University of East Anglia
COMPUTING SCIENCES

CMP-7025A Database Manipulation

Student Name:	
Reg No:	100498877
Marker:	

Please follow the instructions below to complete the assessment document.

First, for part 1, insert your SQL(DDL) statements including constraints, trigger, functions, views, ER diagram and training (train) data.

Second, for Part 2, before you begin running the SQL (DML) commands for transactions of interest, delete all the training data (your own data) and insert the assessment data using the script provided (Assessment_data.txt); **use tests at the end of the script to ensure all data has been inserted correctly.**

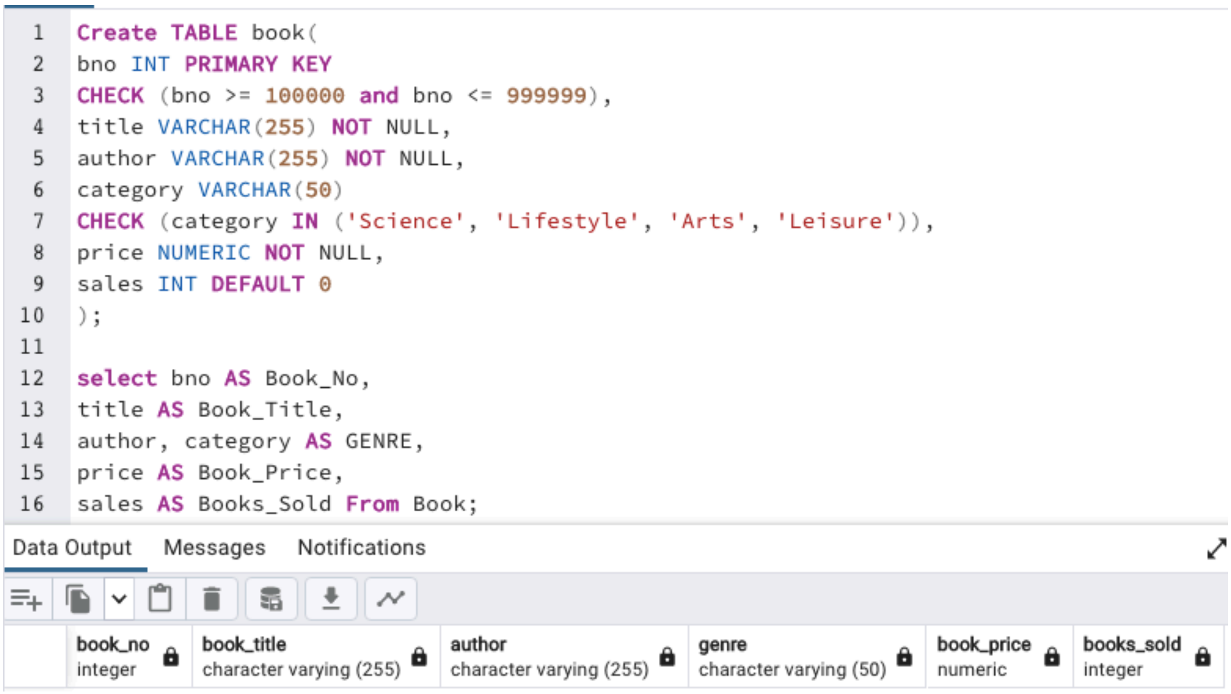
For the individual SQL queries, where your output is expected (in red), insert the SQL statement you run and then the output of running the statement on this document. It may be that a task requires you to run more than one SQL statement; in such cases insert in order the statements you run to achieve the task. You can insert any screenshots of running the tasks in PGAdmin as they will contain both the SQL and the output, but please make sure any text is of a size that can be read easily for marking and if you are showing a table output, ALL rows are visible in the screenshot. If the running of the SQL command results on an error reported by the SQL environment, then record that error (i.e., copy it and paste the screenshot with the error shown). When asked, also insert the contents of tables (i.e., show the result of selecting all tuples from that table). If you have not implemented a particular task, write 'NOT DONE' as the output. Please **do not delete** the marks allocated to each section (in blue) as those will be used by markers. An example of how to fill the form is in the appendix at the end of the document.

Third, run the python program with the data in input file (**input.txt**). Enter/paste to the end of this document (python program output section) the output file (**output.txt**) that results from running the program with input.txt.

Submit all your files as requested in the given format (Sample_submission_folder__directory_and_file_structure), together with this important document in the relevant directory.

DATE: Friday, 2 December 2022

Part 1: Database definition and loading

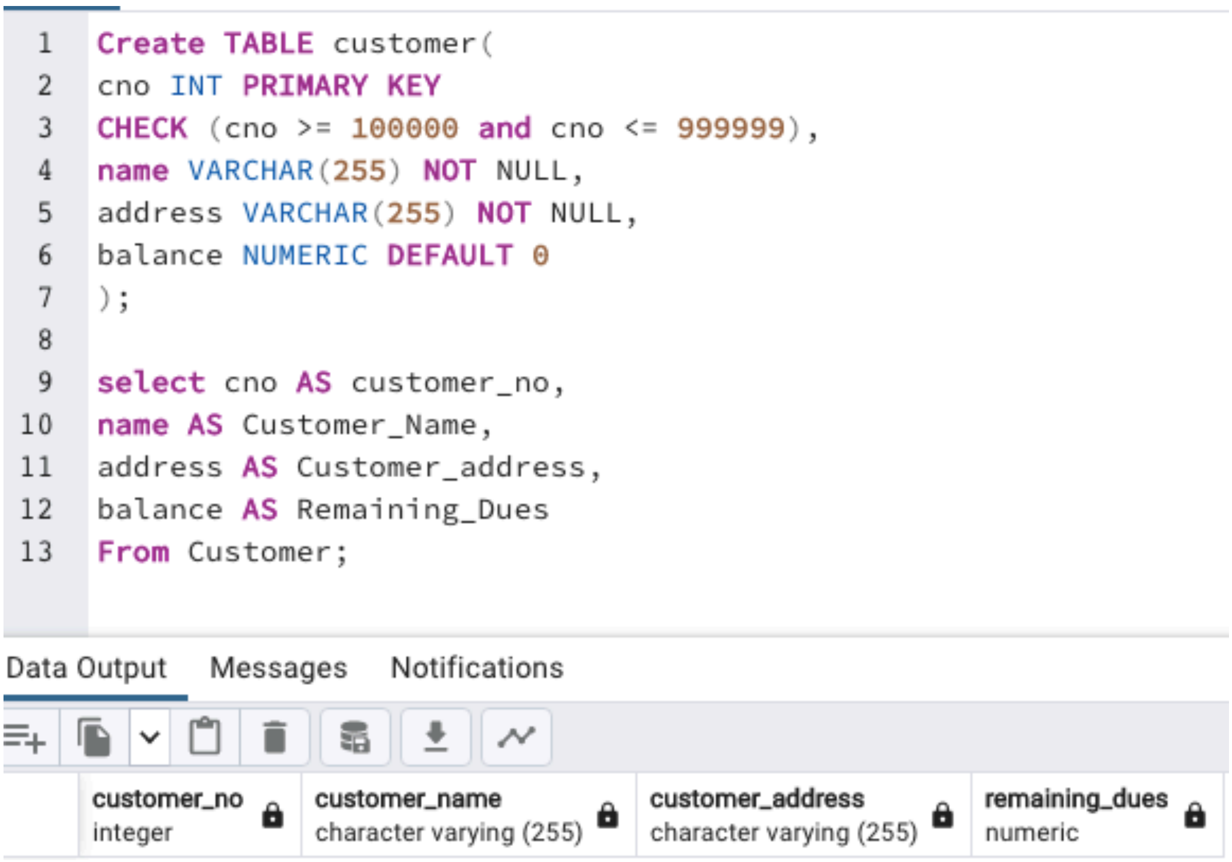


```
Create TABLE book(
bno INT PRIMARY KEY
CHECK (bno >= 100000 and bno <= 999999),
title VARCHAR(255) NOT NULL,
author VARCHAR(255) NOT NULL,
category VARCHAR(50)
CHECK (category IN ('Science', 'Lifestyle', 'Arts', 'Leisure')),
price NUMERIC NOT NULL,
sales INT DEFAULT 0
);

select bno AS Book_No,
title AS Book_Title,
author, category AS GENRE,
price AS Book_Price,
sales AS Books_Sold From Book;
```

Data Output Messages Notifications

book_no integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric	books_sold integer



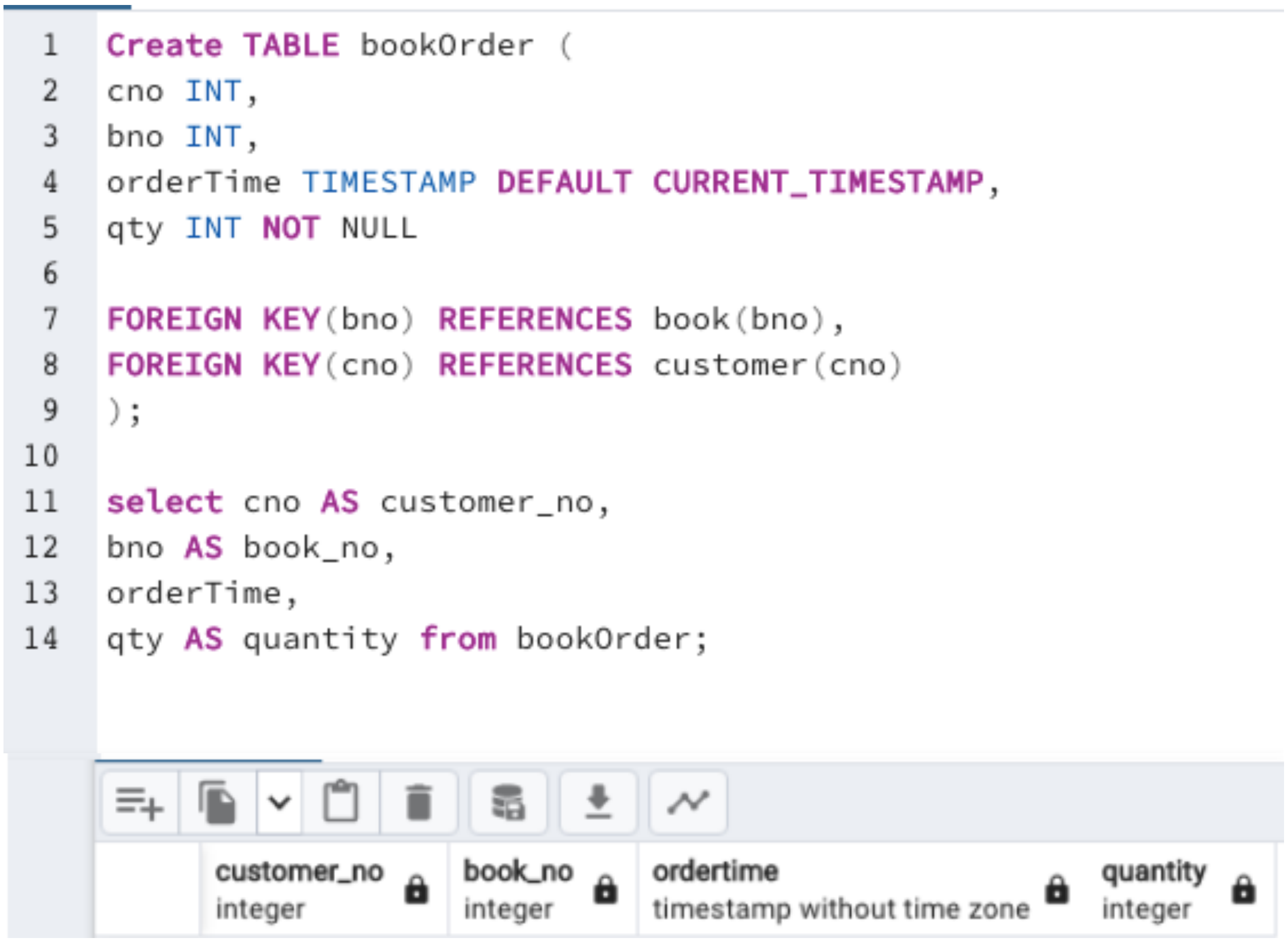
```
Create TABLE customer(
cno INT PRIMARY KEY
CHECK (cno >= 100000 and cno <= 999999),
name VARCHAR(255) NOT NULL,
address VARCHAR(255) NOT NULL,
balance NUMERIC DEFAULT 0
);

select cno AS customer_no,
name AS Customer_Name,
address AS Customer_address,
balance AS Remaining_Dues
From Customer;
```

Data Output Messages Notifications

customer_no integer	customer_name character varying (255)	customer_address character varying (255)	remaining_dues numeric

DATE: Friday, 2 December 2022



```sql
Create TABLE bookOrder (
cno INT,
bno INT,
orderTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
qty INT NOT NULL

FOREIGN KEY(bno) REFERENCES book(bno),
FOREIGN KEY(cno) REFERENCES customer(cno)
);

select cno AS customer_no,
bno AS book_no,
orderTime,
qty AS quantity from bookOrder;
```

	customer_no integer	book_no integer	ordertime timestamp without time zone	quantity integer

Primary Keys	/3
Check constraints	/7
Foreign Keys	/6
Triggers/functions	/4
Others (Views, Indexes, ER diagram)	/4
Train data	/1
TOTAL	/25

University of East Anglia
COMPUTING SCIENCES

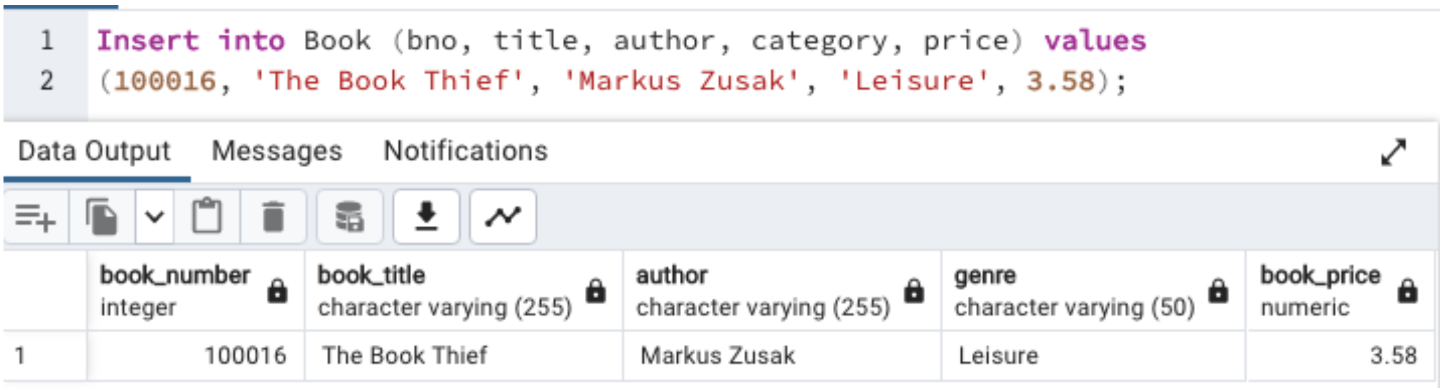
Part 2. Interactive SQL version of the transactions

(For each of the transactions below, please provide sufficient evidence that the transaction is successful. For example, if you are doing an insert statement, show (using SELECT statement) the relevant table after the insert operation is executed.)

