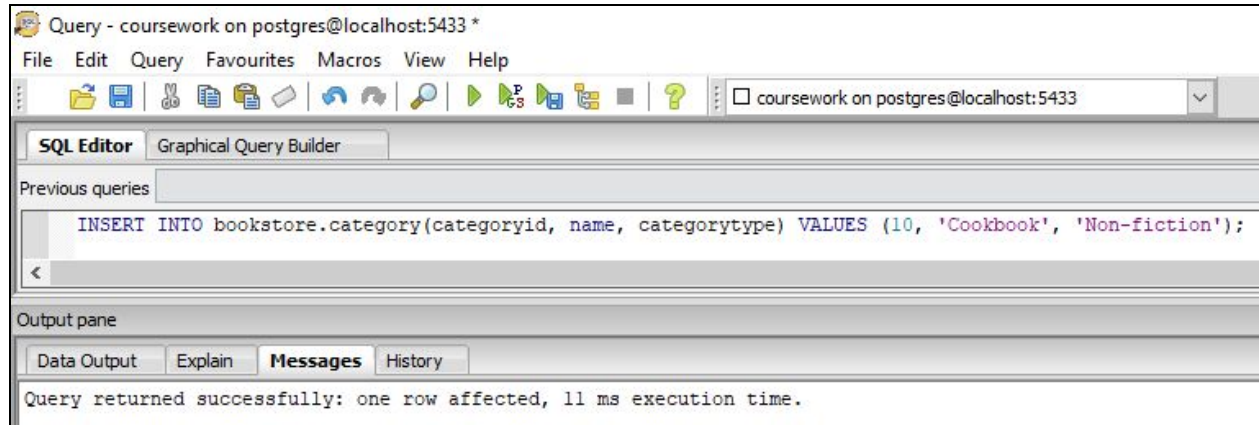


SQL Statements: Testing Evidence

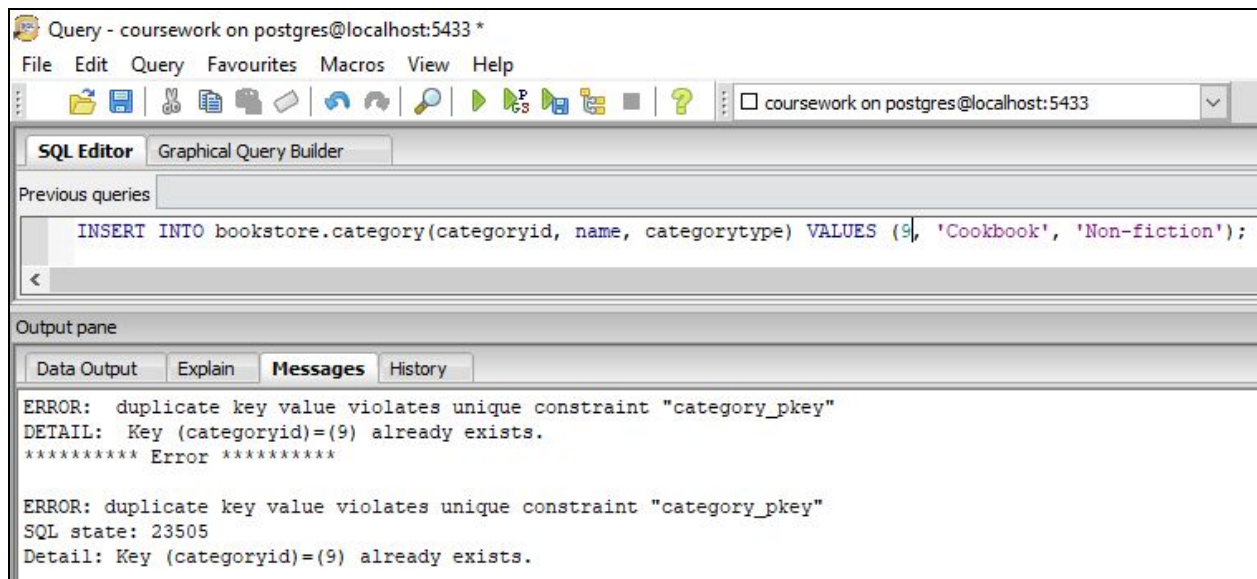
Task 1

“Given a category ID, name and type, create a new category”.

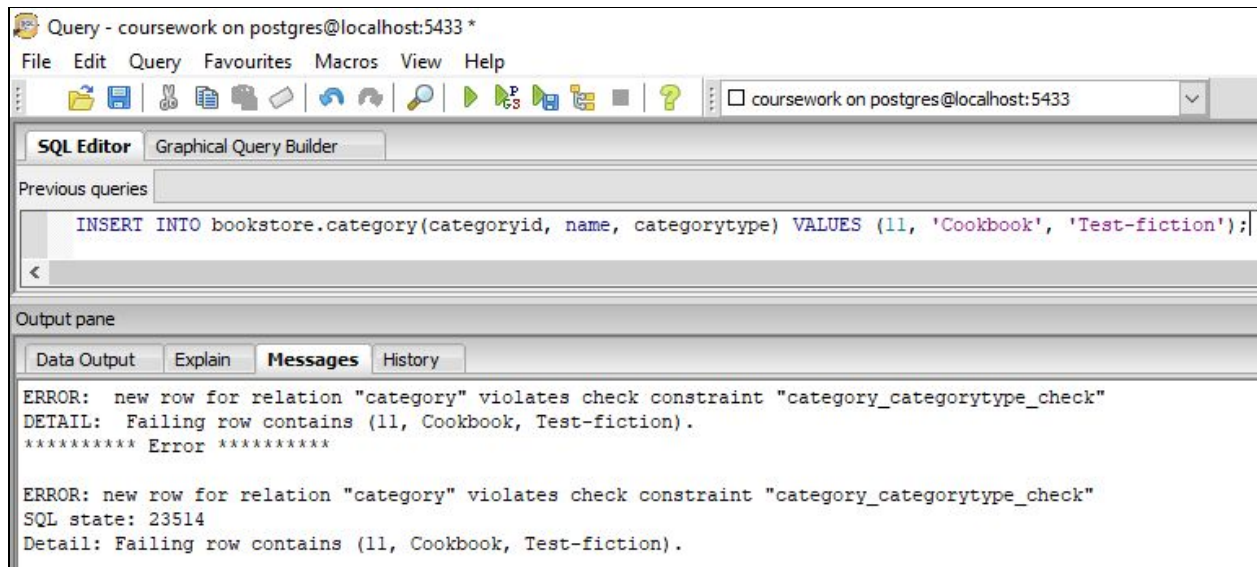
INSERT INTO bookstore.category(categoryid, name, categorytype) VALUES (10, 'Cookbook', 'Non-fiction');



