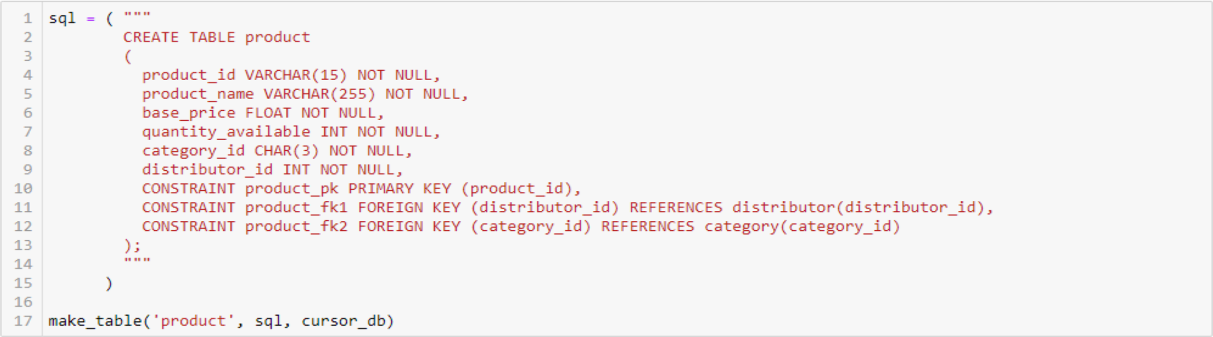


```python
sql_cu = (   """
            INSERT INTO customer ( customer_first_name, customer_last_name, street_address, city, state, country,
            postal_code, segment)
            VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
            """
         )

```



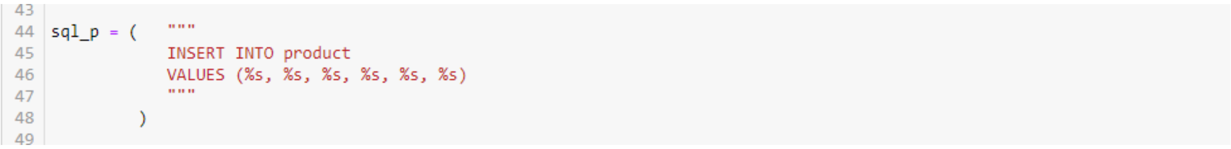
```python
read_insert_direct('Dataset/customer.csv', sql_cu, 'customer')
```

2. Product Table Creation

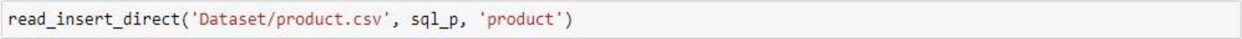


```python
sql = ( """
        CREATE TABLE product
        (
          product_id VARCHAR(15) NOT NULL,
          product_name VARCHAR(255) NOT NULL,
          base_price FLOAT NOT NULL,
          quantity_available INT NOT NULL,
          category_id CHAR(3) NOT NULL,
          distributor_id INT NOT NULL,
          CONSTRAINT product_pk PRIMARY KEY (product_id),
          CONSTRAINT product_fk1 FOREIGN KEY (distributor_id) REFERENCES distributor(distributor_id),
          CONSTRAINT product_fk2 FOREIGN KEY (category_id) REFERENCES category(category_id)
        );
        """
      )

make_table('product', sql, cursor_db)
```

Data Insertion in Product Table

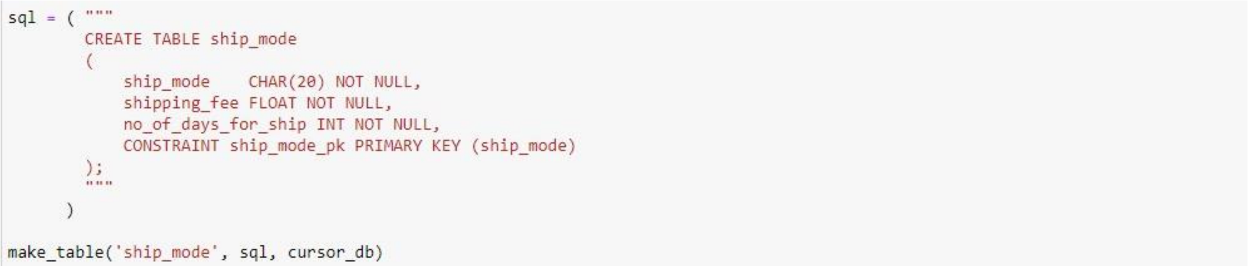


```
sql_p = (   """
            INSERT INTO product
            VALUES (%s, %s, %s, %s, %s, %s)
            """
         )
```



