

2023 Spring CS504 Project

Keerthana Cheruvu (G01160202)

GMU CS504

Objective:

Designing and Implementation of the DBMS for the given public library database which will help the staff for the easy access of the materials.

The design is explained with the help of the ER model and implementation of the database is performed using the PostgreSQL where manipulated the data using different queries.

Database Design:

The public library database is organized into Material, Catalog, Genre, Borrow, Author, Authorship, Member, Staff.

The Catalog of the library explains about the type of book such as books, magazines, e-books etc., available in which exact location.

The Genre of the library stores the genre type and the description of the genre explaining the type of the genre.

The Material management stores the information such as books, magazines, e-books, and audiobooks, including their titles, authors, publication dates, and genres.

The Borrow represents borrow, return and due dates of the materials. If a material is borrowed then the staff should update the material details such as borrow, return and due dates along with the member, material details.

The Member management manages the details related to members of the library, their contact information, library members who can borrow and reserve materials.

The Staff stores information about the library staff, their contact information and their hiring date.

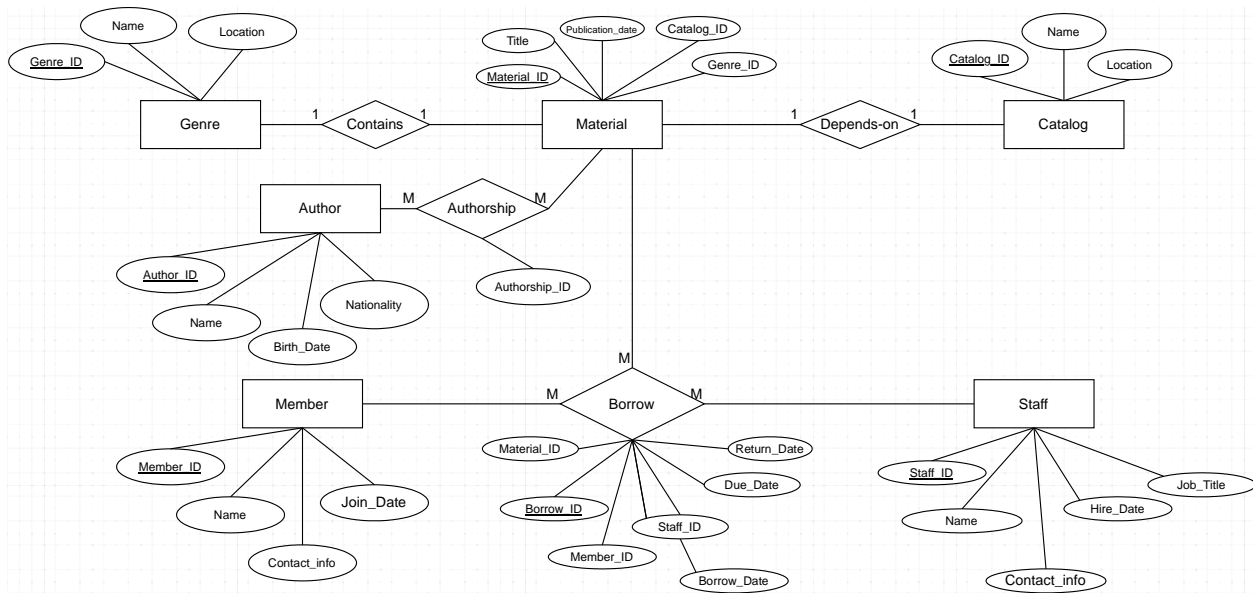
The Author entity gives the details of the authors of the materials such as their name and nationality. The Authorship gives the relation between the author and the material entities.

The database design is explained with the help of the ER model which is shown in the following image:

The material entity contains the genre_id and the catalog_id. The relation between material and genre entities is 1:1 where each material has a genre_id. The relation between material and catalog is 1:1 where each material has a catalog_id.

The relation between material and member entities is M:M where borrow acts as the relation between them. The relation between material and staff entities is M:M where borrow acts as the relation between them.

