SQL EVIDENCE

```
1 Create TABLE book(
 2 bno INT PRIMARY KEY
 3 CHECK (bno >= 100000 and bno <= 999999),
   title VARCHAR(255) NOT NULL,
    author VARCHAR(255) NOT NULL,
 6 category VARCHAR(50)
 7 CHECK (category IN ('Science', 'Lifestyle', 'Arts', 'Leisure')),
 8 price NUMERIC NOT NULL,
    sales INT DEFAULT 0
10
11
12 select bno AS Book_No,
13 title AS Book_Title,
   author, category AS GENRE,
15 price AS Book_Price,
16 sales AS Books_Sold From Book;
Data Output Messages Notifications
book_no
                               author
             book title
                                                genre
                                                                 book_price
                                                                           books_sold
             character varying (255)
                               character varying (255)
                                                character varying (50)
                                                                           integer
      Create TABLE customer(
  1
  2
      cno INT PRIMARY KEY
  3
      CHECK (cno >= 100000 and cno <= 999999),
  4
      name VARCHAR(255) NOT NULL,
      address VARCHAR(255) NOT NULL,
  5
  6
      balance NUMERIC DEFAULT 0
  7
      );
  8
  9
      select cno AS customer_no,
10
      name AS Customer_Name,
11
      address AS Customer_address,
12
      balance AS Remaining_Dues
13
      From Customer:
Data Output
                          Notifications
              Messages
                     customer_name
                                             customer_address
      customer_no
                                                                    remaining_dues
                     character varying (255)
      integer
                                             character varying (255)
                                                                    numeric
```

```
Create TABLE bookOrder (
1
2
    cno INT,
 3 bno INT,
    orderTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 4
    qty INT NOT NULL
5
6
    );
7
8
    select cno AS customer_no,
    bno AS book_no,
9
    orderTime,
10
    qty AS quantity from bookOrder;
11
    Data Output
                 Messages
                           Notifications
    =+
          customer_no
                        book_no
                                  ordertime
                                                           quantity
                                  timestamp without time zone
                                                           integer
          integer
                        integer
```

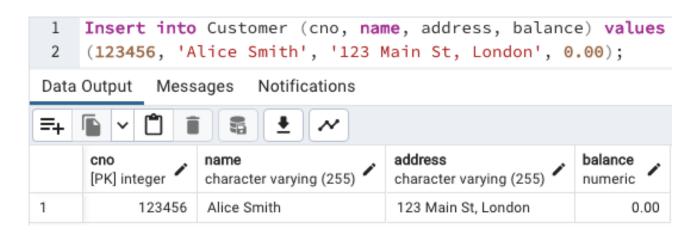
A) Insert a new book



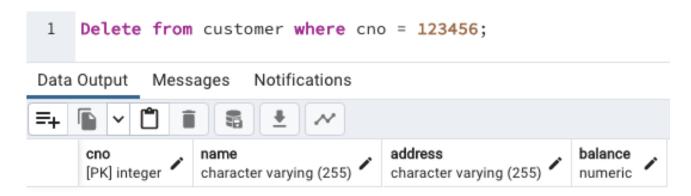
B) Delete a Book



C) Insert a Customer



D) Delete a Customer



- E) Place an order for a customer for a specified number of copies of a book. The copies ordered are assumed to be sold and will have to be paid for. Books are not supplied on 'sale or return' terms business. Check if the book and customer is already in the system. If there is book entry or customer information is not available then create required entries and then perform the operation.
- i) Initial values in the Tables Customer and Book.