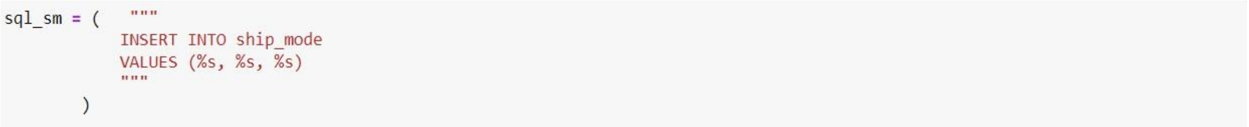
```
read_insert_direct('Dataset/product.csv', sql_p, 'product')
```

3. Ship_mode Table Creation

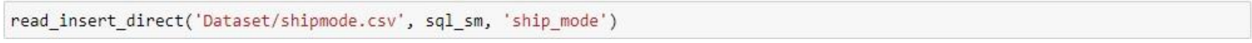


```
sql = ( """
        CREATE TABLE ship_mode
        (
            ship_mode    CHAR(20) NOT NULL,
            shipping_fee FLOAT NOT NULL,
            no_of_days_for_ship INT NOT NULL,
            CONSTRAINT ship_mode_pk PRIMARY KEY (ship_mode)
        );
        """
      )

make_table('ship_mode', sql, cursor_db)
```

Data Insertion in Ship_mode Table

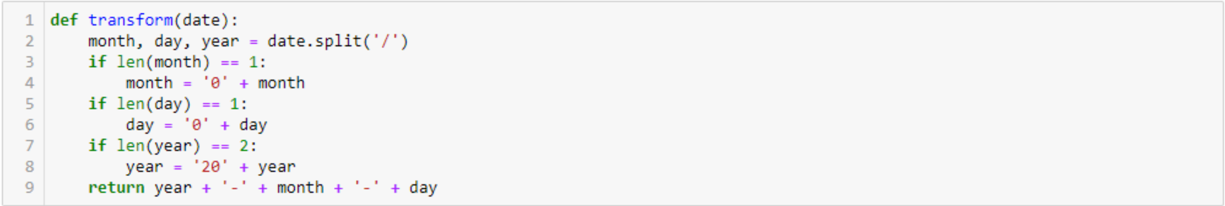


```
sql_sm = (   """
            INSERT INTO ship_mode
            VALUES (%s, %s, %s)
            """
        )
```

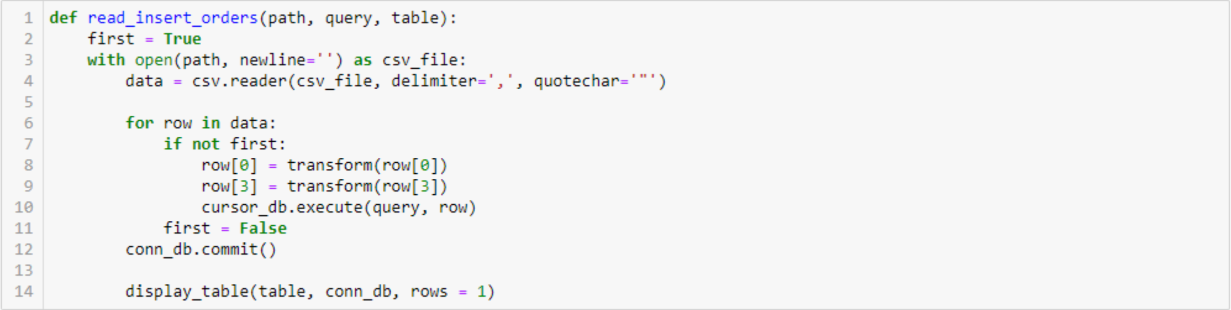


```
read_insert_direct('Dataset/shipmode.csv', sql_sm, 'ship_mode')
```