The relation between material and author entities is M:M where authorship acts as the relation between them.



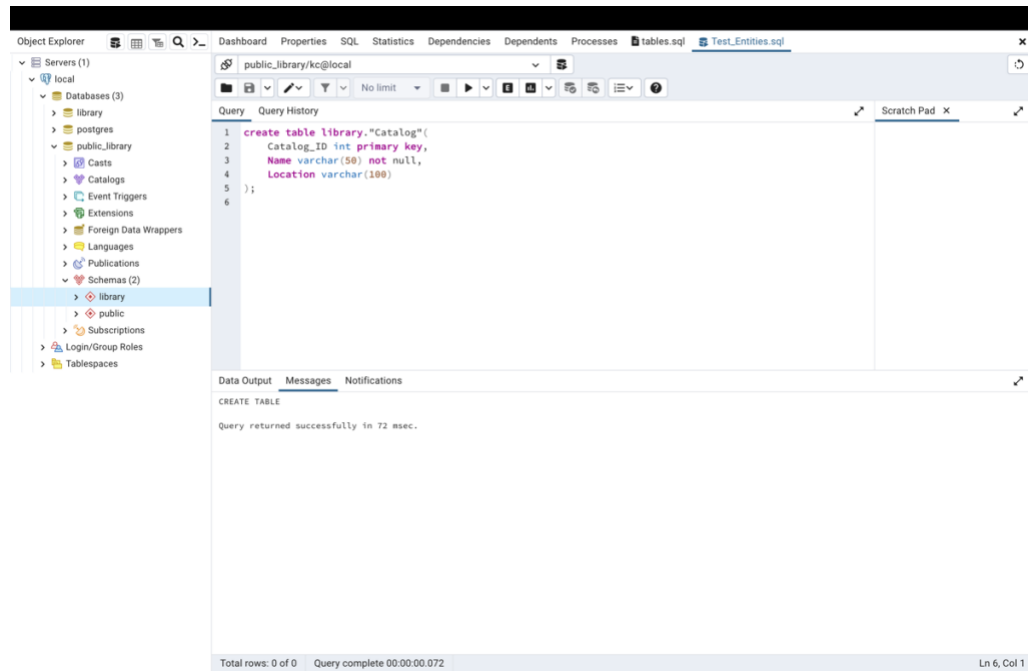
Database Implementation:

- The Database Management System (DBMS) used is Postgre with the help of pgAdmin.
- Created a database named public library under which created a schema named library.
- Imported the given sample data into the library schema using DDL commands and performed the manipulation of the database such as update, delete, search etc.,

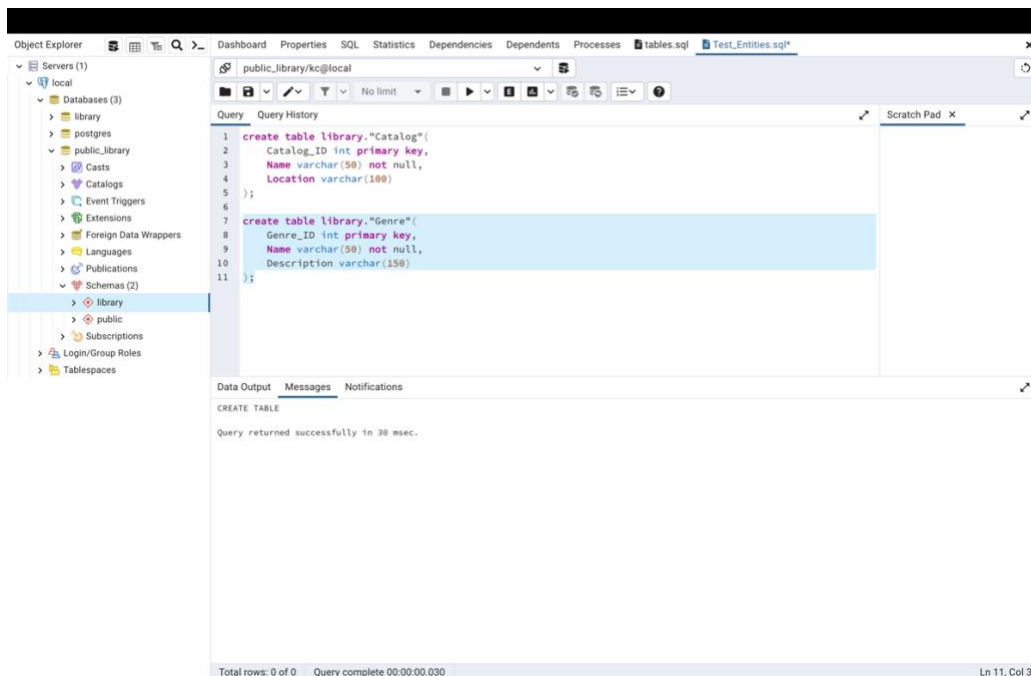
Creating Entities:

In the query library represents schema.

The following image shows the SQL query creation of the catalog entity.



The following image shows the SQL query creation of the Genre entity.



The following image shows the SQL query creation of the Material entity.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the 'public_library' database selected. The central pane displays the SQL query editor with the following code:

```
4 Location varchar(100)
5 };
6
7 create table library."Genre"(
8 Genre_ID int primary key,
9 Name varchar(50) not null,
10 Description varchar(150)
11 );
12
13 create table library."Material"(
14 Material_ID int primary key,
15 Title varchar(100) not null,
16 Publication_Date varchar(15),
17 Catalog_ID int,
18 Genre_ID int,
19 constraint fk_catalog
20 foreign key(catalog_ID)
21 references library."Catalog"(Catalog_ID),
22 constraint fk_genre
23 foreign key(Genre_ID)
24 references library."Genre"(Genre_ID)
25 );
```

The Data Output pane at the bottom shows the message: "Query returned successfully in 42 msec."

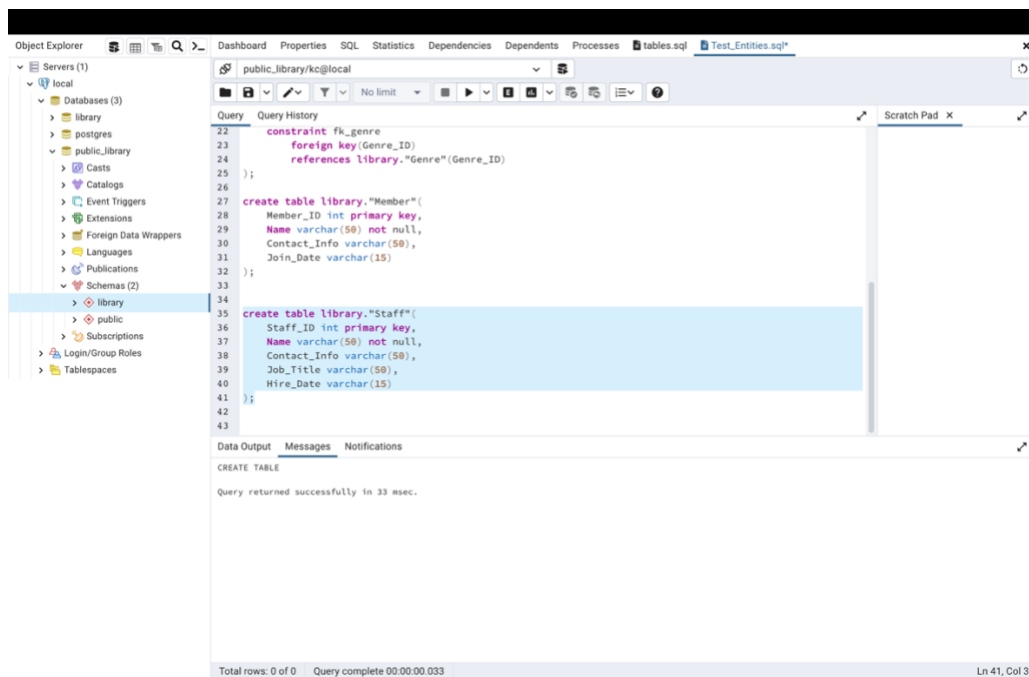
The following image shows the SQL query creation of the Member entity.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the 'public_library' database selected. The central pane displays the SQL query editor with the following code:

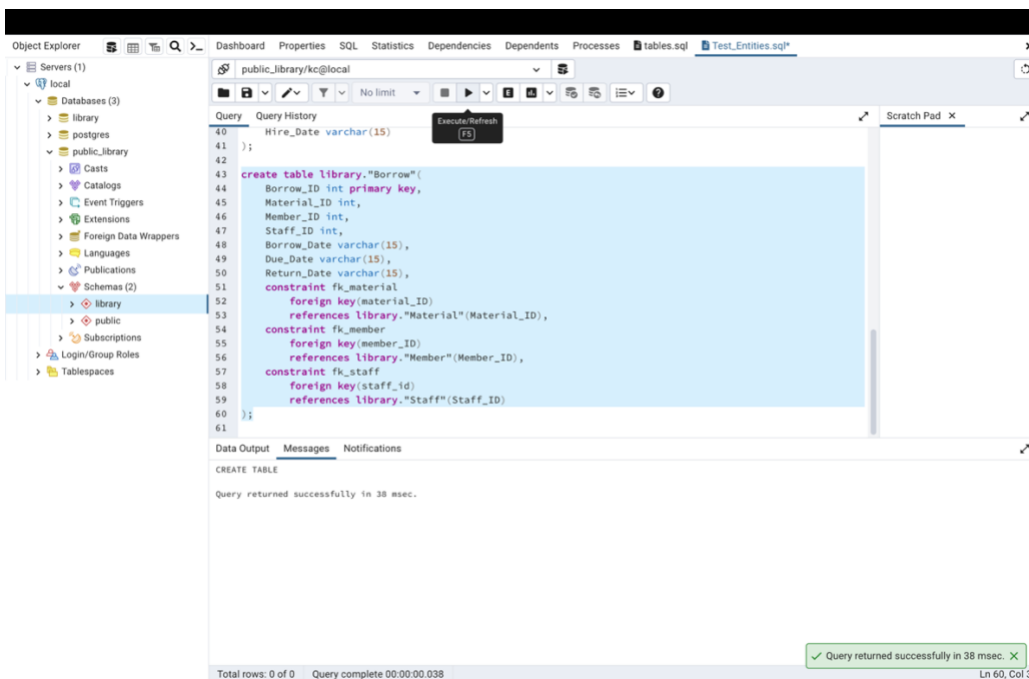
```
14 Material_ID int primary key,
15 Title varchar(100) not null,
16 Publication_Date varchar(15),
17 Catalog_ID int,
18 Genre_ID int,
19 constraint fk_catalog
20 foreign key(catalog_ID)
21 references library."Catalog"(Catalog_ID),
22 constraint fk_genre
23 foreign key(Genre_ID)
24 references library."Genre"(Genre_ID)
25 );
26
27 create table library."Member"(
28 Member_ID int primary key,
29 Name varchar(50) not null,
30 Contact_Info varchar(50),
31 Join_Date varchar(15)
32 );
33
34
35
```

The Data Output pane at the bottom shows the message: "Query returned successfully in 68 msec."

The following image shows the SQL query creation of the Staff entity.



The following image shows the SQL query creation of the Borrow entity.