Testing task A

Insert a new book

BNO: 100016, Title:'The Book Thief', Author: 'Markus Zusak', Category`:'Leisure', Price: 3.58



```
Insert into Book (bno, title, author, category, price) values
(100016, 'The Book Thief', 'Markus Zusak', 'Leisure', 3.58);
```

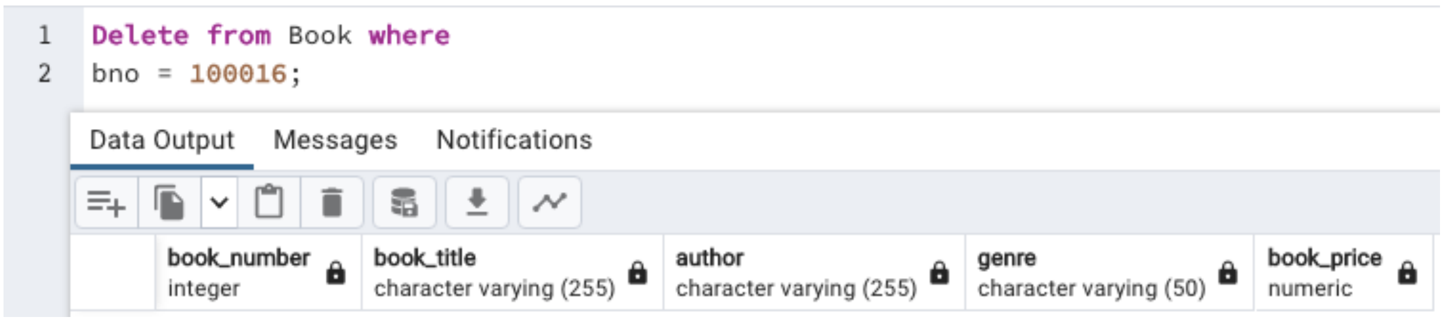
	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100016	The Book Thief	Markus Zusak	Leisure	3.58

Marks: /2

Testing task B

Delete book

BNO: '100016'



```
Delete from Book where
bno = 100016;
```

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric

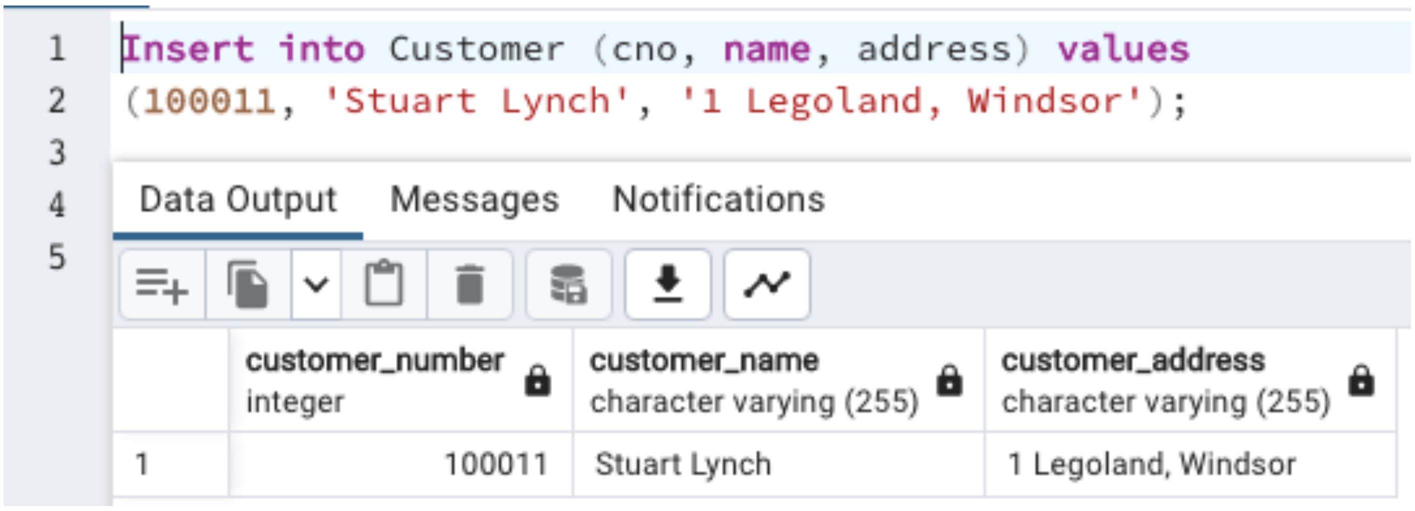
Marks: /2

DATE: Friday, 2 December 2022

Testing task C

Insert customer

CNO: 100011, name: 'Stuart Lynch', address: '1 Legoland, Windsor'



```
Insert into Customer (cno, name, address) values
(100011, 'Stuart Lynch', '1 Legoland, Windsor');
```

Data Output   Messages   Notifications

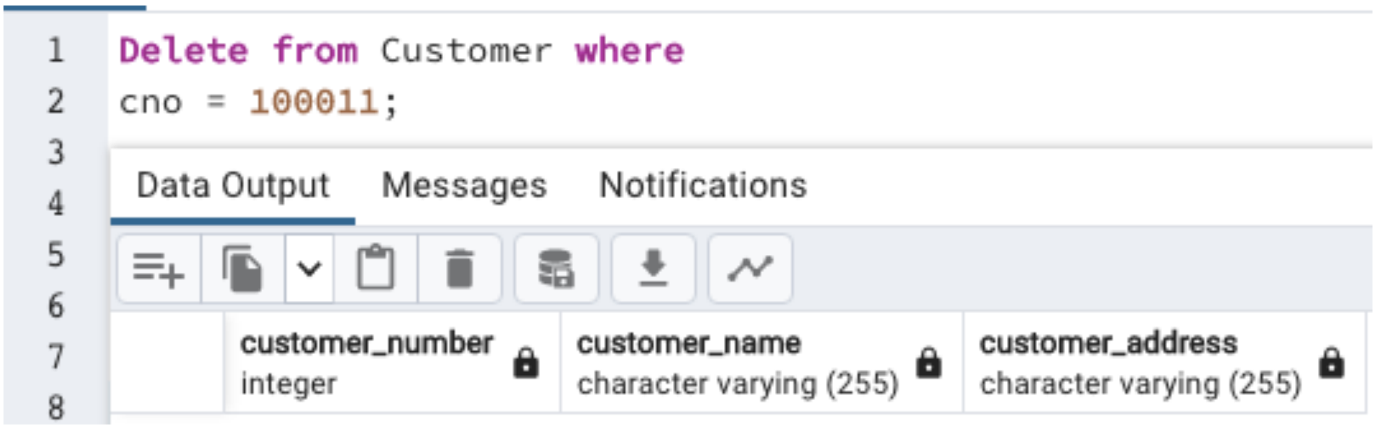
	customer_number integer	customer_name character varying (255)	customer_address character varying (255)
1	100011	Stuart Lynch	1 Legoland, Windsor

Marks: /2

Testing task D

Delete customer

CNO: '100011'



```
Delete from Customer where
cno = 100011;
```

Data Output   Messages   Notifications

	customer_number integer	customer_name character varying (255)	customer_address character varying (255)

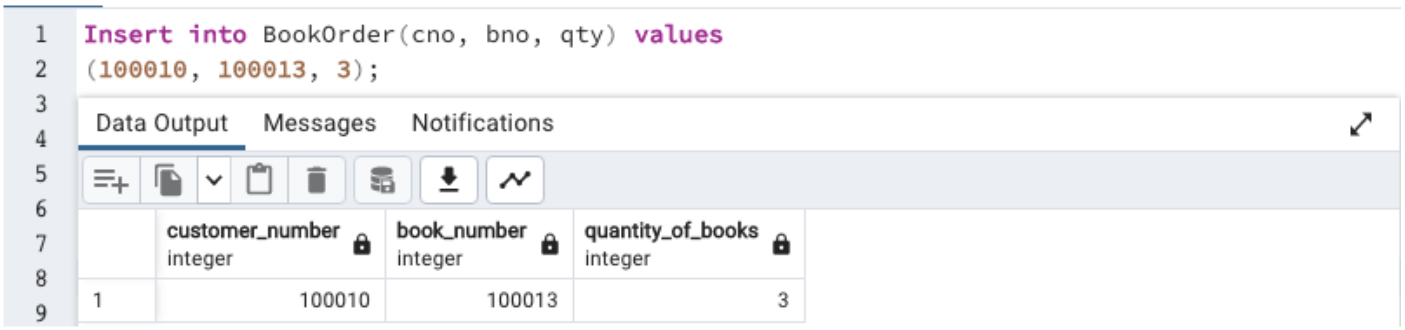
Marks: /2

Testing task E

Place orders

CNO: 100010, BNO: 100013, qty: 3

(***hints: evidence the successful transaction with successful insert and select * from bookOrder, Select * from book and Select * from customer. Show the outputs below***)

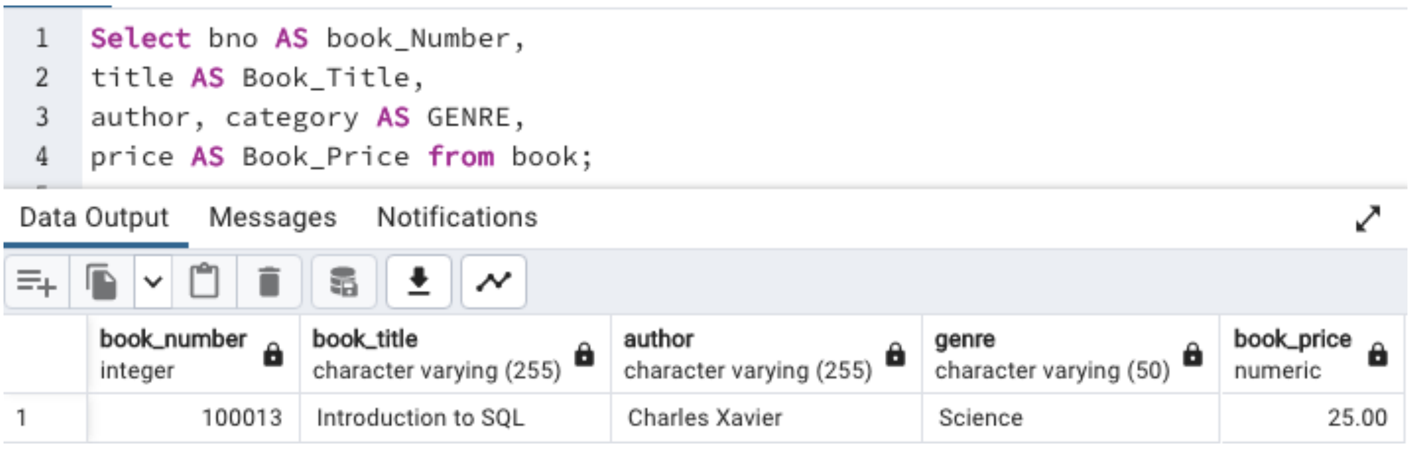


```sql
Insert into BookOrder(cno, bno, qty) values
(100010, 100013, 3);
```

Data Output | Messages | Notifications

	customer_number integer	book_number integer	quantity_of_books integer
1	100010	100013	3

TABLE: Book

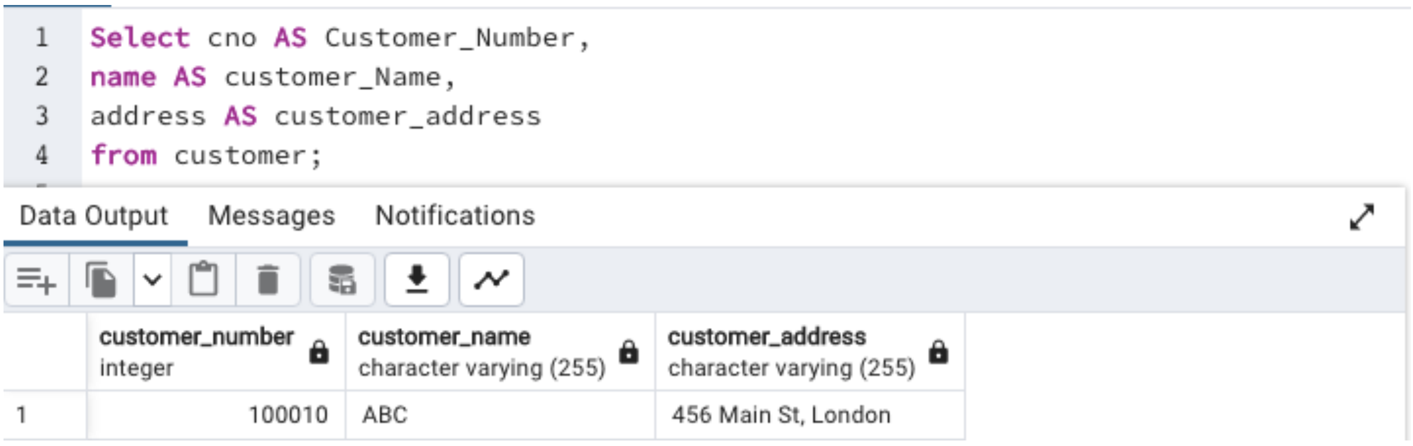


```sql
Select bno AS book_Number,
title AS Book_Title,
author, category AS GENRE,
price AS Book_Price from book;
```