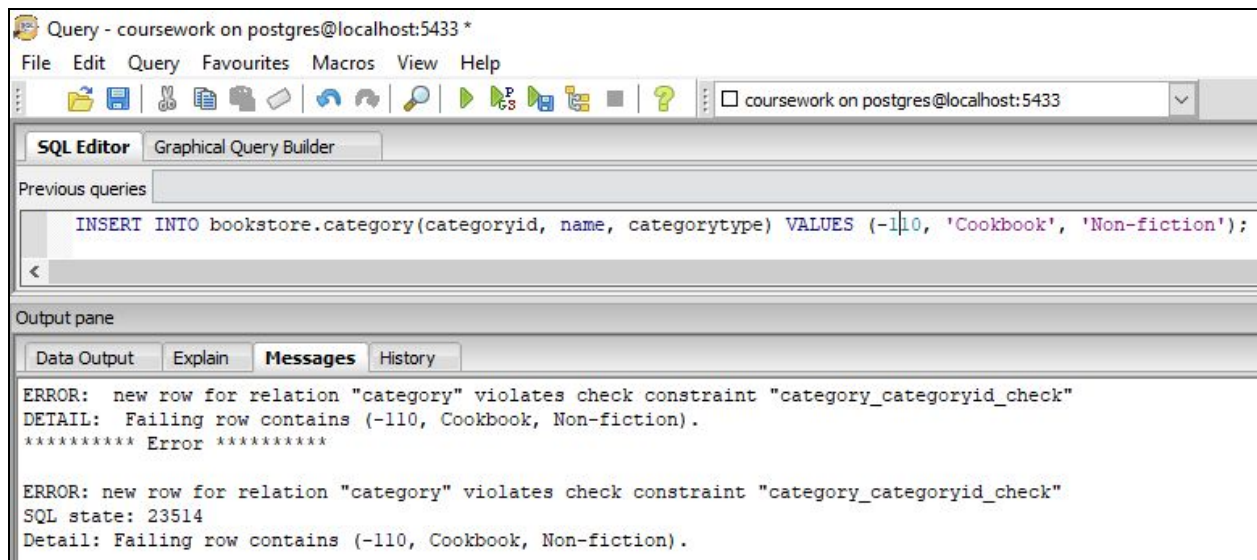
INSERT INTO bookstore.category(categoryid, name, categorytype) VALUES (9, 'Cookbook', 'Non-fiction');



INSERT INTO bookstore.category(categoryid, name, categorytype) VALUES (11, 'Cookbook', 'Test-fiction');



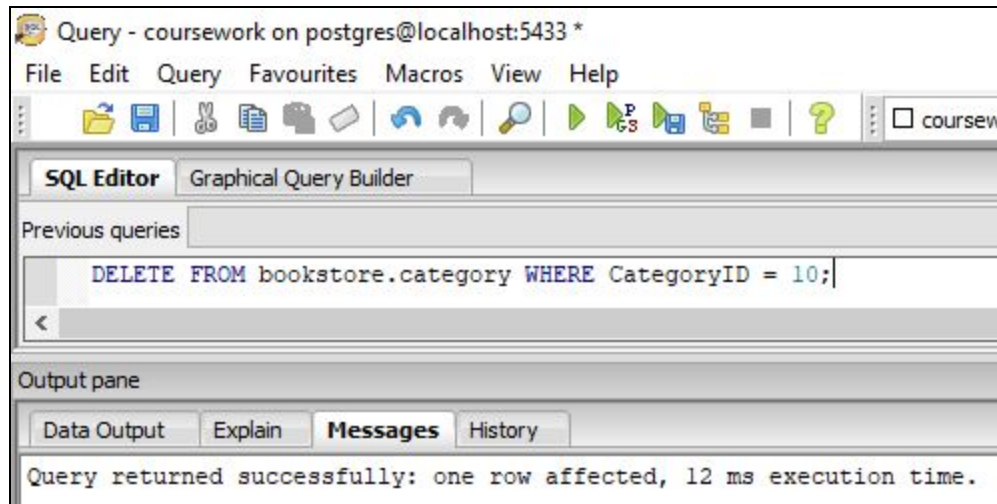
INSERT INTO bookstore.category(categoryid, name, categorytype) VALUES (-110, 'Cookbook', 'Non-fiction');



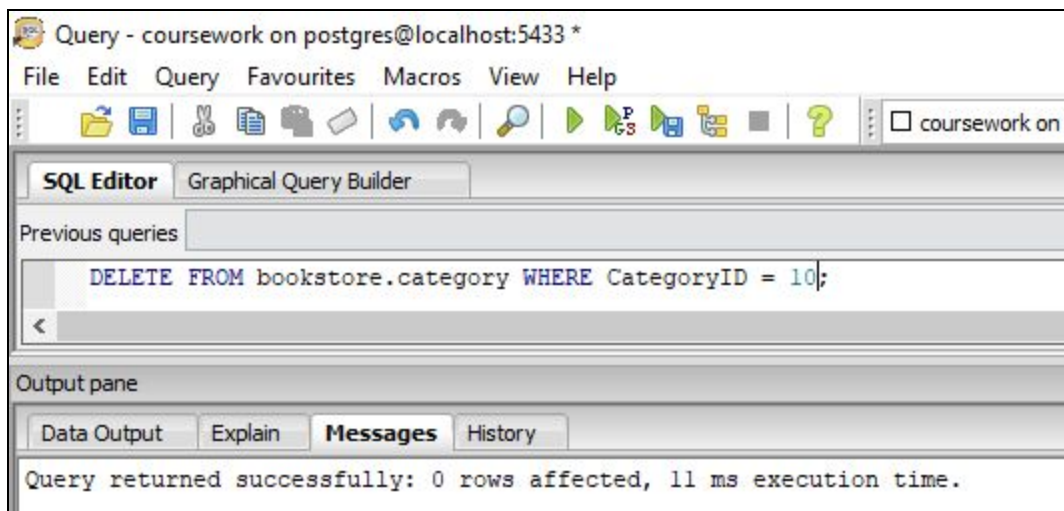
Task 2

“Given a category ID, remove the record for that category.”

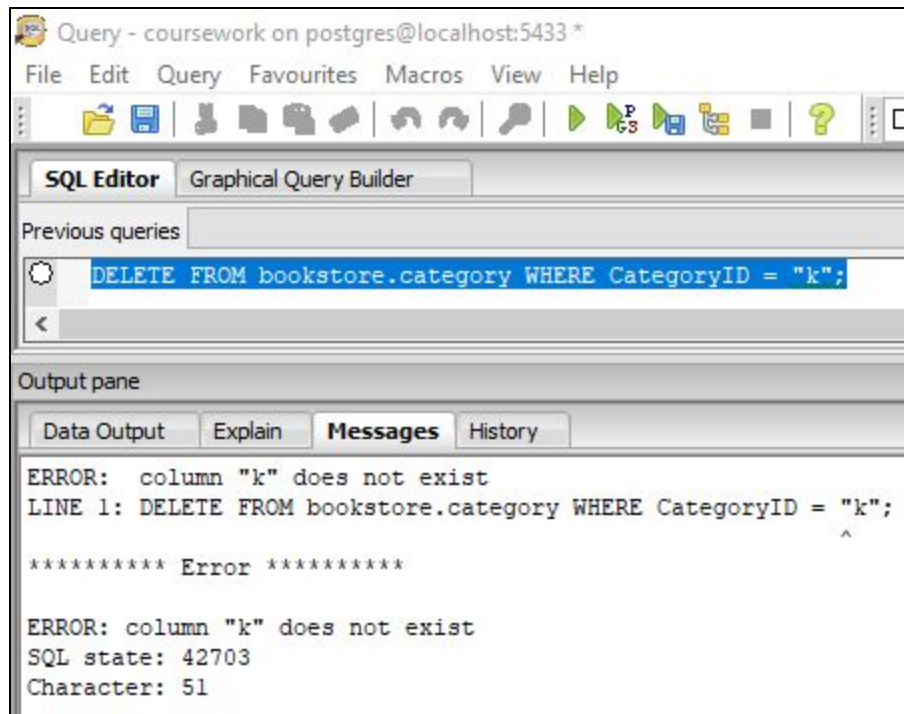
DELETE FROM bookstore.category WHERE CategoryID = 10;



DELETE FROM bookstore.category WHERE CategoryID = 10;



DELETE FROM bookstore.category WHERE CategoryID = "k";



Task 3

“Produce a summary report of books available in each category. The report should include the number of book titles and the average price in each category as well as an appropriate report header and a summary line with totals (hint: summary line may be produced by a separate query). Format your field values appropriately.”