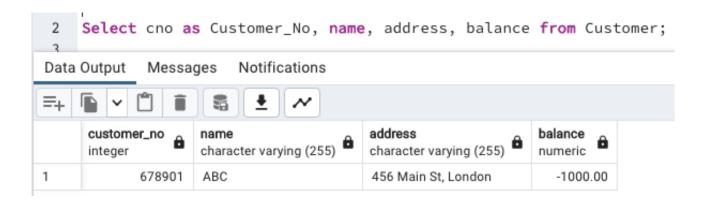




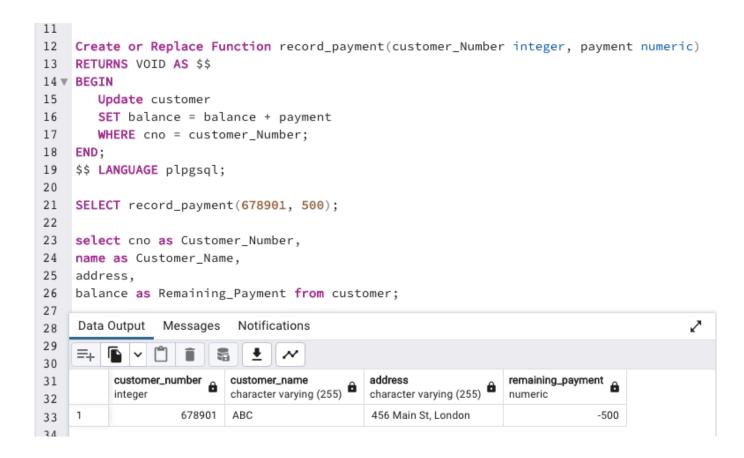
```
Create Or Replace Function handleorder()
 1
2
   Returns Trigger AS $$
3
   DECLARE
       bookPrice NUMERIC;
4
5
       bookExists Boolean;
 6
       customerExists Boolean;
7 ▼ BEGIN
       IF NOT EXISTS(Select 1 from Book where bno = NEW.bno)then
8 ₩
9
         Insert Into Book(bno, title, author, category, price, sales) values
10
          (New.bno, 'Gulliver Travels', 'Jonathan Swift', 'Leisure', 50.00, NEW.qty);
11
       End IF;
       SELECT price INTO bookPrice
12
13
       FROM Book
14
       WHERE bno = NEW.bno;
15 ₹
       IF NOT EXISTS(Select 1 from Customer where cno=NEW.cno) then
16
         Insert Into Customer(cno, name, address, balance) values
17
          (NEW.cno, 'ABC', '456 Main St, London', 0.00);
18
       End IF;
19
20
       Update customer
       SET balance = balance - (bookPrice*NEW.qty)
21
22
       where cno = NEW.cno;
23
24
       Return NEW;
   END;
   $$ LANGUAGE plpgsql;
27
28
29
   CREATE Trigger handle_order_trigger
30 AFTER INSERT ON bookOrder
31 For EACH ROW
   EXECUTE Function handleorder();
 1
    Insert into bookOrder (bno, cno, qty) values
     (123456, 678901, 10);
 2
 3
     Select bno as Book_No, cno as Customer_No, qty as Quantity from bookOrder;
Data Output
              Messages
                         Notifications
=+
      book_no
                 customer_no
                                quantity
      integer
                 integer
1
         123456
                        678901
                                       10
```

ii) Updated Book Table and Customer Table after inserting.

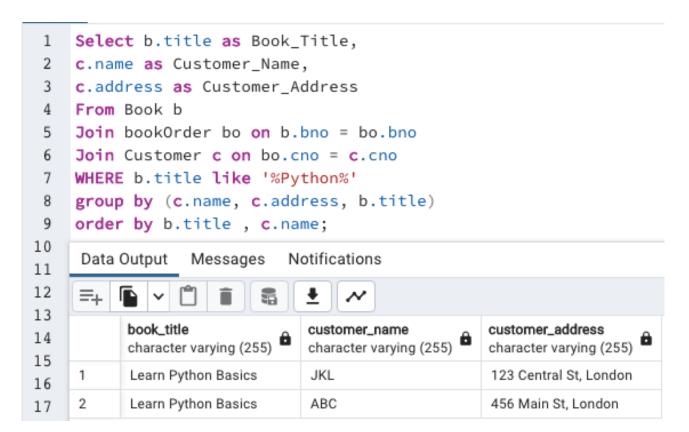




F) Record a payment by a customer. The payment is subtracted from the customer's balance.



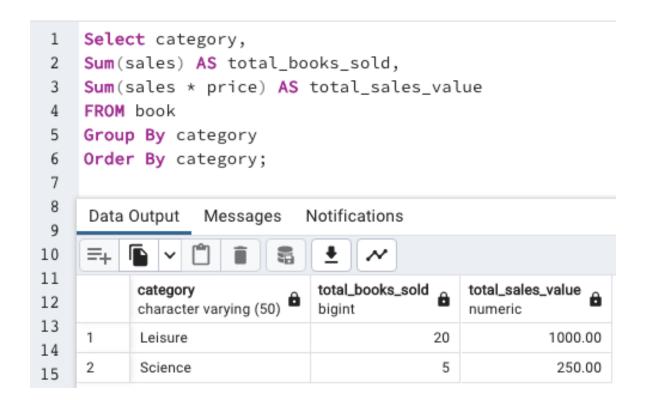
G) Find details of customers who have current orders for a book with a given text fragment in the book title. For example, find customers with orders for books with 'Python' in the title. This transaction produces a report with lines showing the full title of a book ordered, the customer name and the customer address relevant to the order. The report is to be sorted by title and then by customer name.



H) Find details of books ordered by a specified customer. The report will show the name of the customer followed by, for each book, the book number, title and author, sorted by book number.

```
1
     SELECT
 2
          distinct(c.name) AS customer_name,
 3
          b.bno AS book_number,
 4
          b.title AS book_title,
 5
          b.author AS book_author
 6
     FROM
 7
          customer c
 8
     JOIN
 9
          bookOrder bo ON c.cno = bo.cno
10
     JOIN
          book b ON bo.bno = b.bno
11
12
     WHERE
          c.cno = 678901
13
14
     ORDER BY
15
          b.bno;
16
      Data Output
                                 Notifications
                    Messages
17
18
      ≡<sub>+</sub>
19
                                   book_number
                                                   book_title
                                                                          book_author
            customer_name
20
                                                                          character varying (255) 🏥
            character varying (255)
                                   integer
                                                   character varying (255)
21
      1
            ABC
                                           133456
                                                   Learn Python Basics
                                                                          John Doe
22
```

I) Produce a book report by category. This report shows, for each category, the number of books sold and the total value of these sales. The total value calculation assumes that the currently held price is used and any earlier changes in the price of a book since it was inserted into the database are ignored.



J) Produce a customer report. This report shows, for each customer, the customer number, customer name and a count of the number of copies of books on order (if any). This report is to be in customer number order.