Data Output | Messages | Notifications

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100013	Introduction to SQL	Charles Xavier	Science	25.00

TABLE: Customer



```sql
Select cno AS Customer_Number,
name AS customer_Name,
address AS customer_address
from customer;
```

Data Output | Messages | Notifications

	customer_number integer	customer_name character varying (255)	customer_address character varying (255)
1	100010	ABC	456 Main St, London

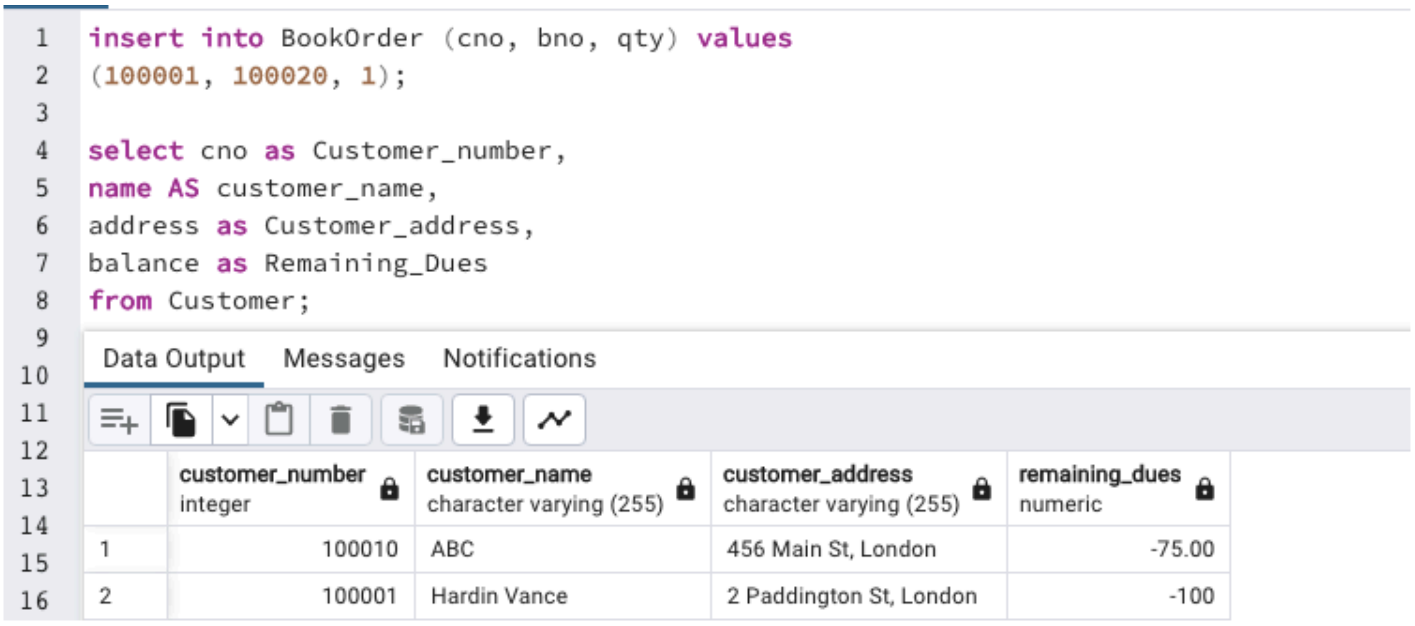
DATE: Friday, 2 December 2022

Marks: /7

Testing task F

Record a Payment by a customer

CNO: 100001, Amount: £100



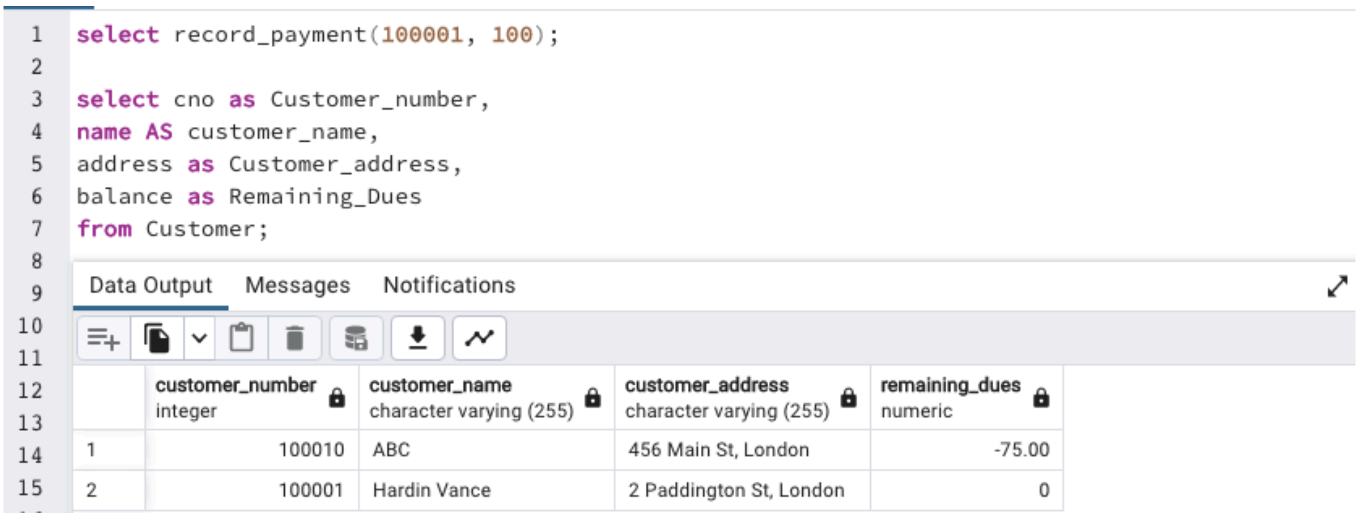
```
insert into BookOrder (cno, bno, qty) values
(100001, 100020, 1);

select cno as Customer_number,
name AS customer_name,
address as Customer_address,
balance as Remaining_Dues
from Customer;
```

Data Output Messages Notifications

	customer_number integer	customer_name character varying (255)	customer_address character varying (255)	remaining_dues numeric
1	100010	ABC	456 Main St, London	-75.00
2	100001	Hardin Vance	2 Paddington St, London	-100

OUTPUT



```
select record_payment(100001, 100);

select cno as Customer_number,
name AS customer_name,
address as Customer_address,
balance as Remaining_Dues
from Customer;
```

Data Output Messages Notifications

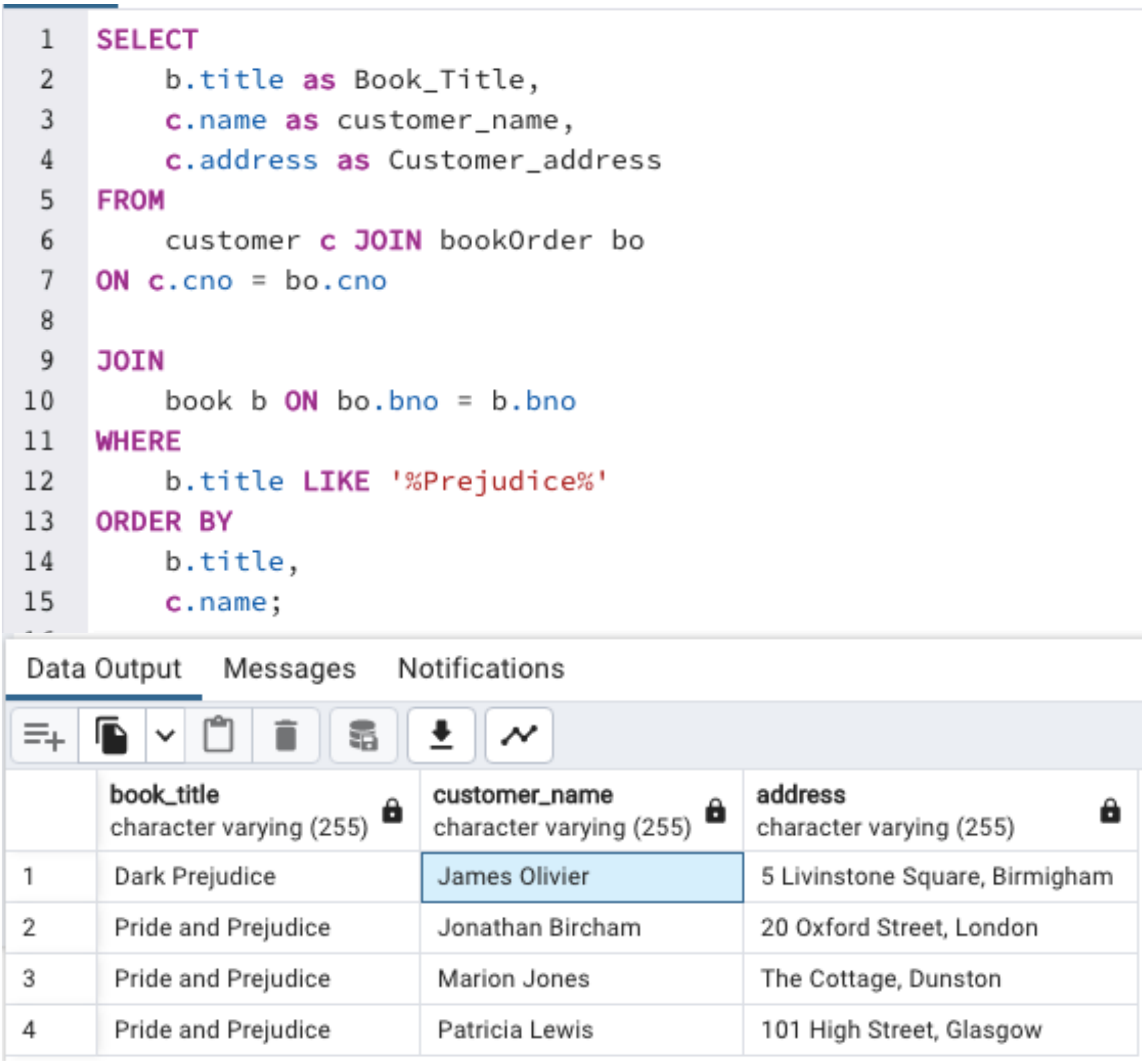
	customer_number integer	customer_name character varying (255)	customer_address character varying (255)	remaining_dues numeric
1	100010	ABC	456 Main St, London	-75.00
2	100001	Hardin Vance	2 Paddington St, London	0

Marks: /2

UEA
University of East Anglia
COMPUTING SCIENCES

Testing task G

Find details of **customers for books like 'Prejudice'/'prejudice'**



```
SELECT
    b.title as Book_Title,
    c.name as customer_name,
    c.address as Customer_address
FROM
    customer c JOIN bookOrder bo
ON c.cno = bo.cno

JOIN
    book b ON bo.bno = b.bno
WHERE
    b.title LIKE '%Prejudice%'
ORDER BY
    b.title,
    c.name;
```

Data Output | Messages | Notifications

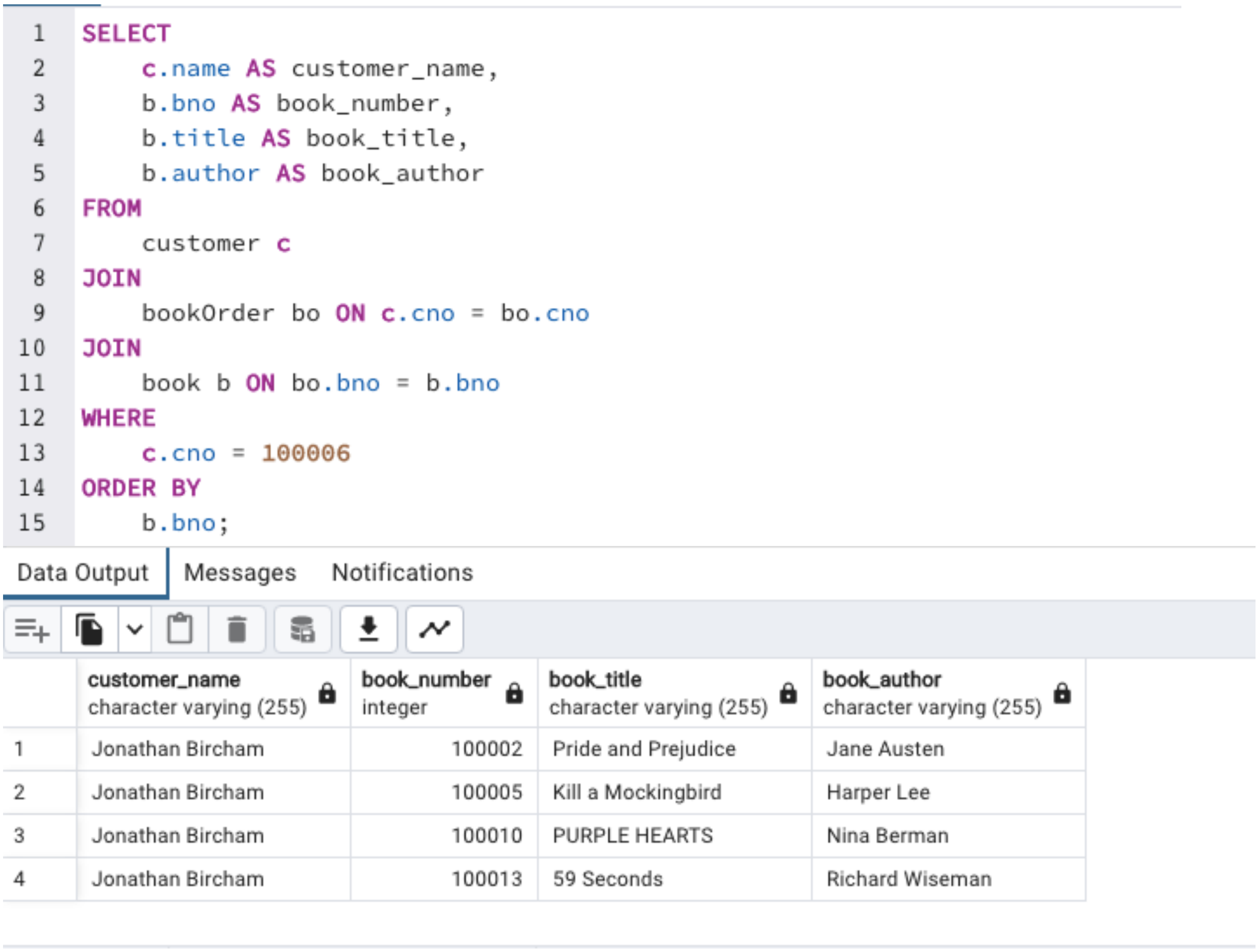
	book_title character varying (255)	customer_name character varying (255)	address character varying (255)
1	Dark Prejudice	James Olivier	5 Livinstone Square, Birmigham
2	Pride and Prejudice	Jonathan Bircham	20 Oxford Street, London
3	Pride and Prejudice	Marion Jones	The Cottage, Dunston
4	Pride and Prejudice	Patricia Lewis	101 High Street, Glasgow