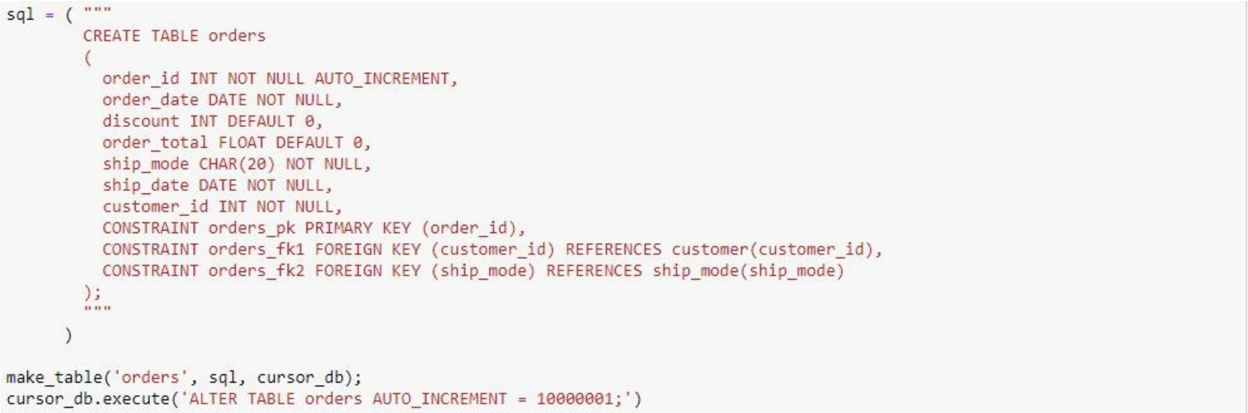
4. Orders Table Creation



```
def transform(date):
    month, day, year = date.split('/')
    if len(month) == 1:
        month = '0' + month
    if len(day) == 1:
        day = '0' + day
    if len(year) == 2:
        year = '20' + year
    return year + '-' + month + '-' + day
```

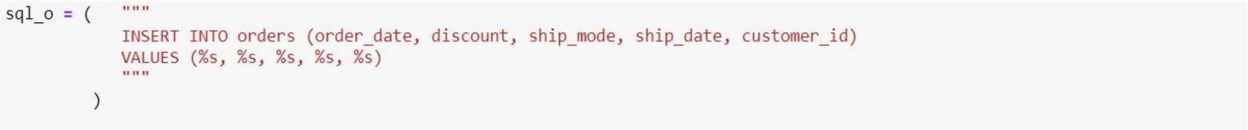


```
def read_insert_orders(path, query, table):
    first = True
    with open(path, newline='') as csv_file:
        data = csv.reader(csv_file, delimiter=',', quotechar='"')

        for row in data:
            if not first:
                row[0] = transform(row[0])
                row[3] = transform(row[3])
                cursor_db.execute(query, row)
            first = False
        conn_db.commit()

        display_table(table, conn_db, rows = 1)
```

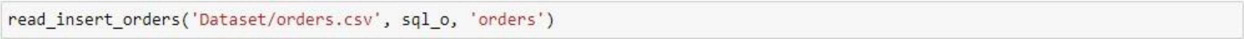


```
sql = ( """
        CREATE TABLE orders
        (
          order_id INT NOT NULL AUTO_INCREMENT,
          order_date DATE NOT NULL,
          discount INT DEFAULT 0,
          order_total FLOAT DEFAULT 0,
          ship_mode CHAR(20) NOT NULL,
          ship_date DATE NOT NULL,
          customer_id INT NOT NULL,
          CONSTRAINT orders_pk PRIMARY KEY (order_id),
          CONSTRAINT orders_fk1 FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
          CONSTRAINT orders_fk2 FOREIGN KEY (ship_mode) REFERENCES ship_mode(ship_mode)
        );
        """
      )

make_table('orders', sql, cursor_db);
cursor_db.execute('ALTER TABLE orders AUTO_INCREMENT = 10000001;')
```