The following image shows the SQL query creation of the Author entity.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure, with the 'library' database selected. The central pane shows a SQL query window with the following code:

```
48 Borrow_Date varchar(15),
49 Due_Date varchar(15),
50 Return_Date varchar(15),
51 constraint fk_material
52 foreign key(material_ID)
53 references library."Material"(Material_ID),
54 constraint fk_member
55 foreign key(member_ID)
56 references library."Member"(Member_ID),
57 constraint fk_staff
58 foreign key(staff_id)
59 references library."Staff"(Staff_ID)
60 );
61
62 create table library."Author"(
63 Author_ID int primary key,
64 Name varchar(50) not null,
65 Birth_Date varchar(15),
66 Nationality varchar(25)
67 );
68
69
```

The Data Output pane at the bottom shows the message: "CREATE TABLE Query returned successfully in 37 msec." A status bar at the bottom indicates "Total rows: 0 of 0 Query complete 00:00:00.037 Ln 67, Col 3".

The following image shows the SQL query creation of the Authorship entity.

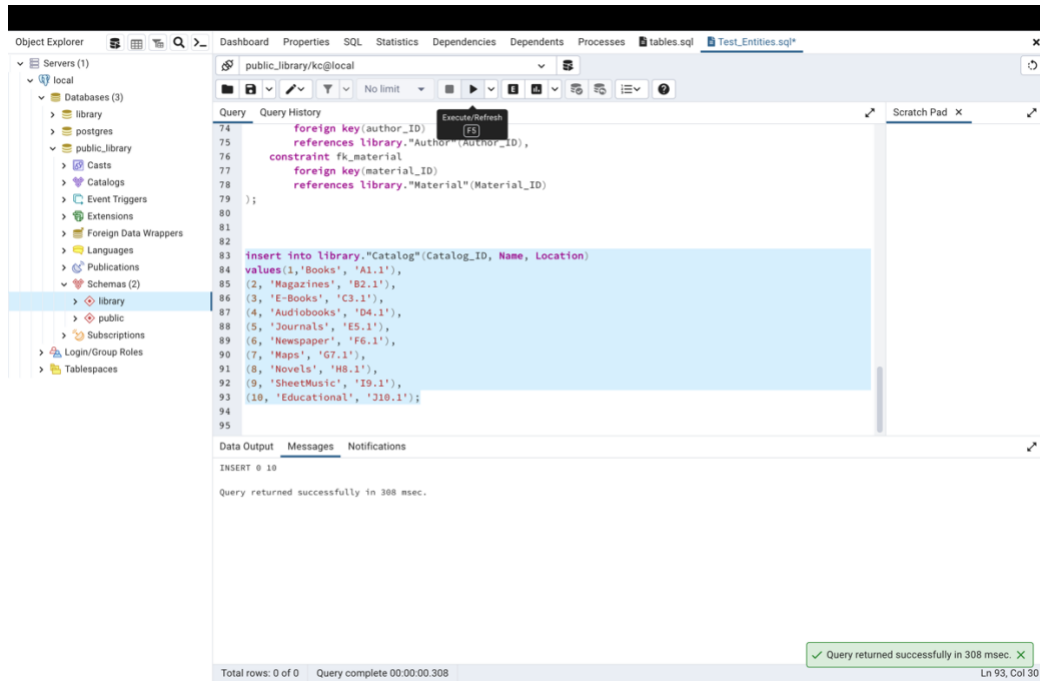
The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure, with the 'library' database selected. The central pane shows a SQL query window with the following code:

```
59 references library."Staff"(Staff_ID)
60 );
61
62 create table library."Author"(
63 Author_ID int primary key,
64 Name varchar(50) not null,
65 Birth_Date varchar(15),
66 Nationality varchar(25)
67 );
68
69 create table library."Authorship"(
70 Authorship_ID int primary key,
71 Author_ID int,
72 Material_ID int,
73 constraint fk_author
74 foreign key(author_ID)
75 references library."Author"(Author_ID),
76 constraint fk_material
77 foreign key(material_ID)
78 references library."Material"(Material_ID)
79 );
80
```