Marks: /6

DATE: Friday, 2 December 2022

Testing task H

Books for customer CNO: 100006



```sql
SELECT
    c.name AS customer_name,
    b.bno AS book_number,
    b.title AS book_title,
    b.author AS book_author
FROM
    customer c
JOIN
    bookOrder bo ON c.cno = bo.cno
JOIN
    book b ON bo.bno = b.bno
WHERE
    c.cno = 100006
ORDER BY
    b.bno;
```

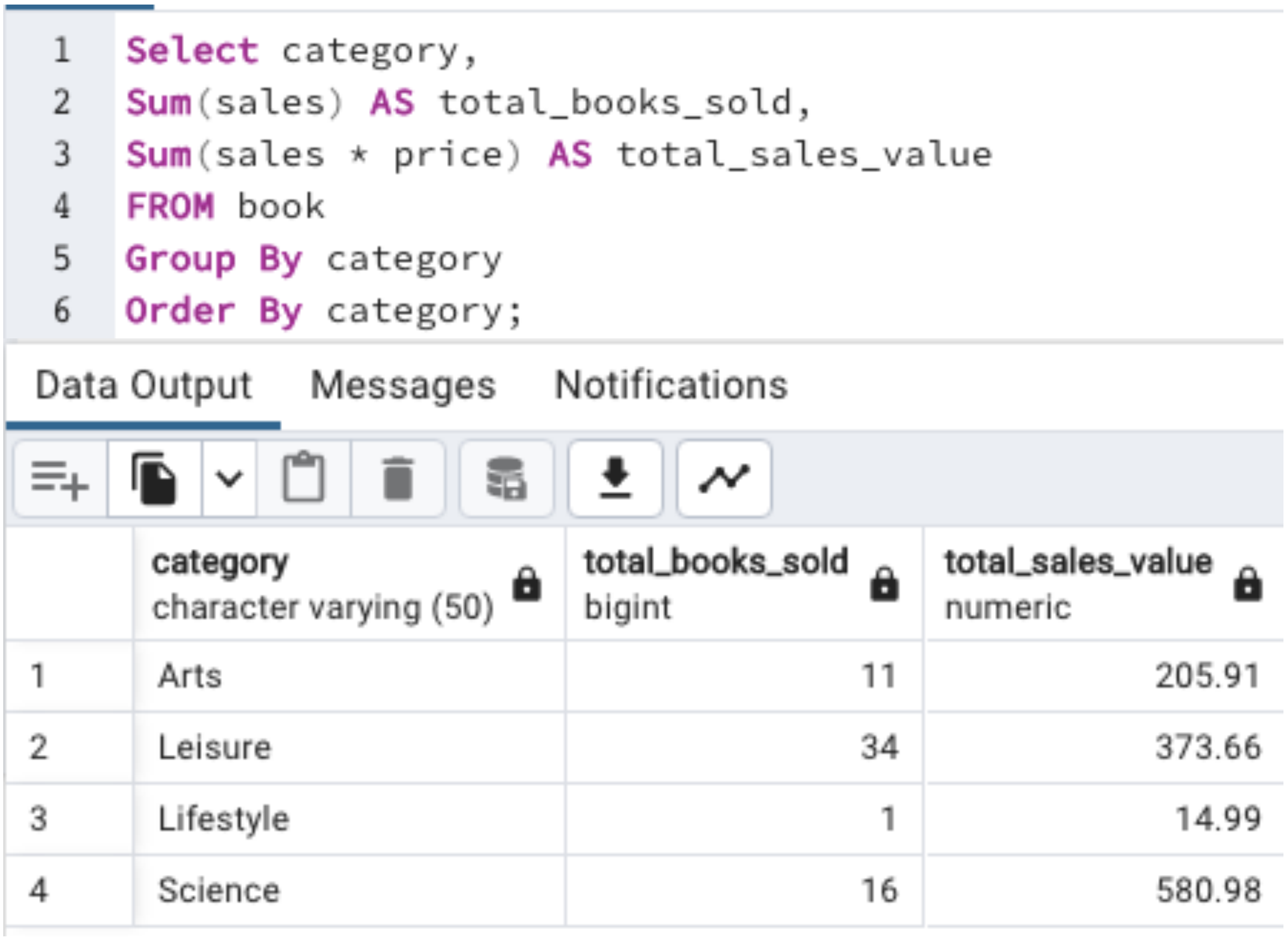
	customer_name character varying (255)	book_number integer	book_title character varying (255)	book_author character varying (255)
1	Jonathan Bircham	100002	Pride and Prejudice	Jane Austen
2	Jonathan Bircham	100005	Kill a Mockingbird	Harper Lee
3	Jonathan Bircham	100010	PURPLE HEARTS	Nina Berman
4	Jonathan Bircham	100013	59 Seconds	Richard Wiseman

Marks: /5

DATE: Friday, 2 December 2022

Testing task I

Book report



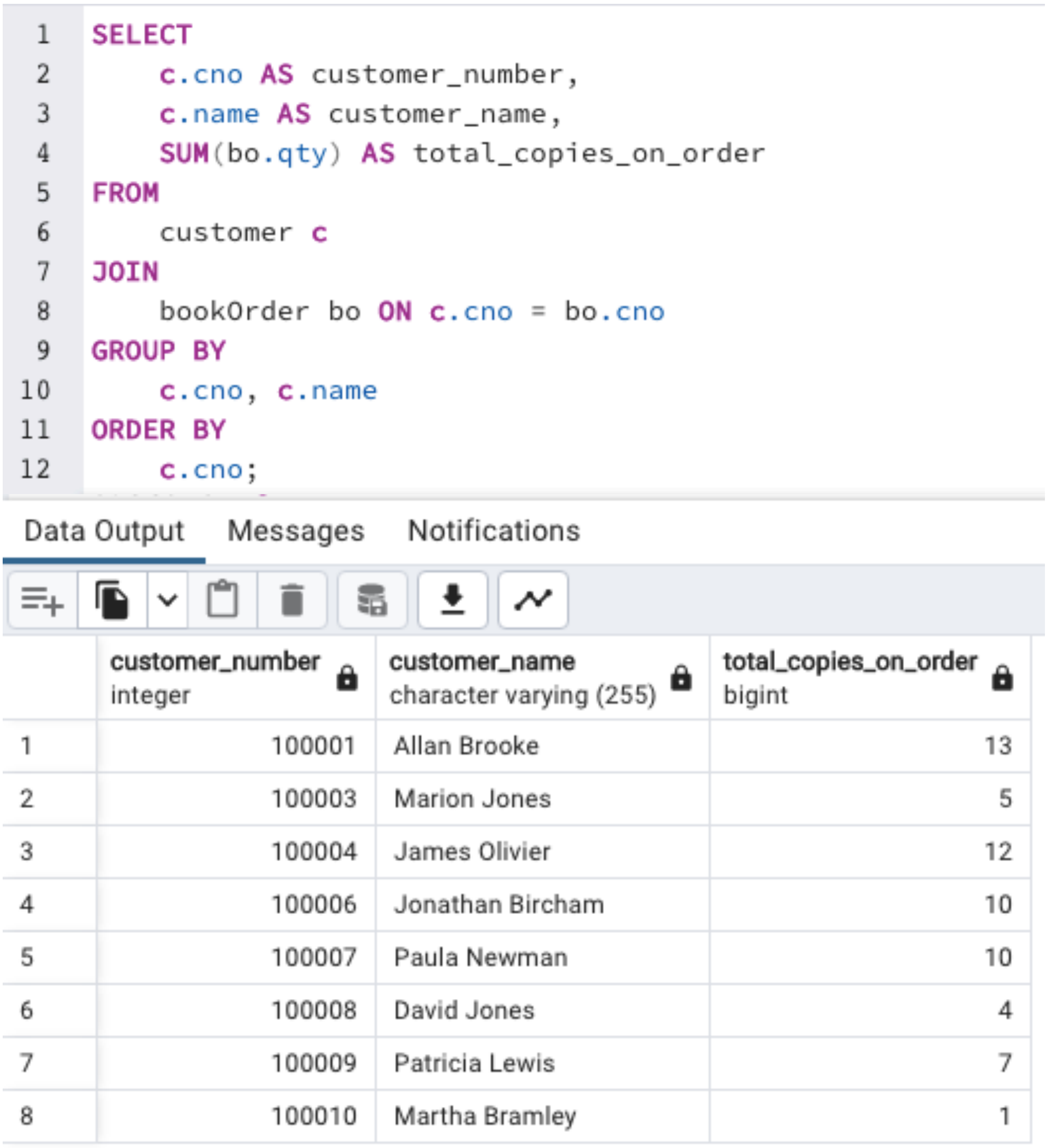
```
Select category,
Sum(sales) AS total_books_sold,
Sum(sales * price) AS total_sales_value
FROM book
Group By category
Order By category;
```

	category character varying (50)	total_books_sold bigint	total_sales_value numeric
1	Arts	11	205.91
2	Leisure	34	373.66
3	Lifestyle	1	14.99
4	Science	16	580.98

Marks: /5

Testing task J

Customer report



```sql
SELECT
    c.cno AS customer_number,
    c.name AS customer_name,
    SUM(bo.qty) AS total_copies_on_order
FROM
    customer c
JOIN
    bookOrder bo ON c.cno = bo.cno
GROUP BY
    c.cno, c.name
ORDER BY
    c.cno;
```

	customer_number integer	customer_name character varying (255)	total_copies_on_order bigint
1	100001	Allan Brooke	13
2	100003	Marion Jones	5
3	100004	James Olivier	12
4	100006	Jonathan Bircham	10
5	100007	Paula Newman	10
6	100008	David Jones	4
7	100009	Patricia Lewis	7
8	100010	Martha Bramley	1

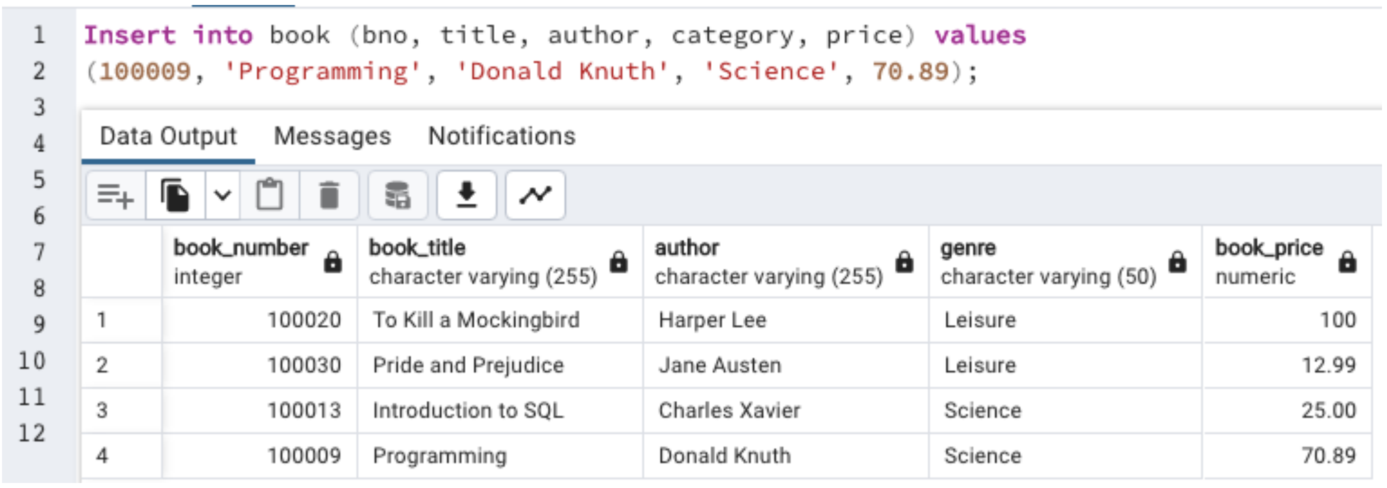
Marks: /7

UEA
University of East Anglia
COMPUTING SCIENCES

Checking the Constraints, Integrity, and Handling of the error cases:

---

1. 'INSERT' BNO: 100009, Title: 'Programming', Author: 'Donald Knuth', Cat: 'Science', Price: 70.89