Data Insertion in Orders Table

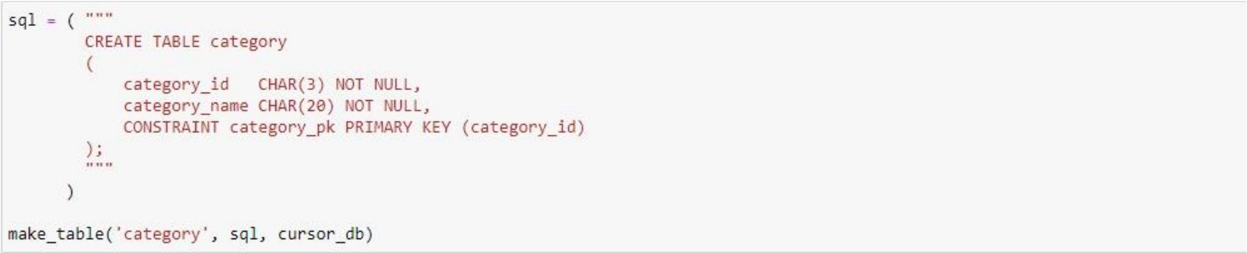


```
sql_o = (   """
            INSERT INTO orders (order_date, discount, ship_mode, ship_date, customer_id)
            VALUES (%s, %s, %s, %s, %s)
            """
         )
```



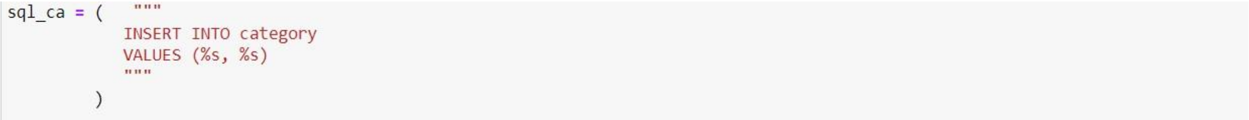
```
read_insert_orders('Dataset/orders.csv', sql_o, 'orders')
```

5. Category Table Creation

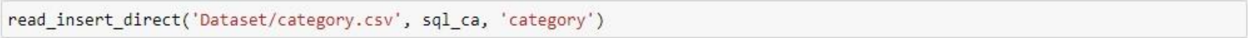


```
sql = ( """
        CREATE TABLE category
        (
            category_id   CHAR(3) NOT NULL,
            category_name CHAR(20) NOT NULL,
            CONSTRAINT category_pk PRIMARY KEY (category_id)
        );
        """
      )

make_table('category', sql, cursor_db)
```

Data Insertion in Category Table

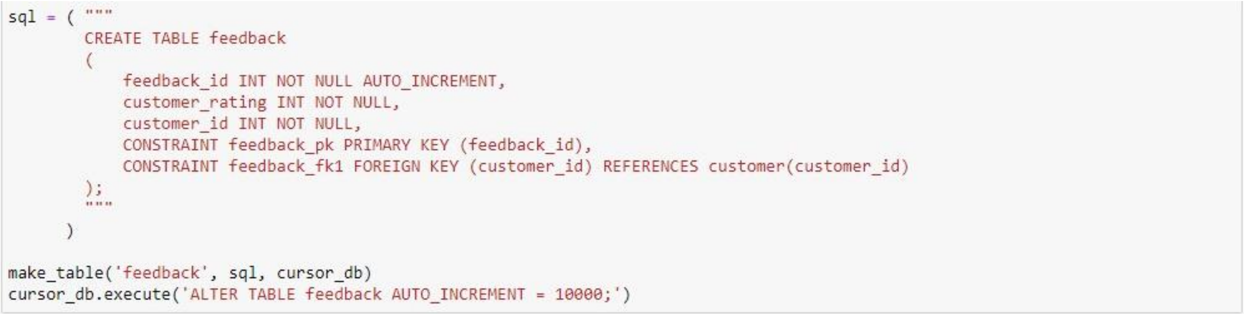


```
sql_ca = (   """
            INSERT INTO category
            VALUES (%s, %s)
            """
         )
```