Main Query

```
SELECT name AS category, count(*) AS numberofbooksincategory, ROUND(AVG(price), 2) AS
averagepriceofcategory
FROM bookstore.category, bookstore.book
WHERE category.categoryid = book.categoryid
GROUP BY category.categoryid;
```

The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The SQL Editor tab is active, displaying the following query:

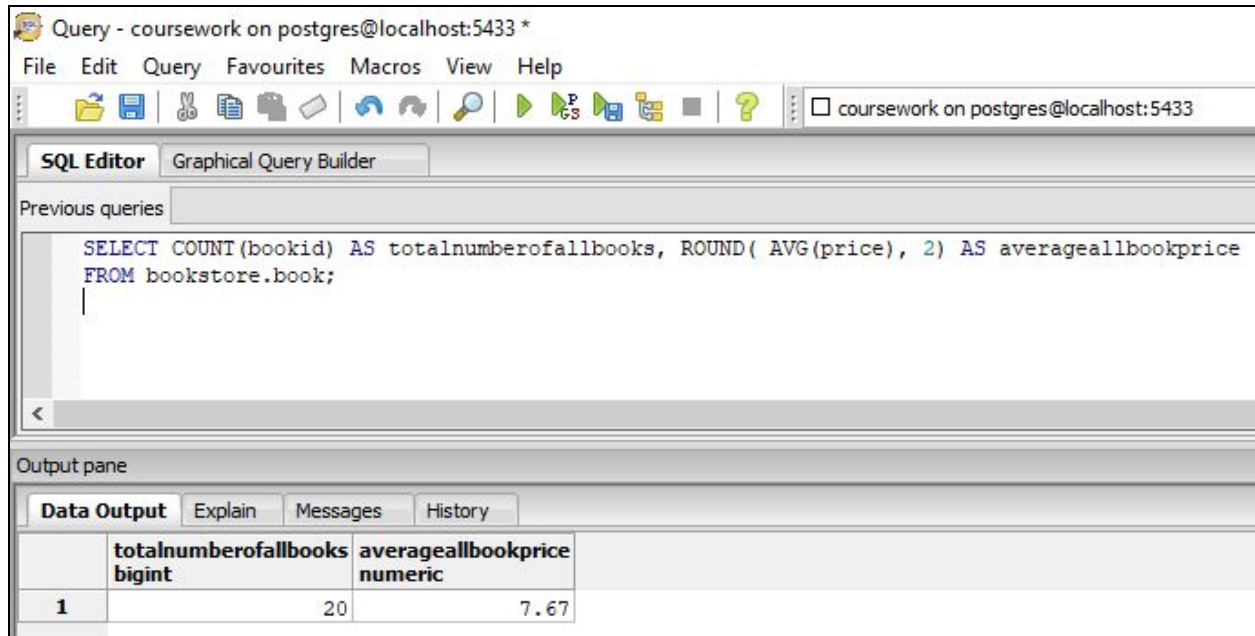
```
SELECT name AS category, count(*) AS numberofbooksincategory, ROUND(AVG(price), 2) AS averagepriceofcategory
FROM bookstore.category, bookstore.book
WHERE category.categoryid = book.categoryid
GROUP BY category.categoryid
```

The Output pane at the bottom shows the results of the query in a table format. The table has four columns: an index, category, numberofbooksincategory, and averagepriceofcategory. The results are as follows:

	category character varying(50)	numberofbooksincategory bigint	averagepriceofcategory numeric
1	Encyclopedia	1	20.95
2	Science-fiction	2	7.99
3	Science	2	7.70
4	Mystery	3	7.64
5	Mystery	2	7.50
6	Drama	2	3.99
7	Mathematics	4	7.96
8	Romance	1	3.95
9	History	2	7.95
10	Dictionary	1	3.45

Summary Line Query

```
SELECT COUNT(bookid) AS totalnumberofallbooks, ROUND( AVG(price), 2) AS  
averageallbookprice  
FROM bookstore.book;
```



The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar with various icons. The "SQL Editor" tab is active, and the query text is as follows:

```
SELECT COUNT(bookid) AS totalnumberofallbooks, ROUND( AVG(price), 2) AS averageallbookprice  
FROM bookstore.book;
```

Below the query editor is the "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is selected, displaying the results of the query in a table format:

	totalnumberofallbooks bigint	averageallbookprice numeric
1	20	7.67

Task 4

“Given a publisher name, produce a report of books ordered by year and month. For each year and month the report should show bookid, title, total number of orders for the title, total quantity and total selling value (both order value and retail value).”

“SELECT

```
TO_CHAR(orderdate, 'Mon') AS month,
extract(year FROM orderdate) AS year,
orderid.bookid, title, COUNT(orderid.bookid) AS totalordersinmonth,
SUM(quantity) AS totalbooksorderedinmonth,
SUM(unitsellingprice * quantity) AS totalordervalueformonth,
SUM(price * quantity) AS totalretailvalueformonth
```

```
FROM bookstore.shoporderid, bookstore.orderid, bookstore.book, bookstore.publisher
WHERE shoporderid.shoporderid = orderid.shoporderid AND orderid.bookid = book.bookid
AND book.publisherid = publisher.publisherid AND name = 'Oxford University Press'
GROUP BY TO_CHAR(orderdate, 'Mon'), extract(year FROM orderdate), title, orderid.bookid
ORDER BY 1, 2”
```

Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT
TO_CHAR(orderdate, 'Mon') AS month,
extract(year FROM orderdate) AS year,
orderid.bookid, title, COUNT(orderid.bookid) AS totalordersinmonth,
SUM(quantity) AS totalbooksorderedinmonth,
SUM(unitsellingprice * quantity) AS totalordervalueformonth,
SUM(price * quantity) AS totalretailvalueformonth