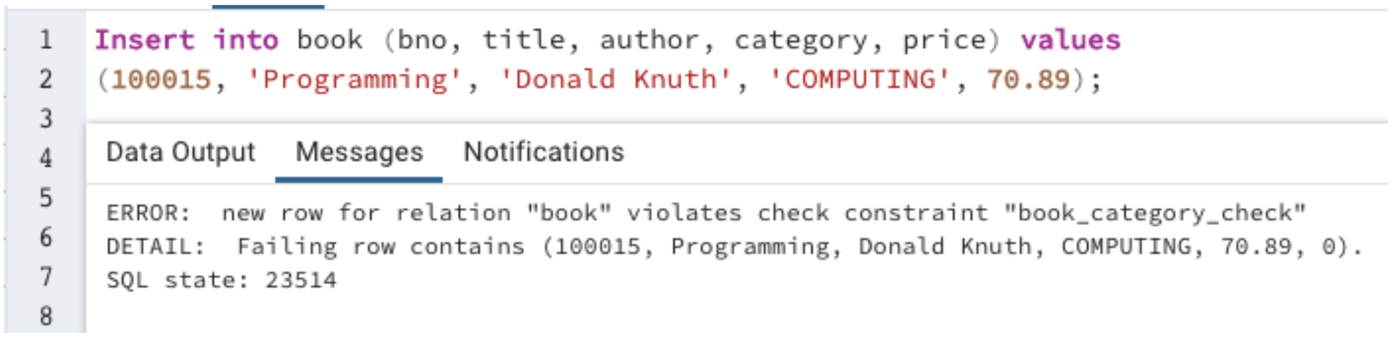


```sql
Insert into book (bno, title, author, category, price) values
(100009, 'Programming', 'Donald Knuth', 'Science', 70.89);
```

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100020	To Kill a Mockingbird	Harper Lee	Leisure	100
2	100030	Pride and Prejudice	Jane Austen	Leisure	12.99
3	100013	Introduction to SQL	Charles Xavier	Science	25.00
4	100009	Programming	Donald Knuth	Science	70.89

Marks: /2

2. 'INSERT' BNO:100015, Title: 'Programming', Author: 'Donald Knuth', Cat: 'Computing', Price: 70.89

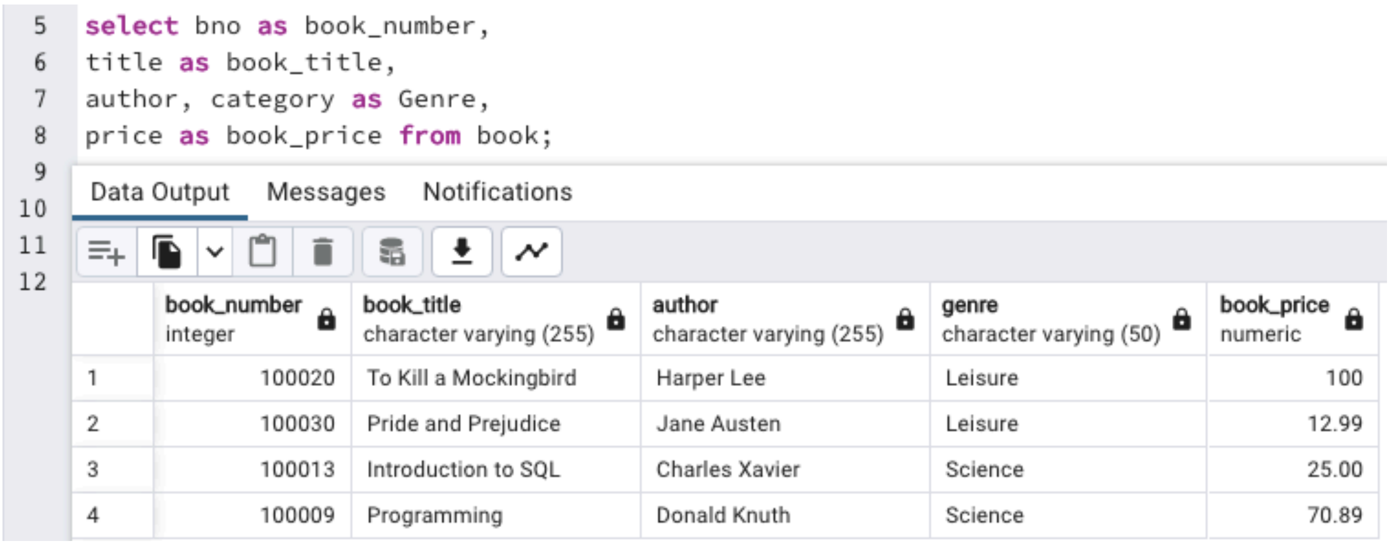


```
Insert into book (bno, title, author, category, price) values
(100015, 'Programming', 'Donald Knuth', 'COMPUTING', 70.89);
```

Data Output Messages Notifications

```
ERROR:  new row for relation "book" violates check constraint "book_category_check"
DETAIL:  Failing row contains (100015, Programming, Donald Knuth, COMPUTING, 70.89, 0).
SQL state: 23514
```

OUTPUT



```
select bno as book_number,
title as book_title,
author, category as Genre,
price as book_price from book;
```

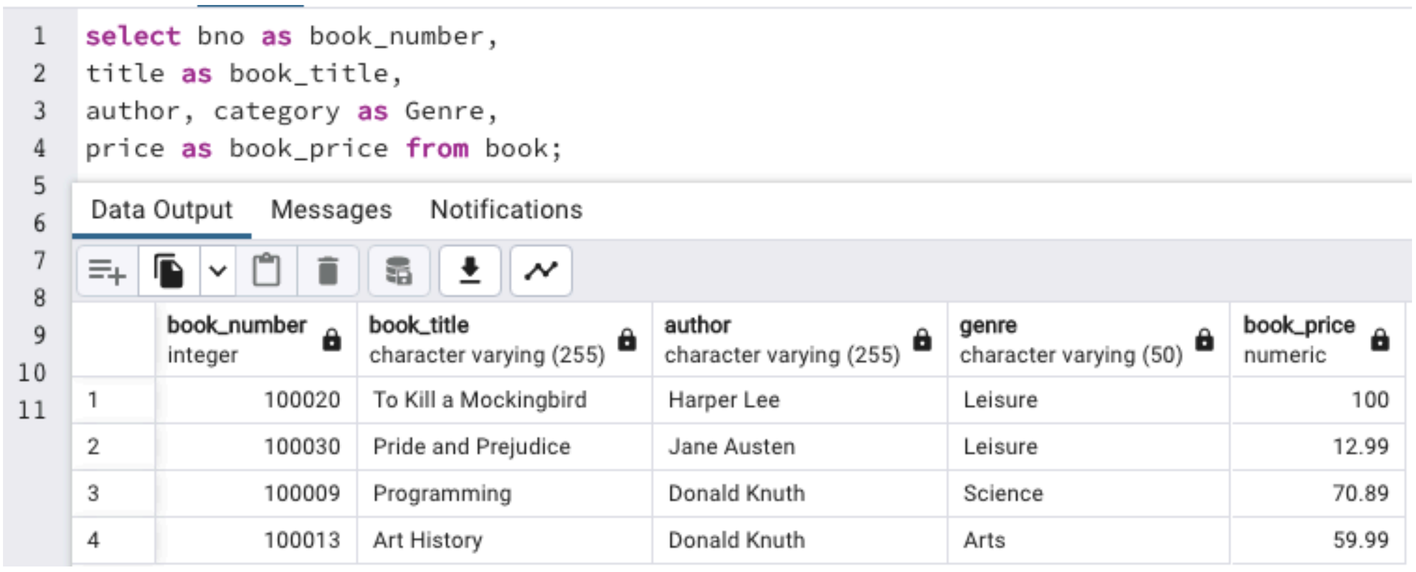
Data Output Messages Notifications

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100020	To Kill a Mockingbird	Harper Lee	Leisure	100
2	100030	Pride and Prejudice	Jane Austen	Leisure	12.99
3	100013	Introduction to SQL	Charles Xavier	Science	25.00
4	100009	Programming	Donald Knuth	Science	70.89

Marks: /2

DATE: Friday, 2 December 2022

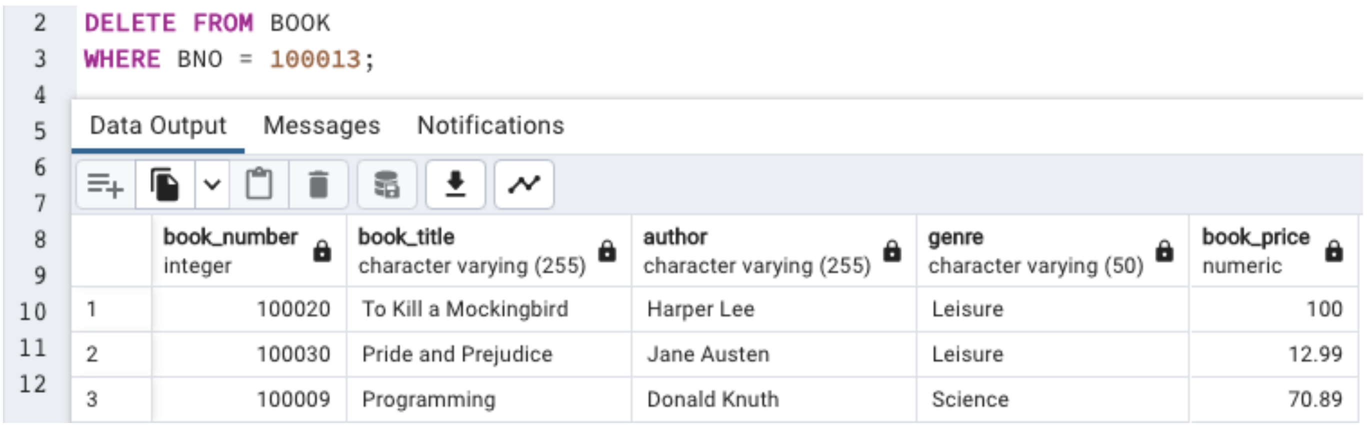
3. **'DELETE' BNO: '100013';**



```
select bno as book_number,
title as book_title,
author, category as Genre,
price as book_price from book;
```

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100020	To Kill a Mockingbird	Harper Lee	Leisure	100
2	100030	Pride and Prejudice	Jane Austen	Leisure	12.99
3	100009	Programming	Donald Knuth	Science	70.89
4	100013	Art History	Donald Knuth	Arts	59.99

OUTPUT

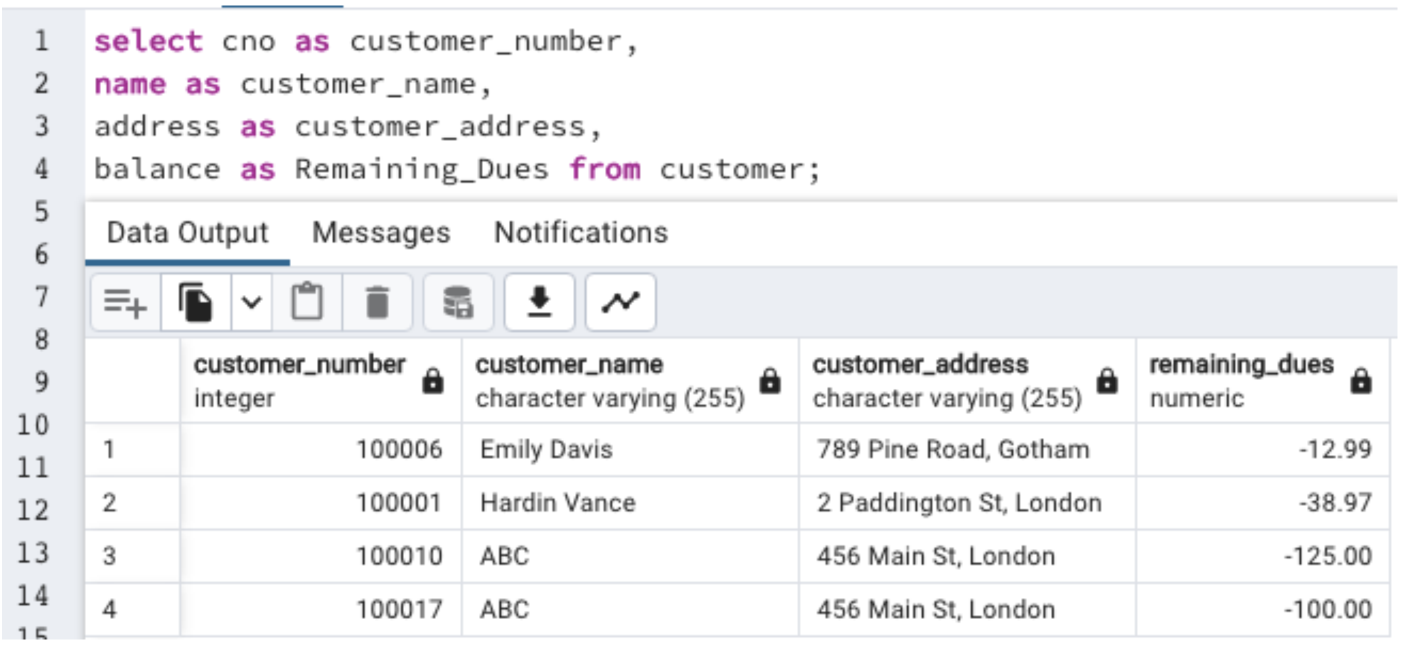


```
DELETE FROM BOOK
WHERE BNO = 100013;
```

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100020	To Kill a Mockingbird	Harper Lee	Leisure	100
2	100030	Pride and Prejudice	Jane Austen	Leisure	12.99
3	100009	Programming	Donald Knuth	Science	70.89