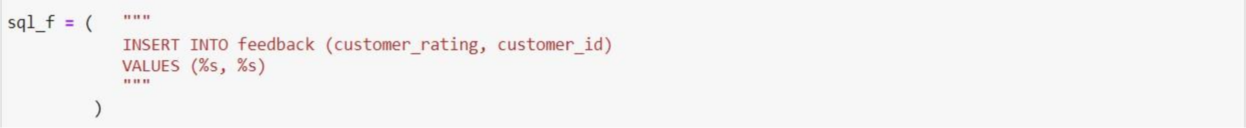
```
read_insert_direct('Dataset/category.csv', sql_ca, 'category')
```

6. Feedback Table Creation

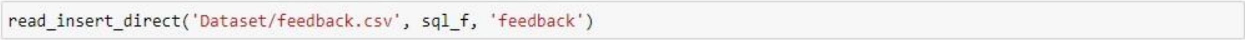


```
sql = ( """
        CREATE TABLE feedback
        (
            feedback_id INT NOT NULL AUTO_INCREMENT,
            customer_rating INT NOT NULL,
            customer_id INT NOT NULL,
            CONSTRAINT feedback_pk PRIMARY KEY (feedback_id),
            CONSTRAINT feedback_fk1 FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
        );
        """
      )

make_table('feedback', sql, cursor_db)
cursor_db.execute('ALTER TABLE feedback AUTO_INCREMENT = 10000;')
```

Data Insertion in Feedback Table

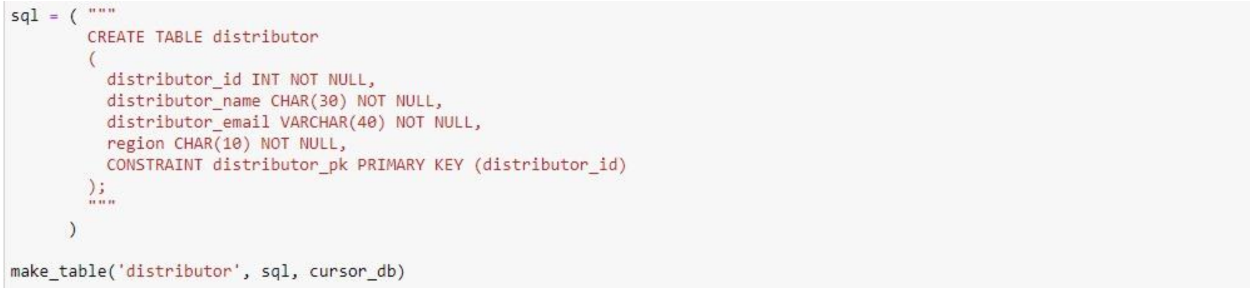


```
sql_f = (    """
             INSERT INTO feedback (customer_rating, customer_id)
             VALUES (%s, %s)
             """
         )
```



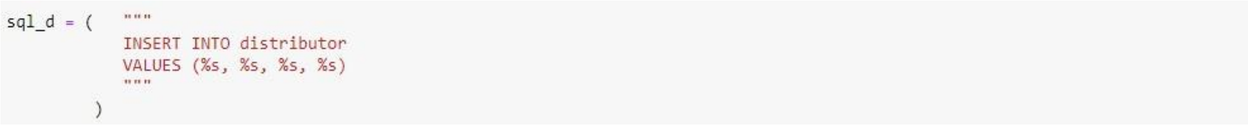
```
read_insert_direct('Dataset/feedback.csv', sql_f, 'feedback')
```