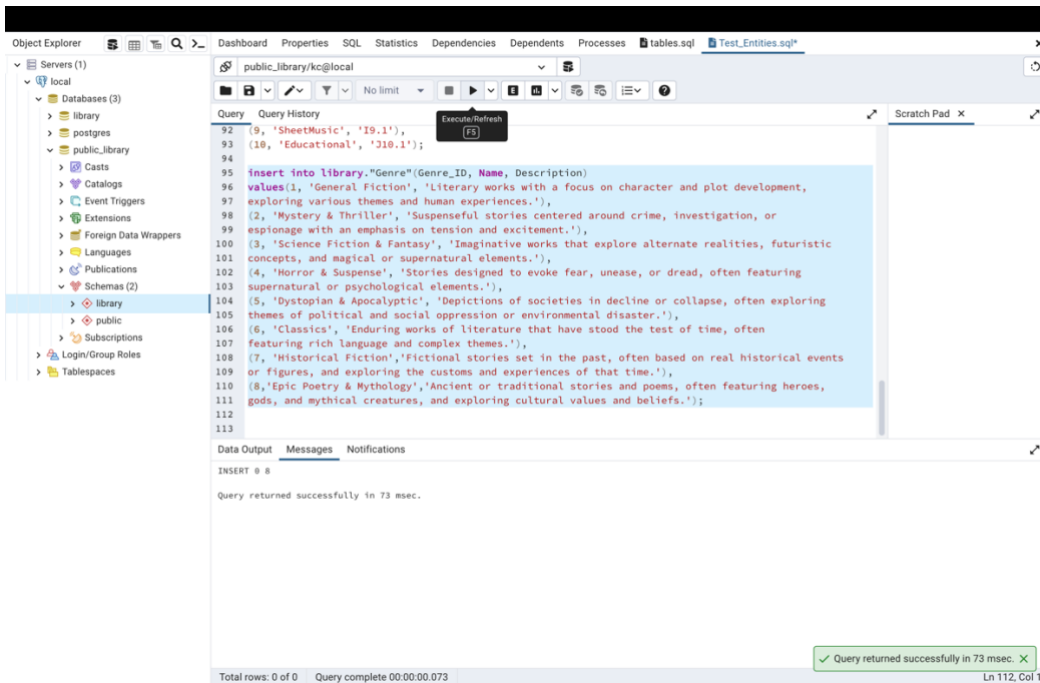
The Data Output pane at the bottom shows the message: "CREATE TABLE Query returned successfully in 34 msec." A status bar at the bottom indicates "Total rows: 0 of 0 Query complete 00:00:00.034 Ln 79, Col 3".

Inserting Values:

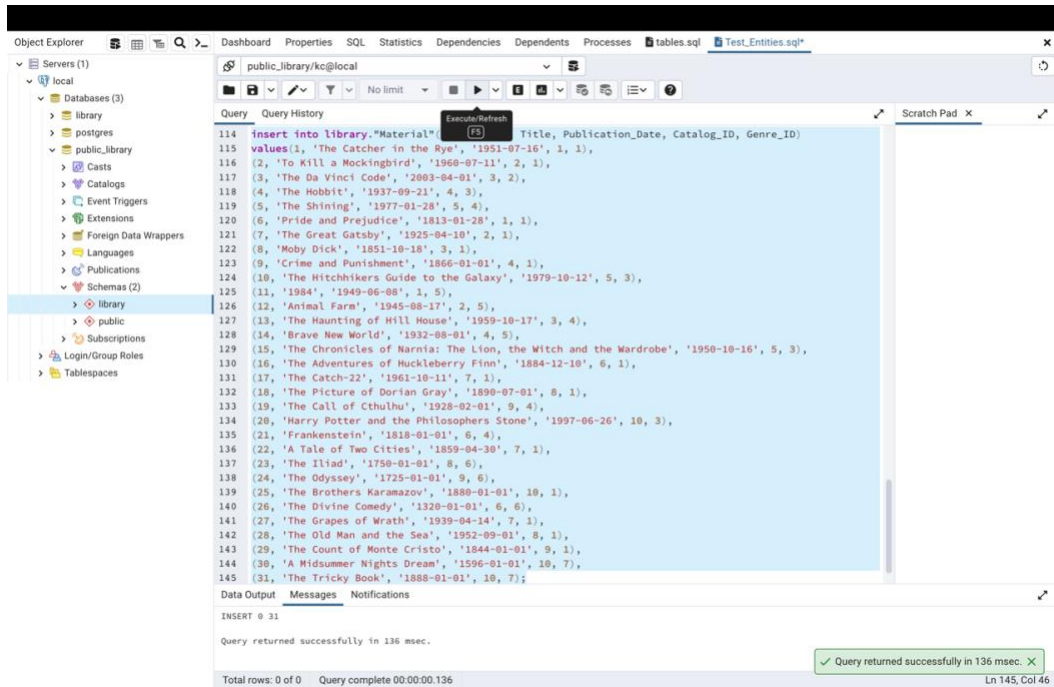
The following image shows the SQL query of inserting values for the Catalog Entity.