Marks: /2

4. **'UPDATE' Update payment for customer CNO: 100017 with amount £100**

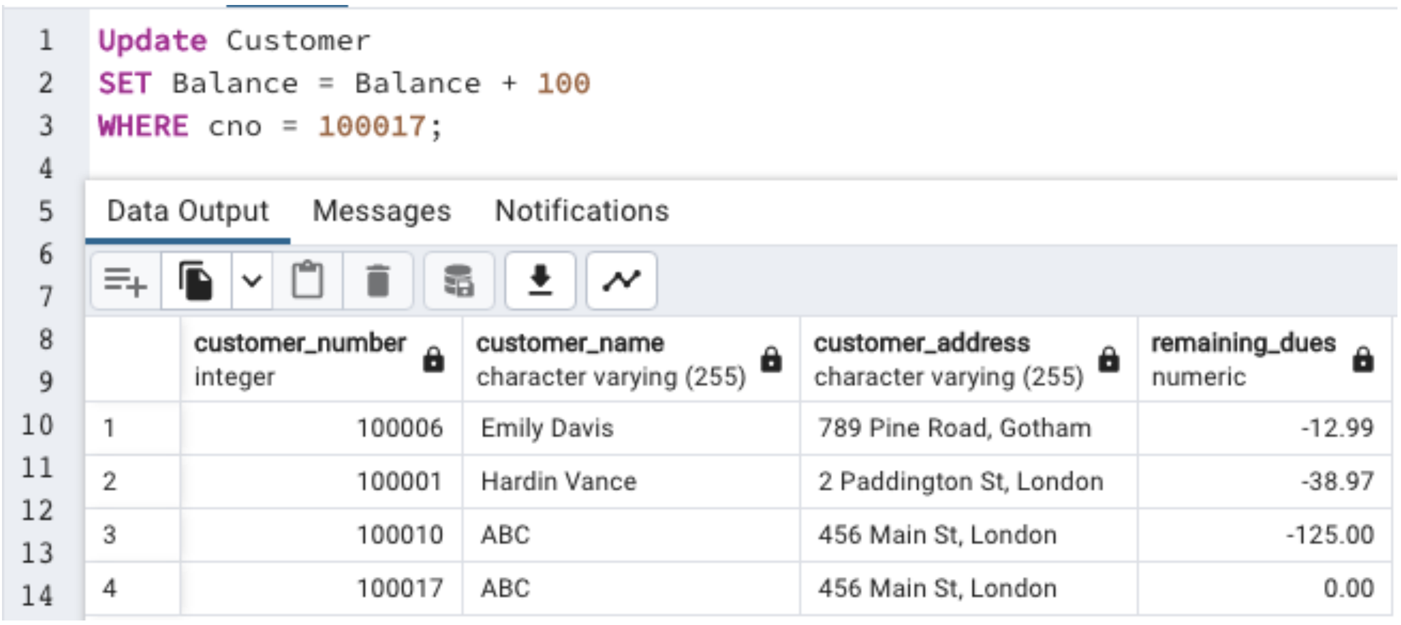


```
select cno as customer_number,
name as customer_name,
address as customer_address,
balance as Remaining_Dues from customer;
```

Data Output Messages Notifications

	customer_number integer	customer_name character varying (255)	customer_address character varying (255)	remaining_dues numeric
1	100006	Emily Davis	789 Pine Road, Gotham	-12.99
2	100001	Hardin Vance	2 Paddington St, London	-38.97
3	100010	ABC	456 Main St, London	-125.00
4	100017	ABC	456 Main St, London	-100.00

OUTPUT



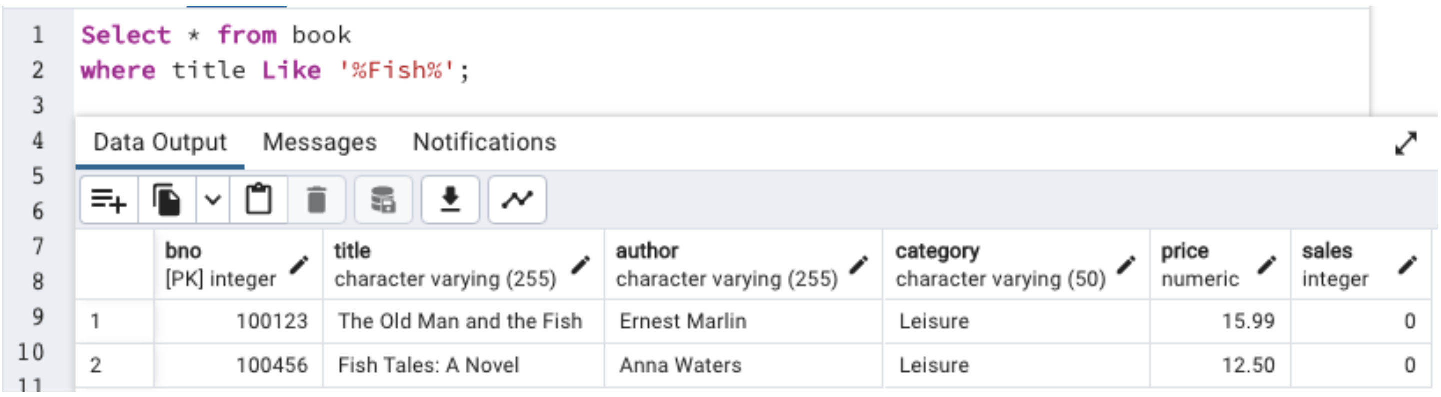
```
Update Customer
SET Balance = Balance + 100
WHERE cno = 100017;
```

Data Output Messages Notifications

	customer_number integer	customer_name character varying (255)	customer_address character varying (255)	remaining_dues numeric
1	100006	Emily Davis	789 Pine Road, Gotham	-12.99
2	100001	Hardin Vance	2 Paddington St, London	-38.97
3	100010	ABC	456 Main St, London	-125.00
4	100017	ABC	456 Main St, London	0.00

Marks: /2

5. Find books with fragment “Fish” in title.

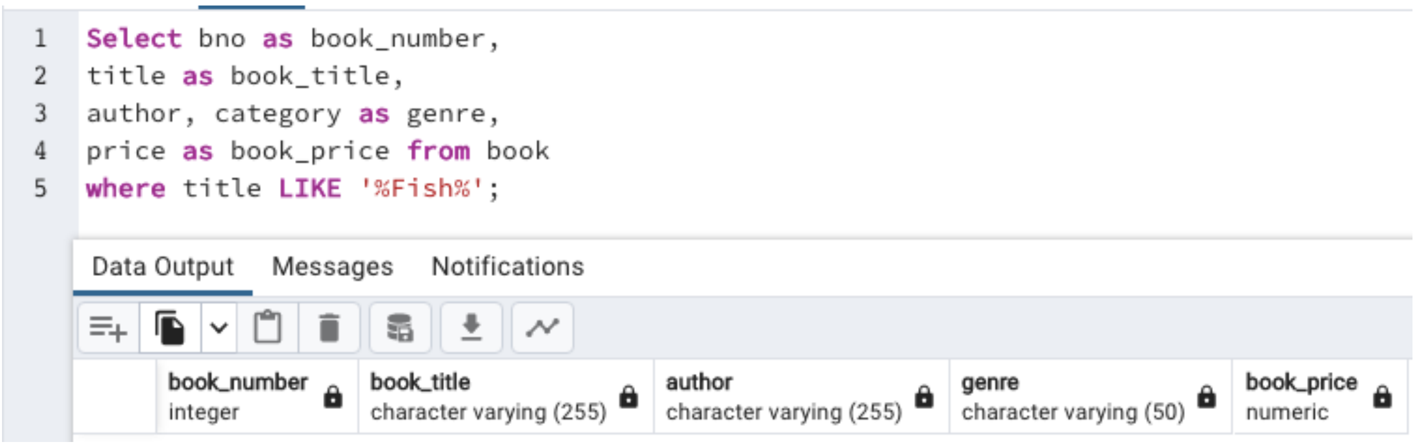


```
Select * from book
where title Like '%Fish%';
```

Data Output | Messages | Notifications

	bno [PK] integer	title character varying (255)	author character varying (255)	category character varying (50)	price numeric	sales integer
1	100123	The Old Man and the Fish	Ernest Marlin	Leisure	15.99	0
2	100456	Fish Tales: A Novel	Anna Waters	Leisure	12.50	0

OUTPUT



```
Select bno as book_number,
title as book_title,
author, category as genre,
price as book_price from book
where title LIKE '%Fish%';
```

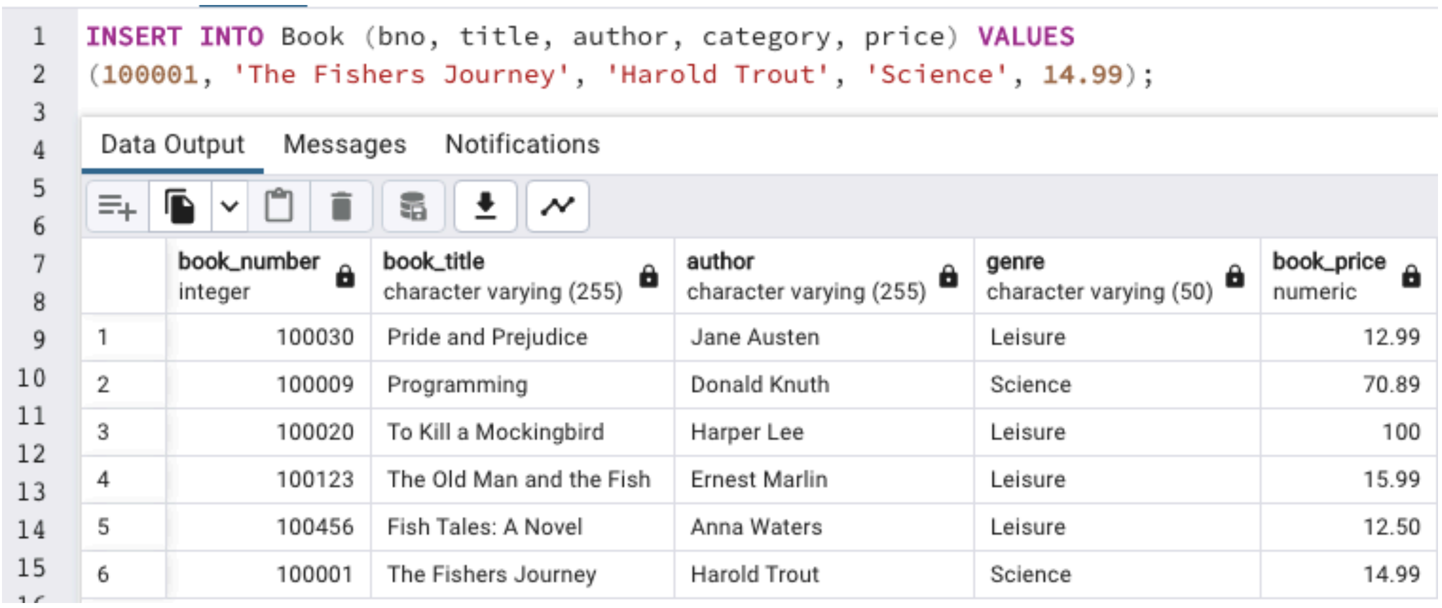
Data Output | Messages | Notifications

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric

Marks: /1

DATE: Friday, 2 December 2022

6. 'DELETE' BNO: '100001'

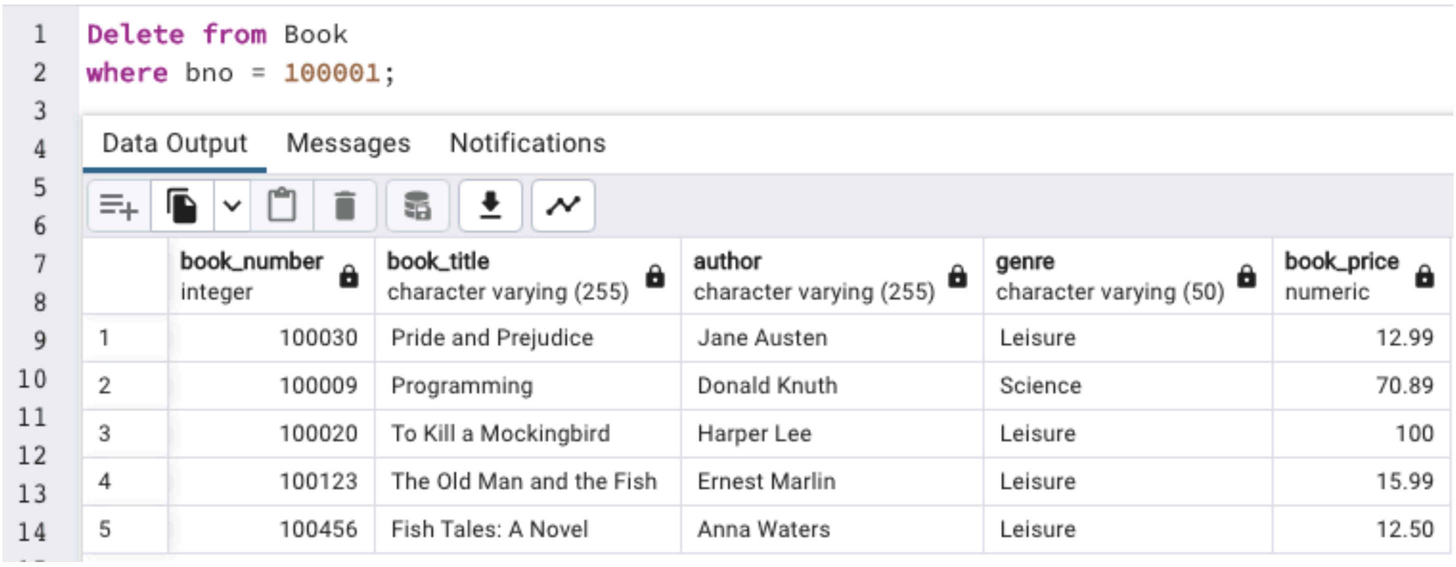


```sql
INSERT INTO Book (bno, title, author, category, price) VALUES
(100001, 'The Fishers Journey', 'Harold Trout', 'Science', 14.99);
```

Data Output | Messages | Notifications

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100030	Pride and Prejudice	Jane Austen	Leisure	12.99
2	100009	Programming	Donald Knuth	Science	70.89
3	100020	To Kill a Mockingbird	Harper Lee	Leisure	100
4	100123	The Old Man and the Fish	Ernest Marlin	Leisure	15.99
5	100456	Fish Tales: A Novel	Anna Waters	Leisure	12.50
6	100001	The Fishers Journey	Harold Trout	Science	14.99

OUTPUT:



```sql
Delete from Book
where bno = 100001;
```

Data Output | Messages | Notifications

	book_number integer	book_title character varying (255)	author character varying (255)	genre character varying (50)	book_price numeric
1	100030	Pride and Prejudice	Jane Austen	Leisure	12.99
2	100009	Programming	Donald Knuth	Science	70.89
3	100020	To Kill a Mockingbird	Harper Lee	Leisure	100
4	100123	The Old Man and the Fish	Ernest Marlin	Leisure	15.99
5	100456	Fish Tales: A Novel	Anna Waters	Leisure	12.50

Marks: /1

Total marks for Part 2: /50

Part 3. Python application program

Python application program communicates with the PostgreSQL server. Write the purpose of the main functions involved in the process.