7. Distributor Table Creation

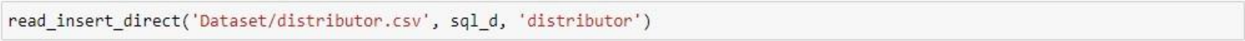


```
sql = ( """
        CREATE TABLE distributor
        (
          distributor_id INT NOT NULL,
          distributor_name CHAR(30) NOT NULL,
          distributor_email VARCHAR(40) NOT NULL,
          region CHAR(10) NOT NULL,
          CONSTRAINT distributor_pk PRIMARY KEY (distributor_id)
        );
        """
      )

make_table('distributor', sql, cursor_db)
```

Data Insertion in Distributor Table

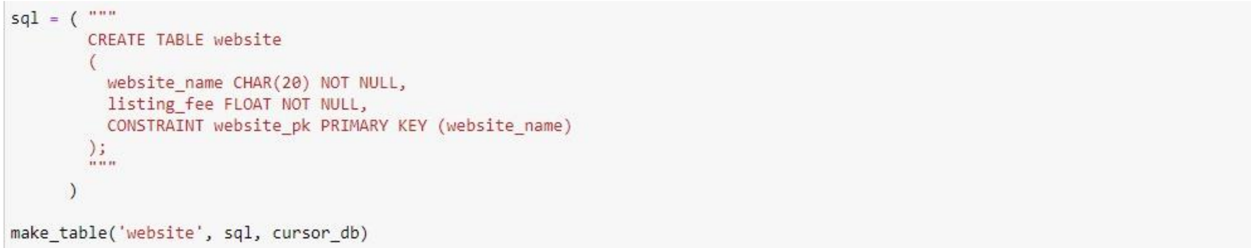


```
sql_d = (    """
             INSERT INTO distributor
             VALUES (%s, %s, %s, %s)
             """
         )
```



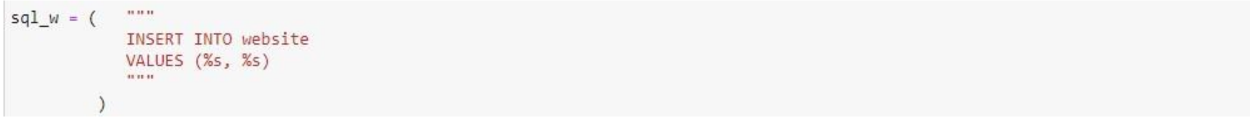
```
read_insert_direct('Dataset/distributor.csv', sql_d, 'distributor')
```

8. Website Table Creation

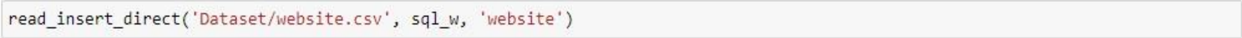


```
sql = ( """
        CREATE TABLE website
        (
          website_name CHAR(20) NOT NULL,
          listing_fee FLOAT NOT NULL,
          CONSTRAINT website_pk PRIMARY KEY (website_name)
        );
        """
      )

make_table('website', sql, cursor_db)
```

Data Insertion in Website Table

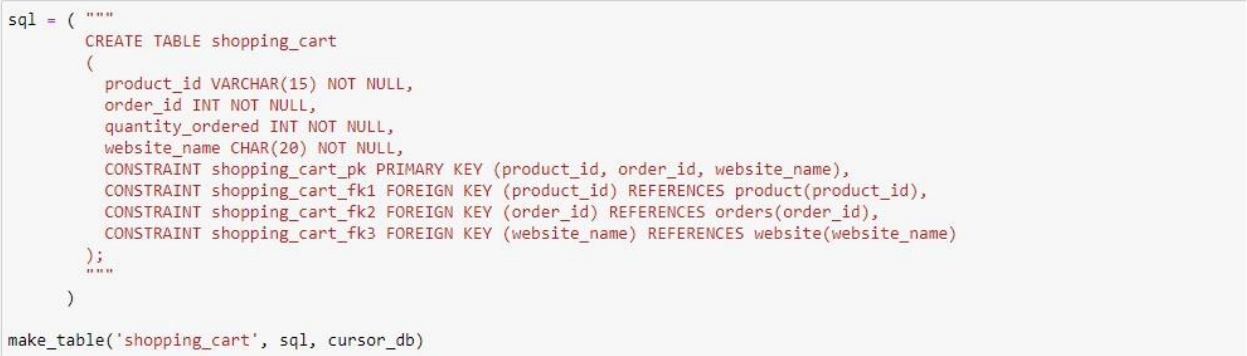


```
sql_w = (    """
            INSERT INTO website
            VALUES (%s, %s)
            """
         )
```