FROM bookstore.shoporderid, bookstore.orderid, bookstore.book, bookstore.publisher
WHERE shoporderid.shoporderid = orderid.shoporderid AND orderid.bookid = book.bookid AND book.publisherid = publisher.publisherid AND name = 'Oxford University Press'
GROUP BY TO_CHAR(orderdate, 'Mon'), extract(year FROM orderdate), title, orderid.bookid
ORDER BY 1, 2
```

Output pane

	month	year	bookid	title	totalordersinmonth	totalbooksorderedinmonth	totalordervalueformonth	totalretailvalueformonth
	text	double precision	integer	character varying(50)	bigint	bigint	numeric	numeric
1	Aug	2017	8	Colony One Mars	1	15	75.00	134.85
2	Aug	2017	3	The Silent Wife	1	5	7.50	24.95
3	Dec	2017	15	Basic English Dictionary	1	20	45.00	69.00
4	Feb	2017	13	"A" is for Alibi	1	30	60.00	148.80
5	Feb	2017	3	The Silent Wife	1	5	7.50	24.95
6	Jan	2017	8	Colony One Mars	4	70	350.00	629.30
7	Jul	2017	3	The Silent Wife	2	65	97.50	324.35
8	Jun	2017	15	Basic English Dictionary	2	45	101.25	155.25
9	May	2017	15	Basic English Dictionary	1	15	33.75	51.75
10	Sep	2017	8	Colony One Mars	1	5	25.00	44.95

SELECT

```

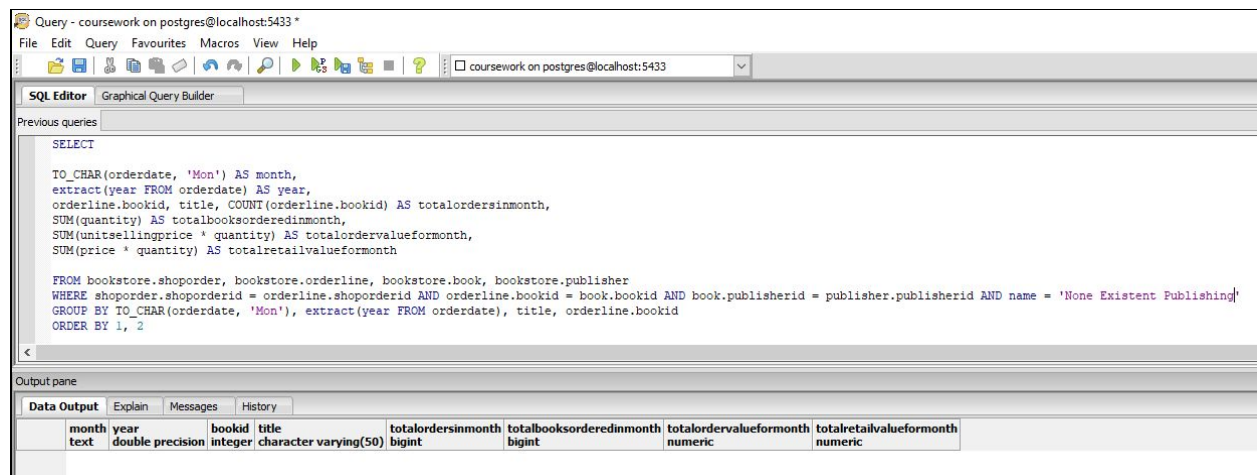
TO_CHAR(orderdate, 'Mon') AS month,
extract(year FROM orderdate) AS year,
orderline.bookid, title, COUNT(orderline.bookid) AS totalordersinmonth,
SUM(quantity) AS totalbooksorderedinmonth,
SUM(unitsellingprice * quantity) AS totalordervalueformonth,
SUM(price * quantity) AS totalretailvalueformonth

```

```

FROM bookstore.shoporder, bookstore.orderline, bookstore.book, bookstore.publisher
WHERE shoporder.shoporderid = orderline.shoporderid AND orderline.bookid = book.bookid
AND book.publisherid = publisher.publisherid AND name = 'None Existent Publishing'
GROUP BY TO_CHAR(orderdate, 'Mon'), extract(year FROM orderdate), title, orderline.bookid
ORDER BY 1, 2

```



SELECT

```

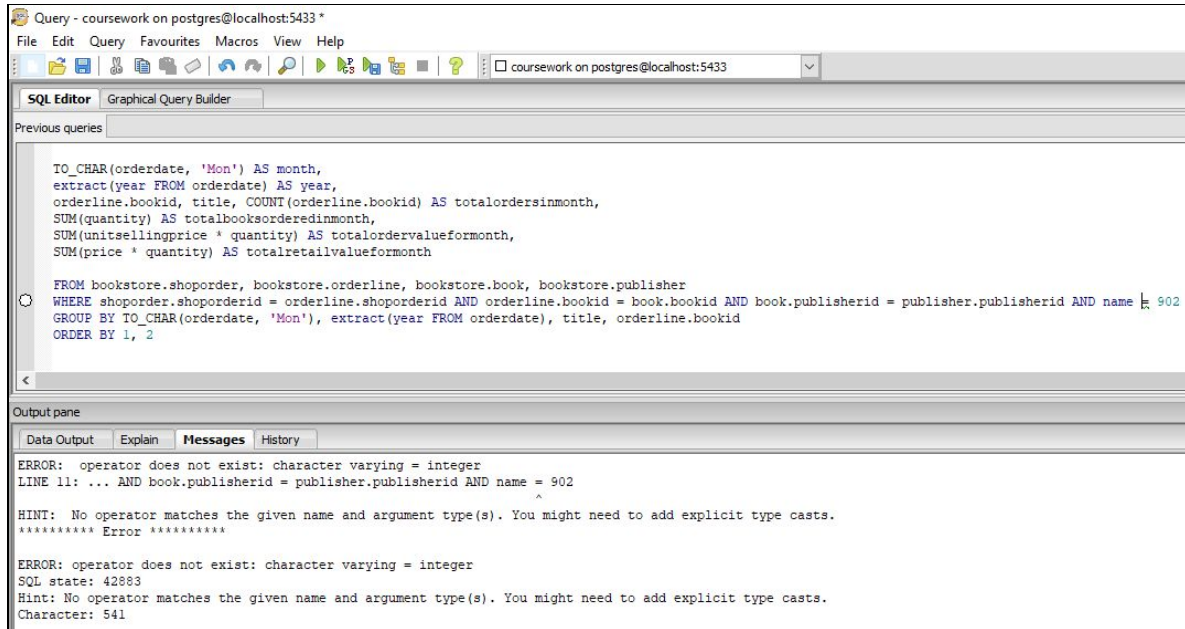
TO_CHAR(orderdate, 'Mon') AS month,
extract(year FROM orderdate) AS year,
orderline.bookid, title, COUNT(orderline.bookid) AS totalordersinmonth,
SUM(quantity) AS totalbooksorderedinmonth,
SUM(unitsellingprice * quantity) AS totalordervalueformonth,
SUM(price * quantity) AS totalretailvalueformonth

```

```

FROM bookstore.shoporder, bookstore.orderline, bookstore.book, bookstore.publisher
WHERE shoporder.shoporderid = orderline.shoporderid AND orderline.bookid = book.bookid
AND book.publisherid = publisher.publisherid AND name = 902
GROUP BY TO_CHAR(orderdate, 'Mon'), extract(year FROM orderdate), title, orderline.bookid
ORDER BY 1, 2

```

The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The SQL Editor tab is active, displaying the following query:

```
TO_CHAR(orderdate, 'Mon') AS month,
extract(year FROM orderdate) AS year,
orderline.bookid, title, COUNT(orderline.bookid) AS totalordersinmonth,
SUM(quantity) AS totalbooksorderedinmonth,
SUM(unitssellingprice * quantity) AS totalordervalueformonth,
SUM(price * quantity) AS totalretailvalueformonth