(Hint: explain psycopg2.connect(), connection.cursor(), cursor.execute(), cursor.fetchall()**)**

psycopg2.connect(): This is a built function in python used for establishing a connection with the PostgreSQL database. The parameter inside the function is a string that contains the database credentials. Imported from a library 'psycopg2'.

open(): A built in function in python used for opening and accessing a file. Several operations like reading, writing etc. could be done.

conn.cursor(): This constructor creates a cursor object. This object is used as an interface between the Python code and the database. The cursor object has several methods to send SQL queries to the database from the python code.

cur.fetchone(): This method of the cursor object retrieves the next row from the result set of a query executed. Each call to the fetchone() function moves the pointer forward and returns one row as a tuple. Here I have used the fetchone() method used for retrieving the result set precisely to assign a value from the result set to a variable.

Split(): This function just splits the string read from a file to substrings based on a delimiter, i.e.('#').

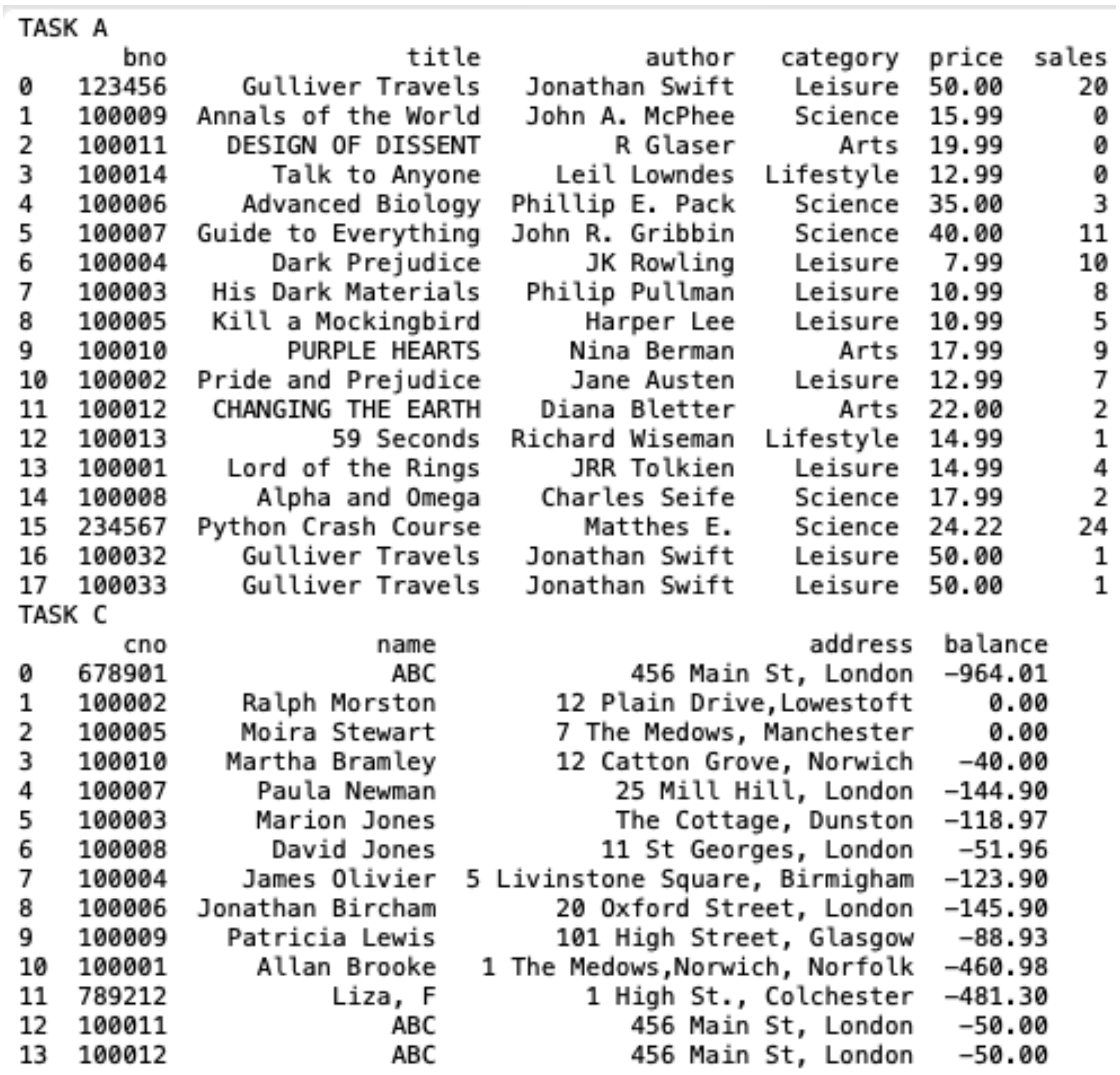
Strip(): Removes the leading and trailing whitespaces from a string. Here the string is from the input.txt file where we apply the strip function for further processes.

Pd.**read_sql_query()**: Reads the result of an SQL query directly into a Pandas Dataframe. This is ideally used to fetch results in a tabular format/Dataframe. Fetches all rows and columns and loads them into the dataframe. Displaying a whole bunch of data, for that read_Sql_query is used. The difference from cur.execute() is that the result will be given in tuples, more granular compared to the former.