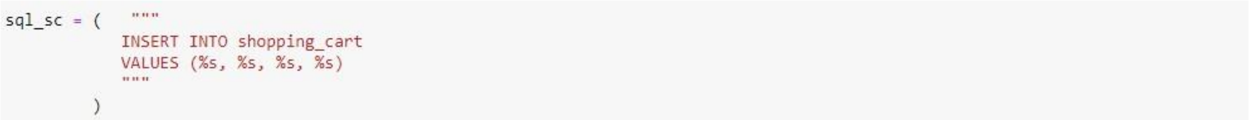
```
read_insert_direct('Dataset/website.csv', sql_w, 'website')
```

9. Shopping Cart Table Creation

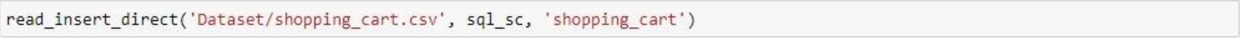


```
sql = ( """
        CREATE TABLE shopping_cart
        (
          product_id VARCHAR(15) NOT NULL,
          order_id INT NOT NULL,
          quantity_ordered INT NOT NULL,
          website_name CHAR(20) NOT NULL,
          CONSTRAINT shopping_cart_pk PRIMARY KEY (product_id, order_id, website_name),
          CONSTRAINT shopping_cart_fk1 FOREIGN KEY (product_id) REFERENCES product(product_id),
          CONSTRAINT shopping_cart_fk2 FOREIGN KEY (order_id) REFERENCES orders(order_id),
          CONSTRAINT shopping_cart_fk3 FOREIGN KEY (website_name) REFERENCES website(website_name)
        );
        """
      )

make_table('shopping_cart', sql, cursor_db)
```

Data Insertion in Shopping Cart Table

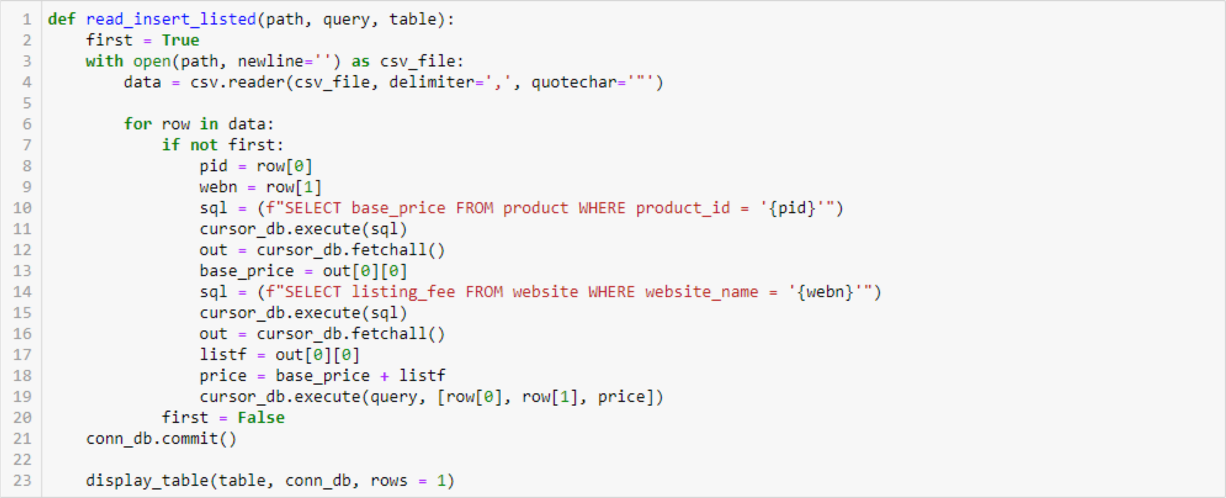


```
sql_sc = (   """
            INSERT INTO shopping_cart
            VALUES (%s, %s, %s, %s)
            """
         )
```