FROM bookstore.shoporder, bookstore.orderline, bookstore.book, bookstore.publisher
WHERE shoporder.shoporderid = orderline.shoporderid AND orderline.bookid = book.bookid AND book.publisherid = publisher.publisherid AND name = 902
GROUP BY TO_CHAR(orderdate, 'Mon'), extract(year FROM orderdate), title, orderline.bookid
ORDER BY 1, 2
```

The Output pane at the bottom shows the following error messages:

```
ERROR: operator does not exist: character varying = integer
LINE 11: ... AND book.publisherid = publisher.publisherid AND name = 902
                                     ^
HINT: No operator matches the given name and argument type(s). You might need to add explicit type casts.
***** Error *****

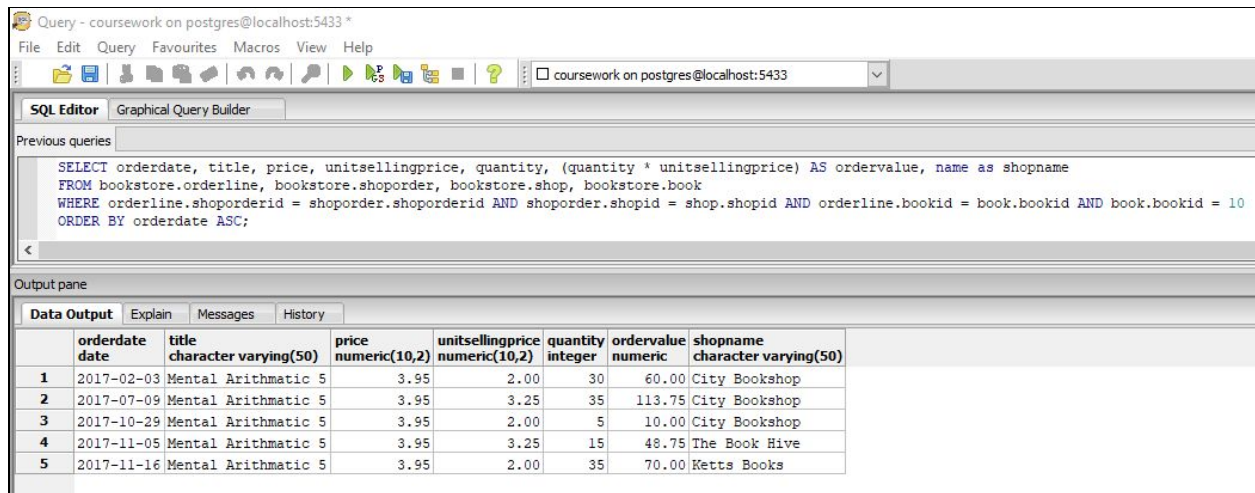
ERROR: operator does not exist: character varying = integer
SQL state: 42883
Hint: No operator matches the given name and argument type(s). You might need to add explicit type casts.
Character: 541
```

Task 5

“Given a book ID, produce the order history (i.e. all order lines) for that book. The query should include order date, order title, price, unitselling price, total quantity, order value and shop name. Include a summary line showing the total number of copies ordered and the total selling value (hint: summary line may be produced by a separate query).”

Main Query

```
SELECT orderdate, title, price, unitsellingprice, quantity, (quantity * unitsellingprice) AS
ordervalue, name as shopname
FROM bookstore.orderline, bookstore.shoporder, bookstore.shop, bookstore.book
WHERE orderline.shoporderid = shoporder.shoporderid AND shoporder.shopid = shop.shopid
AND orderline.bookid = book.bookid AND book.bookid = 10
ORDER BY orderdate ASC;
```



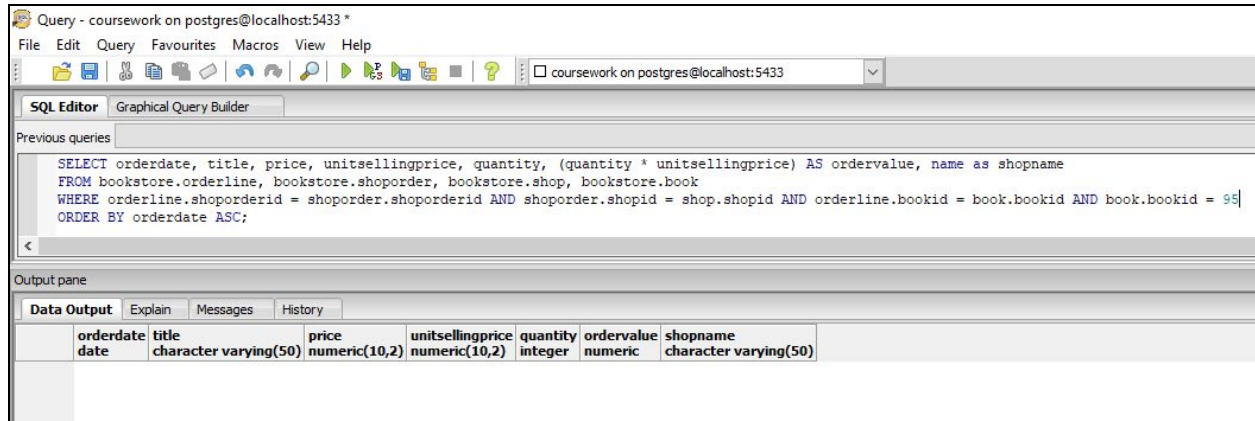
The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The SQL Editor tab is active, displaying the following query:

```
SELECT orderdate, title, price, unitsellingprice, quantity, (quantity * unitsellingprice) AS ordervalue, name as shopname
FROM bookstore.orderline, bookstore.shoporder, bookstore.shop, bookstore.book
WHERE orderline.shoporderid = shoporder.shoporderid AND shoporder.shopid = shop.shopid AND orderline.bookid = book.bookid AND book.bookid = 10
ORDER BY orderdate ASC;
```

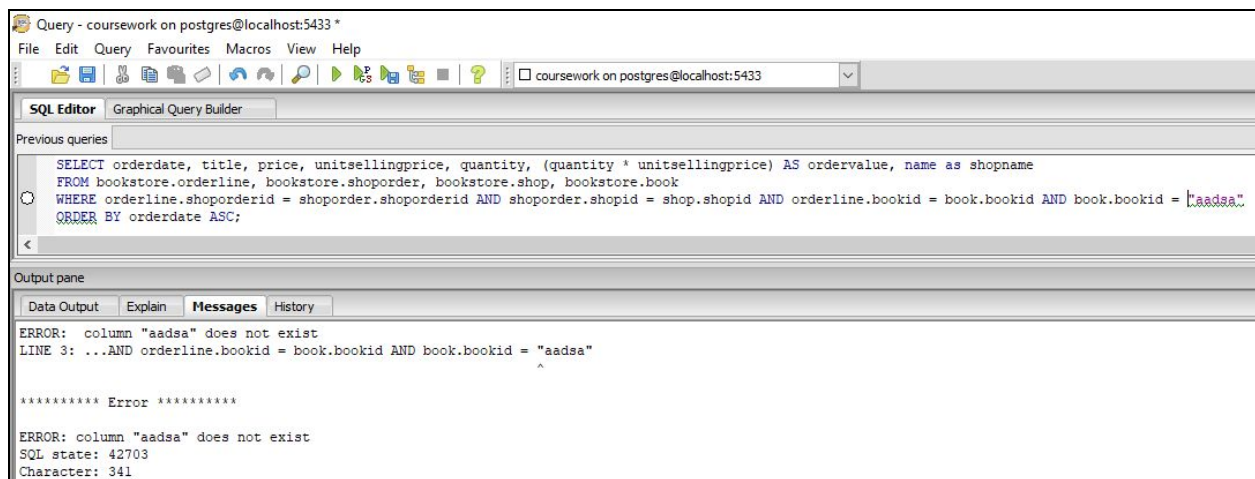
The Output pane shows the results of the query in a table format. The table has 8 columns: orderdate, title, price, unitsellingprice, quantity, ordervalue, and shopname. The results are as follows:

	orderdate	title	price	unitsellingprice	quantity	ordervalue	shopname
	date	character varying(50)	numeric(10,2)	numeric(10,2)	integer	numeric	character varying(50)
1	2017-02-03	Mental Arithmetic 5	3.95	2.00	30	60.00	City Bookshop
2	2017-07-09	Mental Arithmetic 5	3.95	3.25	35	113.75	City Bookshop
3	2017-10-29	Mental Arithmetic 5	3.95	2.00	5	10.00	City Bookshop
4	2017-11-05	Mental Arithmetic 5	3.95	3.25	15	48.75	The Book Hive
5	2017-11-16	Mental Arithmetic 5	3.95	2.00	35	70.00	Ketts Books