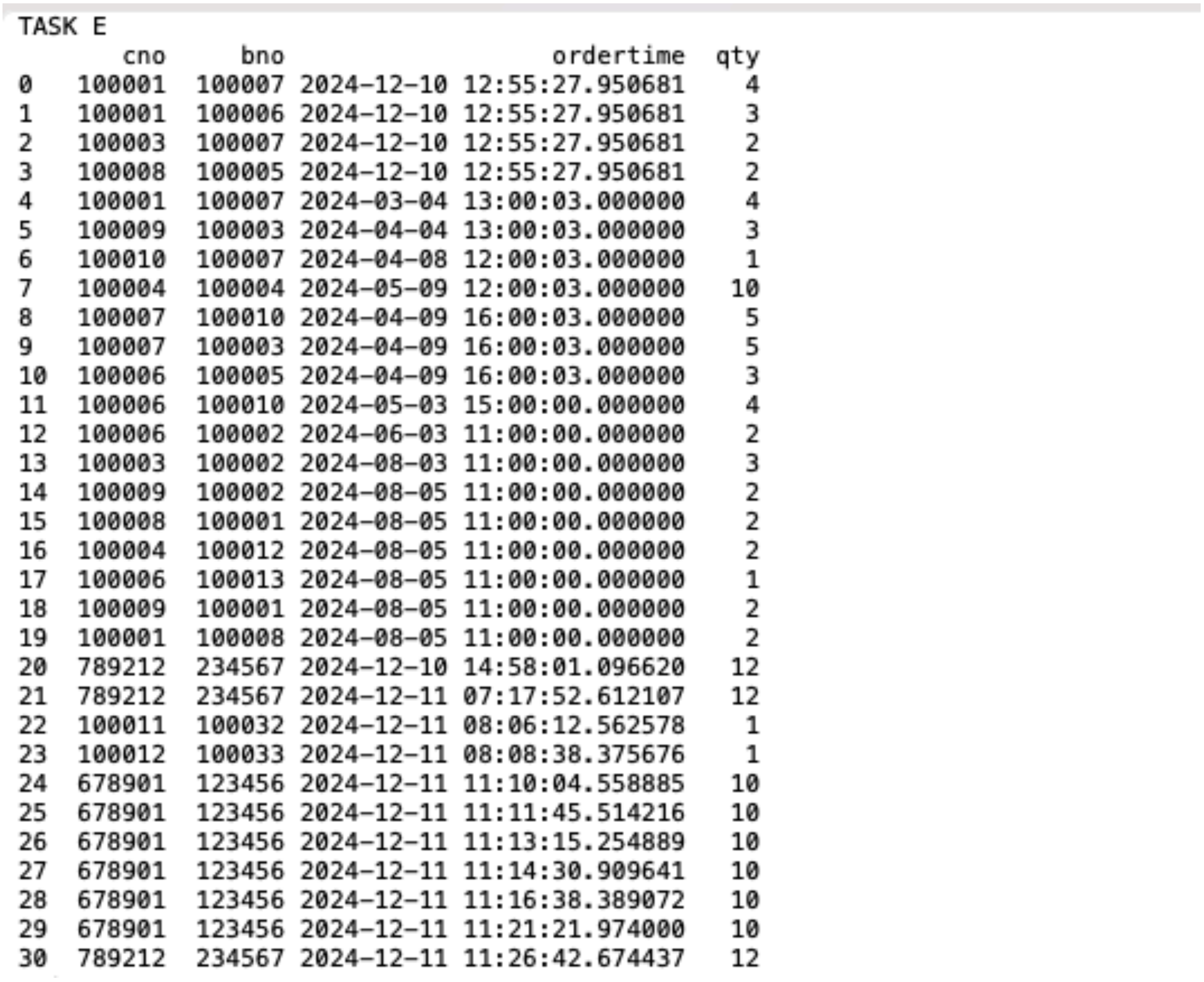
Marks: /4

Python Program outputs

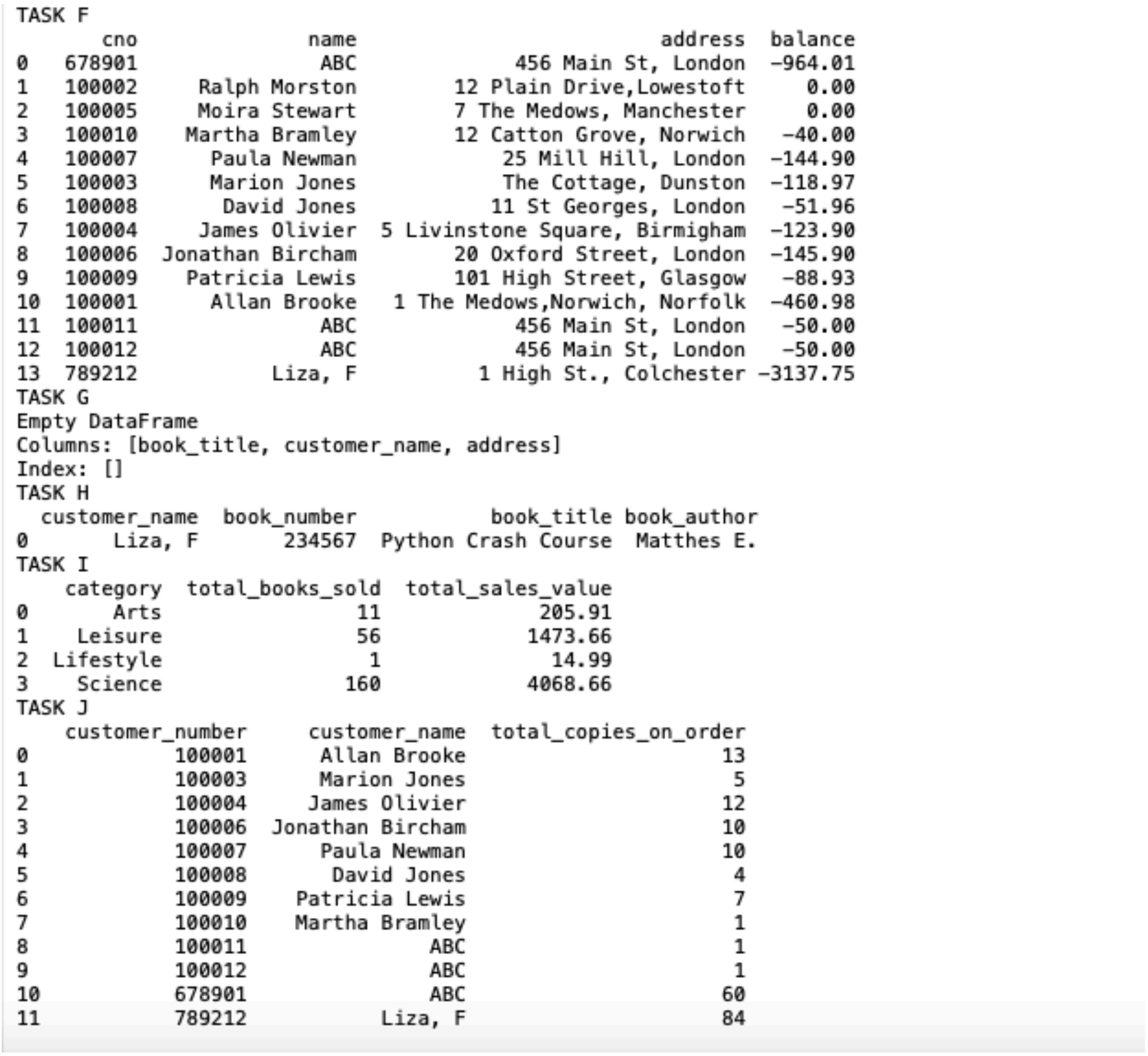
Enter/paste here the output from running input.txt.
(Hint: the content of the **output.txt** file for the given **input.txt** file)



```
TASK A
       bno                 title            author   category  price  sales
0   123456       Gulliver Travels   Jonathan Swift    Leisure  50.00     20
1   100009    Annals of the World    John A. McPhee    Science  15.99      0
2   100011      DESIGN OF DISSENT          R Glaser       Arts  19.99      0
3   100014         Talk to Anyone      Leil Lowndes  Lifestyle  12.99      0
4   100006       Advanced Biology   Phillip E. Pack    Science  35.00      3
5   100007    Guide to Everything   John R. Gribbin    Science  40.00     11
6   100004         Dark Prejudice        JK Rowling    Leisure   7.99     10
7   100003     His Dark Materials   Philip Pullman    Leisure  10.99      8
8   100005      Kill a Mockingbird        Harper Lee    Leisure  10.99      5
9   100010          PURPLE HEARTS       Nina Berman       Arts  17.99      9
10  100002    Pride and Prejudice       Jane Austen    Leisure  12.99      7
11  100012     CHANGING THE EARTH     Diana Bletter       Arts  22.00      2
12  100013             59 Seconds  Richard Wiseman  Lifestyle  14.99      1
13  100001      Lord of the Rings       JRR Tolkien    Leisure  14.99      4
14  100008        Alpha and Omega     Charles Seife    Science  17.99      2
15  234567     Python Crash Course        Matthes E.    Science  24.22     24
16  100032       Gulliver Travels   Jonathan Swift    Leisure  50.00      1
17  100033       Gulliver Travels   Jonathan Swift    Leisure  50.00      1
TASK C
       cno             name                         address  balance
0   678901              ABC              456 Main St, London  -964.01
1   100002    Ralph Morston       12 Plain Drive,Lowestoft     0.00
2   100005    Moira Stewart     7 The Medows, Manchester     0.00
3   100010   Martha Bramley       12 Catton Grove, Norwich   -40.00
4   100007     Paula Newman           25 Mill Hill, London  -144.90
5   100003     Marion Jones         The Cottage, Dunston  -118.97
6   100008      David Jones          11 St Georges, London   -51.96
7   100004    James Olivier  5 Livinstone Square, Birmigham  -123.90
8   100006 Jonathan Bircham       20 Oxford Street, London  -145.90
9   100009   Patricia Lewis       101 High Street, Glasgow   -88.93
10  100001     Allan Brooke    1 The Medows,Norwich, Norfolk  -460.98
11  789212         Liza, F        1 High St., Colchester  -481.30
12  100011              ABC              456 Main St, London   -50.00
13  100012              ABC              456 Main St, London   -50.00
```



```
TASK E
        cno     bno                 ordertime  qty
0    100001  100007 2024-12-10 12:55:27.950681    4
1    100001  100006 2024-12-10 12:55:27.950681    3
2    100003  100007 2024-12-10 12:55:27.950681    2
3    100008  100005 2024-12-10 12:55:27.950681    2
4    100001  100007 2024-03-04 13:00:03.000000    4
5    100009  100003 2024-04-04 13:00:03.000000    3
6    100010  100007 2024-04-08 12:00:03.000000    1
7    100004  100004 2024-05-09 12:00:03.000000   10
8    100007  100010 2024-04-09 16:00:03.000000    5
9    100007  100003 2024-04-09 16:00:03.000000    5
10   100006  100005 2024-04-09 16:00:03.000000    3
11   100006  100010 2024-05-03 15:00:00.000000    4
12   100006  100002 2024-06-03 11:00:00.000000    2
13   100003  100002 2024-08-03 11:00:00.000000    3
14   100009  100002 2024-08-05 11:00:00.000000    2
15   100008  100001 2024-08-05 11:00:00.000000    2
16   100004  100012 2024-08-05 11:00:00.000000    2
17   100006  100013 2024-08-05 11:00:00.000000    1
18   100009  100001 2024-08-05 11:00:00.000000    2
19   100001  100008 2024-08-05 11:00:00.000000    2
20   789212  234567 2024-12-10 14:58:01.096620   12
21   789212  234567 2024-12-11 07:17:52.612107   12
22   100011  100032 2024-12-11 08:06:12.562578    1
23   100012  100033 2024-12-11 08:08:38.375676    1
24   678901  123456 2024-12-11 11:10:04.558885   10
25   678901  123456 2024-12-11 11:11:45.514216   10
26   678901  123456 2024-12-11 11:13:15.254889   10
27   678901  123456 2024-12-11 11:14:30.909641   10
28   678901  123456 2024-12-11 11:16:38.389072   10
29   678901  123456 2024-12-11 11:21:21.974000   10
30   789212  234567 2024-12-11 11:26:42.674437   12
```



```
TASK F
       cno               name                            address  balance
0   678901                ABC                 456 Main St, London  -964.01
1   100002       Ralph Morston          12 Plain Drive,Lowestoft     0.00
2   100005       Moira Stewart          7 The Medows, Manchester     0.00
3   100010     Martha Bramley          12 Catton Grove, Norwich    -40.00
4   100007        Paula Newman              25 Mill Hill, London   -144.90
5   100003        Marion Jones            The Cottage, Dunston   -118.97
6   100008         David Jones            11 St Georges, London    -51.96
7   100004      James Olivier  5 Livinstone Square, Birmigham   -123.90
8   100006  Jonathan Bircham          20 Oxford Street, London   -145.90
9   100009     Patricia Lewis          101 High Street, Glasgow    -88.93
10  100001        Allan Brooke   1 The Medows,Norwich, Norfolk   -460.98
11  100011                ABC                 456 Main St, London    -50.00
12  100012                ABC                 456 Main St, London    -50.00
13  789212            Liza, F            1 High St., Colchester -3137.75
TASK G
Empty DataFrame
Columns: [book_title, customer_name, address]
Index: []
TASK H
  customer_name  book_number           book_title book_author
0       Liza, F       234567  Python Crash Course  Matthes E.
TASK I
    category  total_books_sold  total_sales_value
0       Arts                11             205.91
1    Leisure                56            1473.66
2  Lifestyle                 1              14.99
3    Science               160            4068.66
TASK J
    customer_number     customer_name  total_copies_on_order
0            100001      Allan Brooke                     13
1            100003      Marion Jones                      5
2            100004     James Olivier                     12
3            100006  Jonathan Bircham                     10
4            100007      Paula Newman                     10
5            100008       David Jones                      4
6            100009    Patricia Lewis                      7
7            100010    Martha Bramley                      1
8            100011               ABC                      1
9            100012               ABC                      1
10           678901               ABC                     60
11           789212           Liza, F                     84
```

Marks: /16

TOTAL PYTHON MARKS /20

Submission Quality:

The quality of the submission is according to the instructions: /5

Total Mark Distribution:

MARKS DDL section	**/25**
MARKS DML section	**/50**
MARKS FOR Python Application	**/20**
Marks for quality of submission	**/5**
TOTAL MARKS	**/100**

DATE: Friday, 2 December 2022

APPENDIX: EXAMPLE OF HOW TO FILL THIS FORM/DOCUMENT (using different indicative tasks).

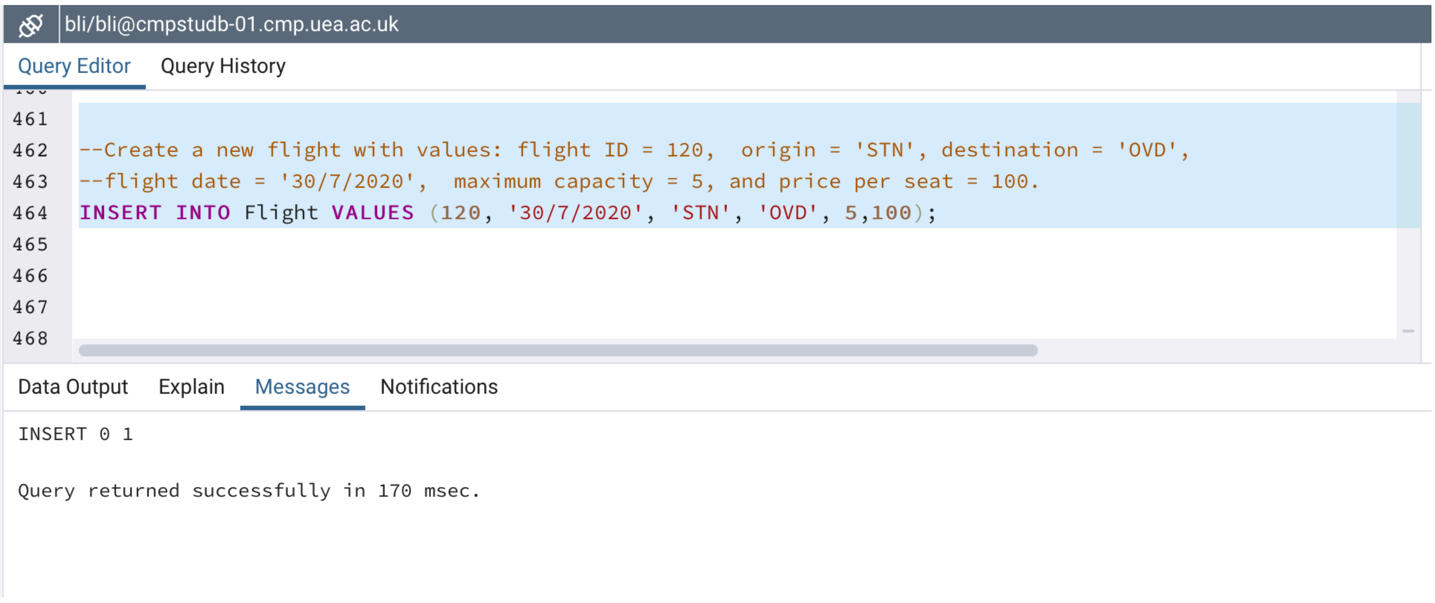
Testing task 1

---

1. Given flight details create new flight record

a) Create a new flight with values: flight ID = **120**, origin = **'STN'**, destination = **'OVD'**, flight date = **'30/7/2020'**, maximum capacity = **5**, and price per seat = **100**.

INSERT YOUR SQL QUERY AND OUTPUT HERE

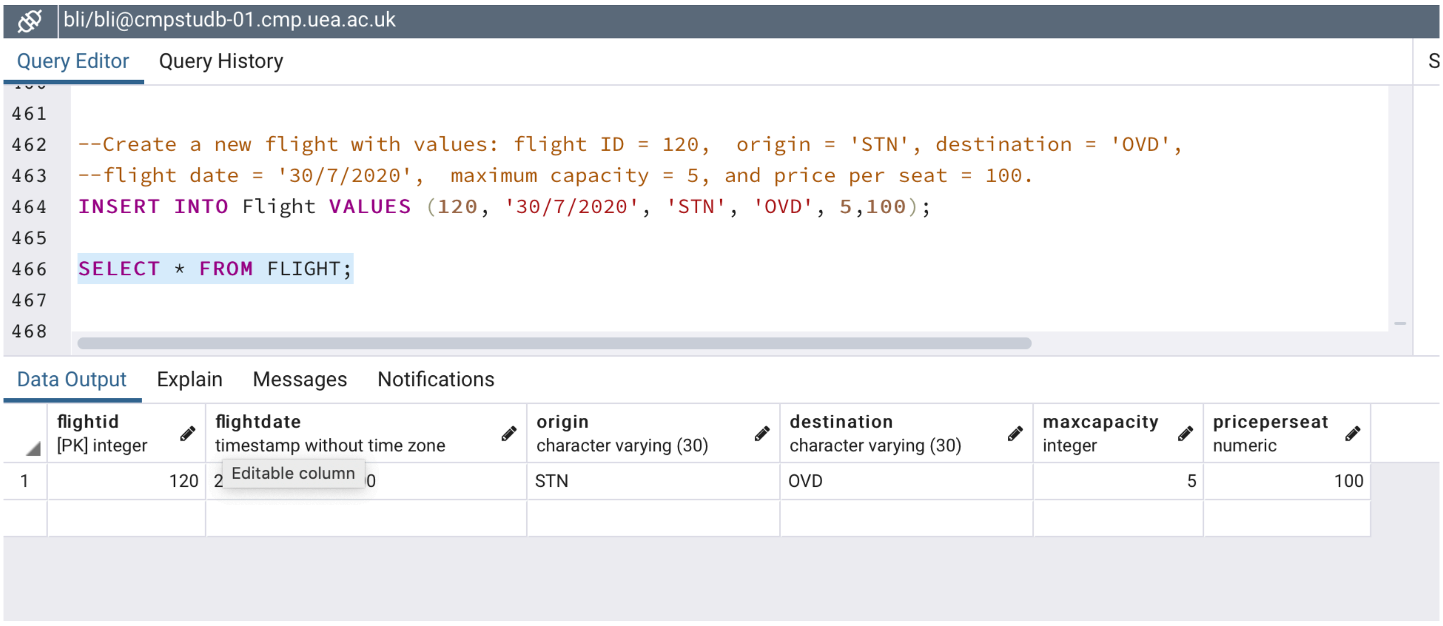


```
--Create a new flight with values: flight ID = 120,  origin = 'STN', destination = 'OVD',
--flight date = '30/7/2020',  maximum capacity = 5, and price per seat = 100.
INSERT INTO Flight VALUES (120, '30/7/2020', 'STN', 'OVD', 5,100);
```

```
INSERT 0 1

Query returned successfully in 170 msec.
```

SHOW THE CONTENTS OF FLIGHT TABLE AT THE END OF THE TASK



```
--Create a new flight with values: flight ID = 120,  origin = 'STN', destination = 'OVD',
--flight date = '30/7/2020',  maximum capacity = 5, and price per seat = 100.
INSERT INTO Flight VALUES (120, '30/7/2020', 'STN', 'OVD', 5,100);

SELECT * FROM FLIGHT;
```

	flightid [PK] integer	flightdate timestamp without time zone	origin character varying (30)	destination character varying (30)	maxcapacity integer	priceperseat numeric
1	120	2 [illegible] 0	STN	OVD	5	100

DATE: Friday, 2 December 2022

2. Produce a query that counts the number of tuples in each table.

INSERT YOUR SQL QUERY AND OUTPUT HERE