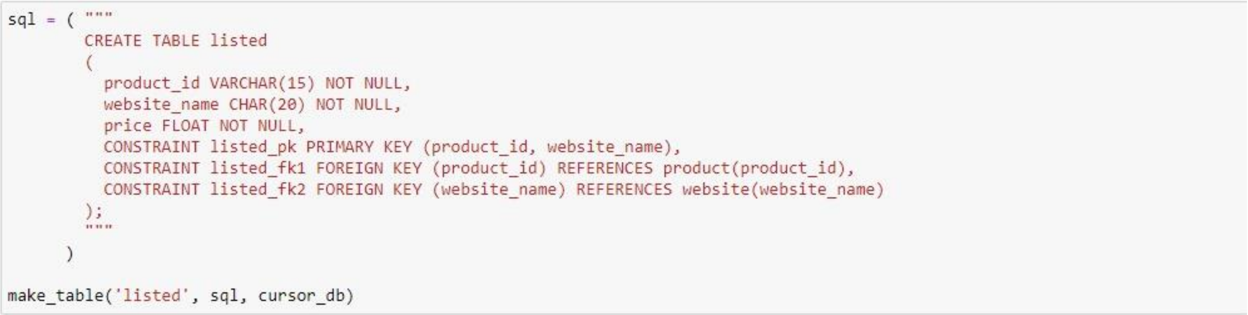


```
read_insert_direct('Dataset/shopping_cart.csv', sql_sc, 'shopping_cart')
```

10. Listed Table Creation

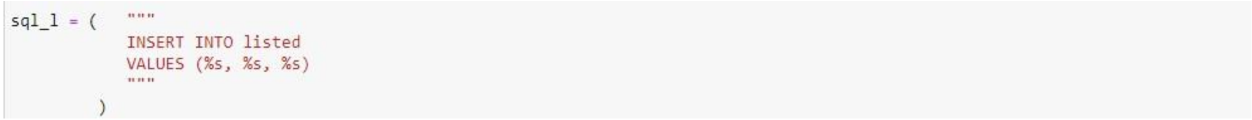


```
def read_insert_listed(path, query, table):
    first = True
    with open(path, newline='') as csv_file:
        data = csv.reader(csv_file, delimiter=',', quotechar='"')

        for row in data:
            if not first:
                pid = row[0]
                webn = row[1]
                sql = (f"SELECT base_price FROM product WHERE product_id = '{pid}'")
                cursor_db.execute(sql)
                out = cursor_db.fetchall()
                base_price = out[0][0]
                sql = (f"SELECT listing_fee FROM website WHERE website_name = '{webn}'")
                cursor_db.execute(sql)
                out = cursor_db.fetchall()
                listf = out[0][0]
                price = base_price + listf
                cursor_db.execute(query, [row[0], row[1], price])
            first = False
    conn_db.commit()

    display_table(table, conn_db, rows = 1)
```

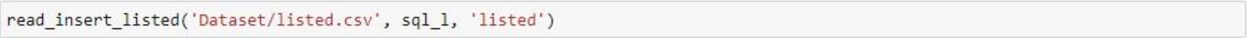


```
sql = ( """
        CREATE TABLE listed
        (
          product_id VARCHAR(15) NOT NULL,
          website_name CHAR(20) NOT NULL,
          price FLOAT NOT NULL,
          CONSTRAINT listed_pk PRIMARY KEY (product_id, website_name),
          CONSTRAINT listed_fk1 FOREIGN KEY (product_id) REFERENCES product(product_id),
          CONSTRAINT listed_fk2 FOREIGN KEY (website_name) REFERENCES website(website_name)
        );
        """
      )

make_table('listed', sql, cursor_db)
```

Data Insertion in Listed Table



```
sql_l = (   """
            INSERT INTO listed
            VALUES (%s, %s, %s)
            """
         )
```



```
read_insert_listed('Dataset/listed.csv', sql_l, 'listed')
```