```
SELECT orderdate, title, price, unitsellingprice, quantity, (quantity * unitsellingprice) AS
ordervalue, name as shopname
FROM bookstore.orderline, bookstore.shoporder, bookstore.shop, bookstore.book
WHERE orderline.shoporderid = shoporder.shoporderid AND shoporder.shopid = shop.shopid
AND orderline.bookid = book.bookid AND book.bookid = 95
ORDER BY orderdate ASC;
```



```
SELECT orderdate, title, price, unitsellingprice, quantity, (quantity * unitsellingprice) AS
ordervalue, name as shopname
FROM bookstore.orderline, bookstore.shoporder, bookstore.shop, bookstore.book
WHERE orderline.shoporderid = shoporder.shoporderid AND shoporder.shopid = shop.shopid
AND orderline.bookid = book.bookid AND book.bookid = "aadsa"
ORDER BY orderdate ASC;
```



Summary Line Query

```
SELECT orderline.bookid, title, COUNT(orderline.bookid) AS numberoforders, SUM((quantity *
unitsellingprice)) AS totalsellingvalue
FROM bookstore.orderline, bookstore.book
WHERE orderline.bookid = 10 AND orderline.bookid = book.bookid
GROUP BY orderline.bookid, title;
```

Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT orderline.bookid, title, COUNT(orderline.bookid) AS numberoforders, SUM((quantity * unitsellingprice)) AS totalsellingvalue
FROM bookstore.orderline, bookstore.book
WHERE orderline.bookid = 10 AND orderline.bookid = book.bookid
GROUP BY orderline.bookid, title;
```

Output pane

Data Output Explain Messages History

	bookid integer	title character varying(50)	numberoforders bigint	totalsellingvalue numeric
1	10	Mental Arithmetic	5	302.50

```
SELECT orderline.bookid, title, COUNT(orderline.bookid) AS numberoforders, SUM((quantity *
unitsellingprice)) AS totalsellingvalue
FROM bookstore.orderline, bookstore.book
WHERE orderline.bookid = 95 AND orderline.bookid = book.bookid
GROUP BY orderline.bookid, title;
```

Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

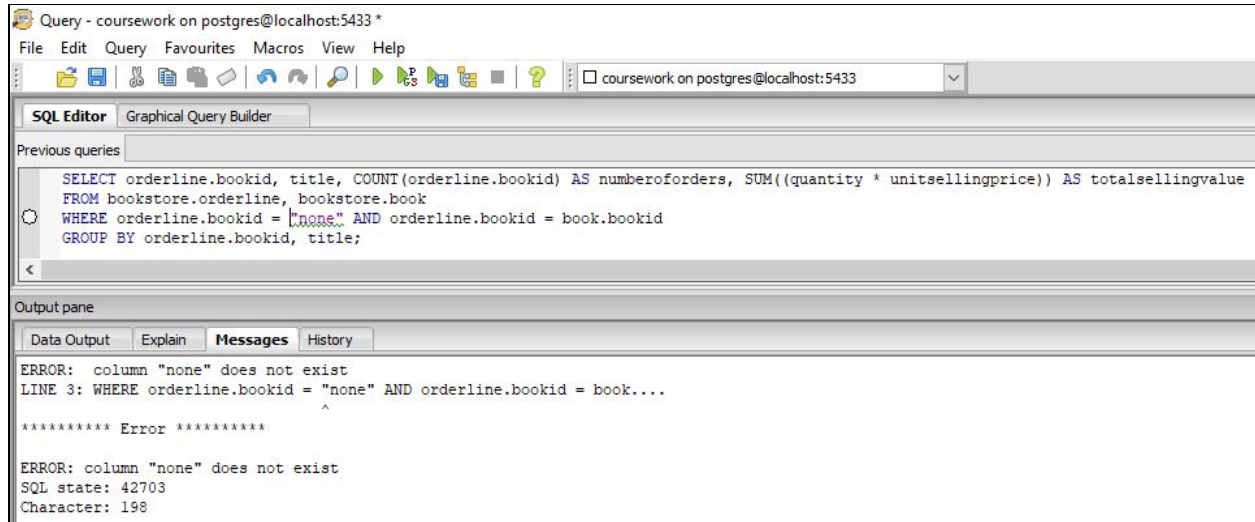
```
SELECT orderline.bookid, title, COUNT(orderline.bookid) AS numberoforders, SUM((quantity * unitsellingprice)) AS totalsellingvalue
FROM bookstore.orderline, bookstore.book
WHERE orderline.bookid = 95 AND orderline.bookid = book.bookid
GROUP BY orderline.bookid, title;
```

Output pane

Data Output Explain Messages History

	bookid integer	title character varying(50)	numberoforders bigint	totalsellingvalue numeric
--	-------------------	--------------------------------	--------------------------	------------------------------

```
SELECT orderline.bookid, title, COUNT(orderline.bookid) AS numberoforders, SUM((quantity *
unitsellingprice)) AS totalsellingvalue
FROM bookstore.orderline, bookstore.book
WHERE orderline.bookid = "none" AND orderline.bookid = book.bookid
GROUP BY orderline.bookid, title;
```



The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main text area contains the following SQL query:

```
SELECT orderline.bookid, title, COUNT(orderline.bookid) AS numeroorders, SUM((quantity * unitsellingprice)) AS totalsellingvalue
FROM bookstore.orderline, bookstore.book
WHERE orderline.bookid = "none" AND orderline.bookid = book.bookid
GROUP BY orderline.bookid, title;
```

Below the query editor is an "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Messages" tab is selected, displaying the following error message:

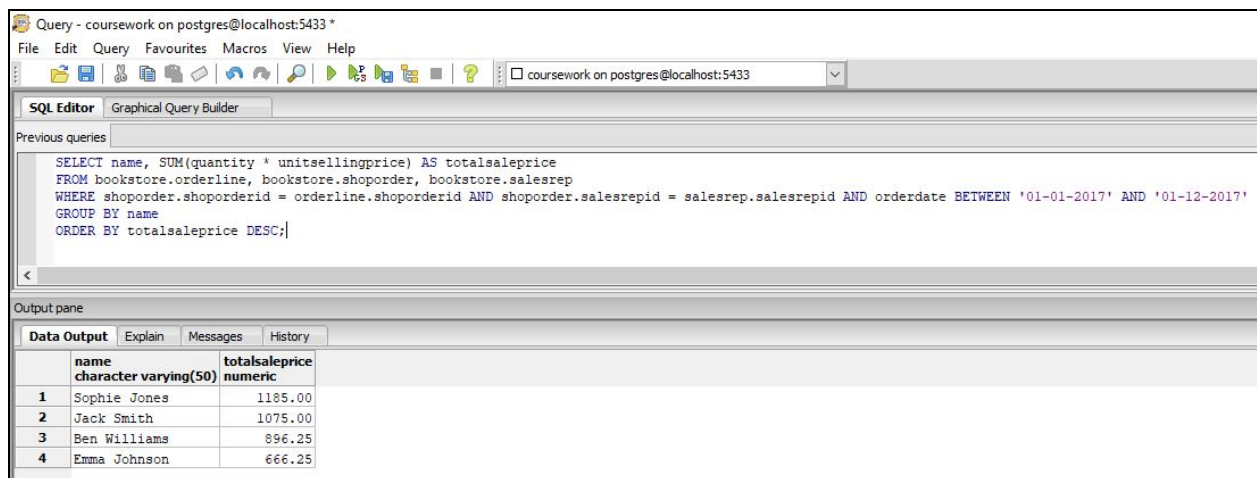
```
ERROR: column "none" does not exist
LINE 3: WHERE orderline.bookid = "none" AND orderline.bookid = book....
                                ^
***** Error *****

ERROR: column "none" does not exist
SQL state: 42703
Character: 198
```

Task 6

“Given start and end dates, produce a report showing the performance of each sales representative over that period. The report should begin with the rep who generated most orders by value and include total units sold and total order value. It should include all sales reps.”

```
“SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid =
salesrep.salesrepid AND orderdate BETWEEN '01-01-2017' AND '01-12-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;”
```



The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The SQL Editor tab is active, displaying the following query:

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid = salesrep.salesrepid AND orderdate BETWEEN '01-01-2017' AND '01-12-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

The Output pane at the bottom shows the results of the query in a table with the following data:

	name character varying(50)	totalsaleprice numeric
1	Sophie Jones	1185.00
2	Jack Smith	1075.00
3	Ben Williams	896.25
4	Emma Johnson	666.25

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid =
salesrep.salesrepid AND orderdate BETWEEN '00-00-2017' AND '01-12-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```


Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid = salesrep.salesrepid AND orderdate BETWEEN '00-00-2017' AND '01-12-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

Output pane

Data Output Explain Messages History

```
ERROR: date/time field value out of range: "00-00-2017"
LINE 3: ...repid = salesrep.salesrepid AND orderdate BETWEEN '00-00-2017...
                                ^
HINT: Perhaps you need a different "datestyle" setting.
***** Error *****

ERROR: date/time field value out of range: "00-00-2017"
SQL state: 22008
Hint: Perhaps you need a different "datestyle" setting.
Character: 254
```

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid =
salesrep.salesrepid AND orderdate BETWEEN '12-12-2017' AND '01-12-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid = salesrep.salesrepid AND orderdate BETWEEN '12-12-2017' AND '01-12-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

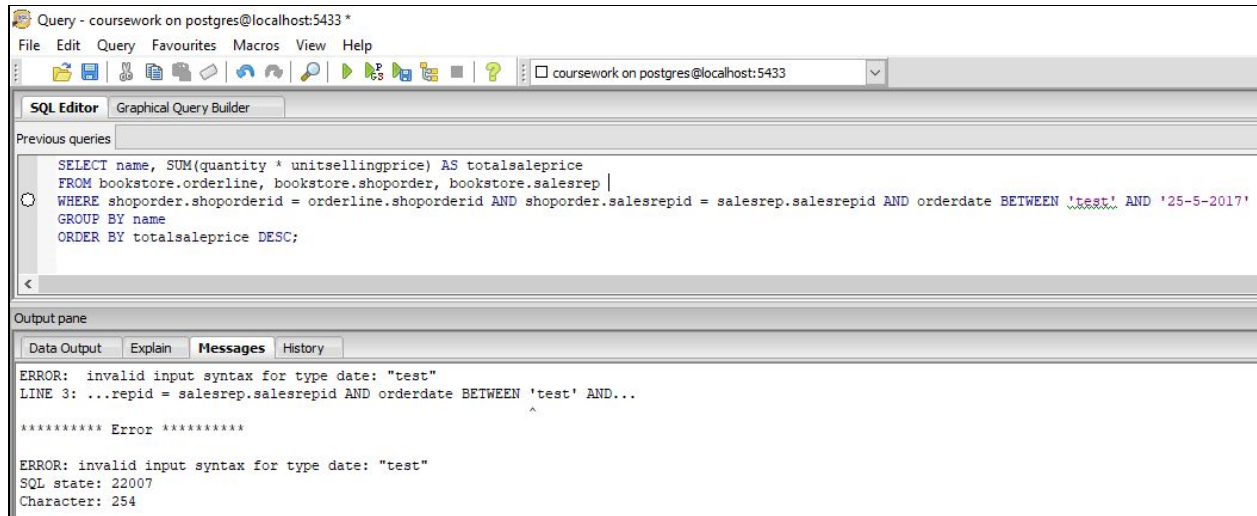
Output pane

Data Output Explain Messages History

name	totalsaleprice
character varying(50)	numeric

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid =
salesrep.salesrepid AND orderdate BETWEEN 'test' AND '25-5-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
```



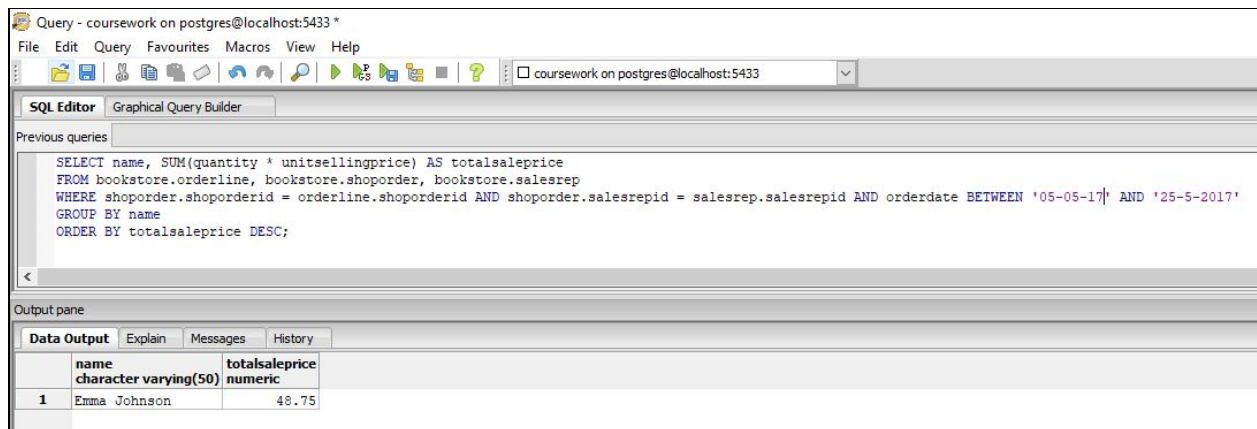
The screenshot shows a PostgreSQL SQL Editor window titled "Query - coursework on postgres@localhost:5433 *". The query is as follows:

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep |
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid = salesrep.salesrepid AND orderdate BETWEEN 'test' AND '25-5-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

The "Output pane" shows the following error message:

```
ERROR: invalid input syntax for type date: "test"
LINE 3: ...repid = salesrep.salesrepid AND orderdate BETWEEN 'test' AND...
                        ^
***** Error *****
ERROR: invalid input syntax for type date: "test"
SQL state: 22007
Character: 254
```

*FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid =
salesrep.salesrepid AND orderdate BETWEEN '05-05-17' AND '25-5-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;*



The screenshot shows the same PostgreSQL SQL Editor window, but the query has been corrected to use the date '05-05-17' instead of 'test'. The query is as follows:

```
SELECT name, SUM(quantity * unitsellingprice) AS totalsaleprice
FROM bookstore.orderline, bookstore.shoporder, bookstore.salesrep
WHERE shoporder.shoporderid = orderline.shoporderid AND shoporder.salesrepid = salesrep.salesrepid AND orderdate BETWEEN '05-05-17' AND '25-5-2017'
GROUP BY name
ORDER BY totalsaleprice DESC;
```

The "Output pane" shows the following result set:

	name character varying(50)	totalsaleprice numeric
1	Emma Johnson	48.75

Task 7

“Given a category ID and discount percentage, apply a discount to the standard price of all books in that category”

Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

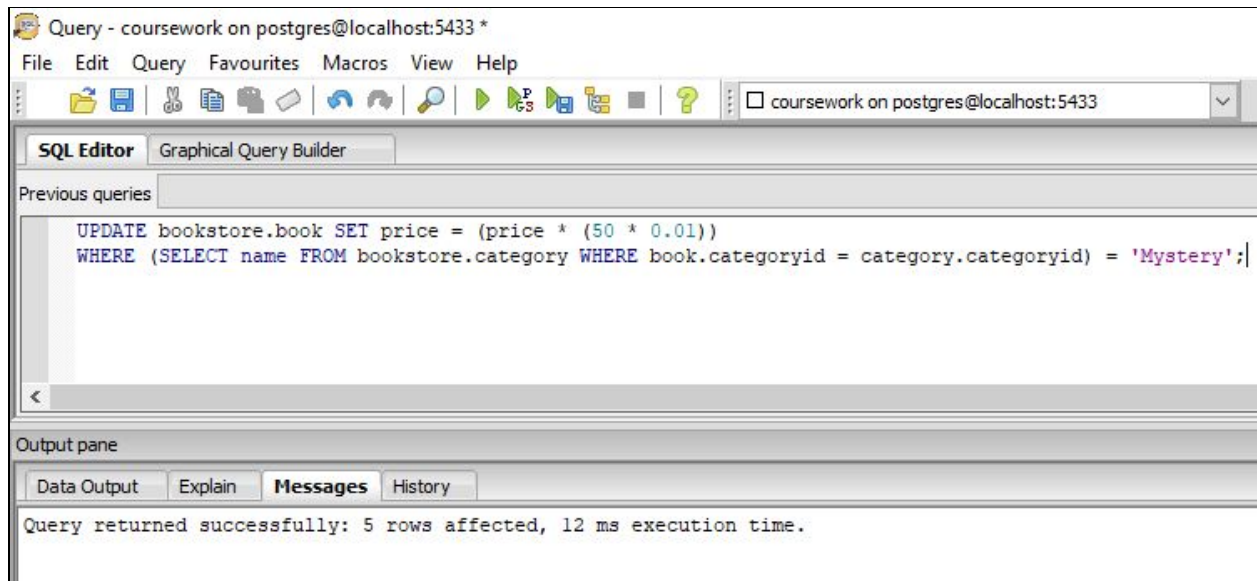
```
SELECT * FROM bookstore.book
```

Output pane

Data Output Explain Messages History

	bookid integer	title character varying(50)	price numeric(10,2)	categoryid integer	publisherid integer
1	0	Children of Time	6.99	0	3
2	2	Functional Skills 4	14.99	5	4
3	3	The Silent Wife	4.99	1	1
4	4	The Selfish Gene	4.45	6	2
5	6	KS3 Complete Study Guide	6.95	5	3
6	8	Colony One Mars	8.99	0	1
7	9	Encyclopedia of Earth: a Visual Guide	20.95	8	0
8	10	Mental Arithmetic 5	3.95	5	4
9	11	On the Origin of Species	10.95	6	3
10	12	Pride and Prejudice	3.95	3	2
11	14	The Art of War	7.45	9	2
12	15	Basic English Dictionary	3.45	7	1
13	16	The GCHQ Puzzle Book	5.96	5	4
14	17	Macbeth	2.99	1	0
15	19	Rise and Fall	8.45	9	3
16	1	Nevernight	4.00	4	0
17	5	The Maltese Falcon	7.00	2	4
18	7	Gone Girl	10.96	2	2
19	13	"A" is for Alibi	4.96	2	1
20	18	A Clash of Kings	11.00	4	3

```
UPDATE bookstore.book SET price = (price * (50 * 0.01))  
WHERE (SELECT name FROM bookstore.category WHERE book.categoryid =  
category.categoryid) = 'Mystery';
```



Query - coursework on postgres@localhost:5433 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT * FROM bookstore.book;
```

Output pane

Data Output Explain Messages History

	bookid integer	title character varying(50)	price numeric(10,2)	categoryid integer	publisherid integer
1	0	Children of Time	6.99	0	3
2	2	Functional Skills 4	14.99	5	4
3	3	The Silent Wife	4.99	1	1
4	4	The Selfish Gene	4.45	6	2
5	6	KS3 Complete Study Guide	6.95	5	3
6	8	Colony One Mars	8.99	0	1
7	9	Encyclopedia of Earth: a Visual Guide	20.95	8	0
8	10	Mental Arithmetic 5	3.95	5	4
9	11	On the Origin of Species	10.95	6	3
10	12	Pride and Prejudice	3.95	3	2
11	14	The Art of War	7.45	9	2
12	15	Basic English Dictionary	3.45	7	1
13	16	The GCHQ Puzzle Book	5.96	5	4
14	17	Macbeth	2.99	1	0
15	19	Rise and Fall	8.45	9	3
16	1	Nevernight	<u>2.00</u>	4	0
17	5	The Maltese Falcon	<u>3.50</u>	2	4
18	7	Gone Girl	<u>5.48</u>	2	2
19	13	"A" is for Alibi	<u>2.48</u>	2	1
20	18	A Clash of Kings	5.50	4	3

UPDATE bookstore.book SET price = (price * (200 * 0.01))
 WHERE (SELECT name FROM bookstore.category WHERE book.categoryid =
 category.categoryid) = 'test';