The following image shows the SQL query of inserting values for the Genre Entity.



The following image shows the SQL query of inserting values for the Material Entity.

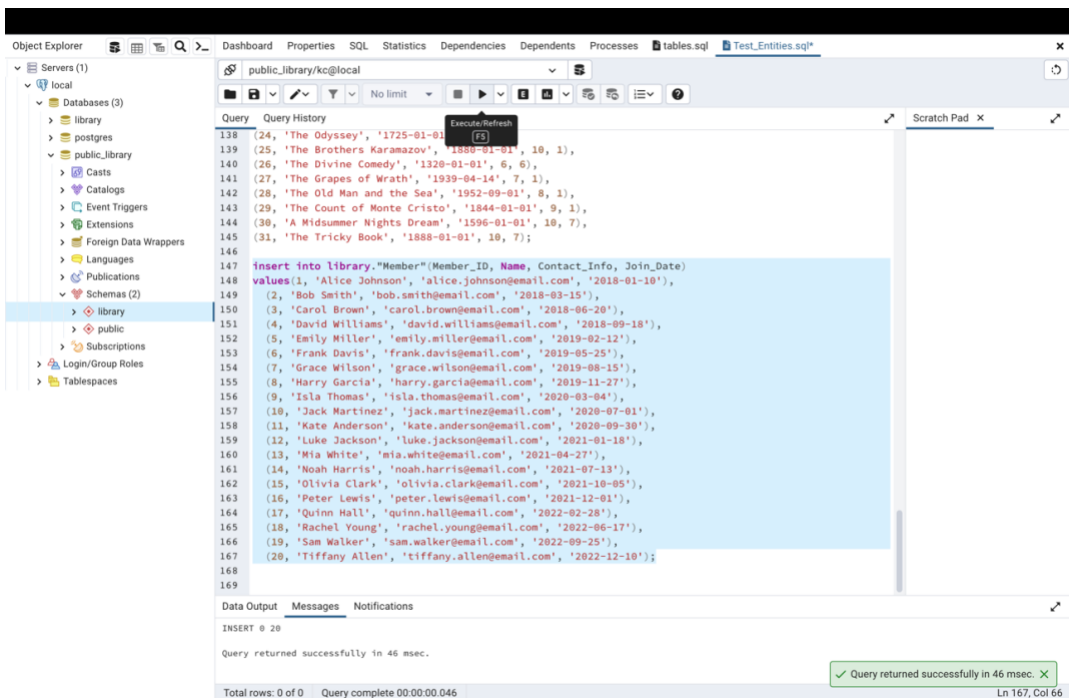


The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'public_library' database selected. The right pane shows the 'Query' window with the following SQL query:

```
114 insert into library."Material" (Title, Publication_Date, Catalog_ID, Genre_ID)
115 values(1, 'The Catcher in the Rye', '1951-07-16', 1, 1),
116 (2, 'To Kill a Mockingbird', '1960-07-11', 2, 1),
117 (3, 'The Da Vinci Code', '2003-04-01', 3, 2),
118 (4, 'The Hobbit', '1937-09-21', 4, 3),
119 (5, 'The Shining', '1977-01-28', 5, 4),
120 (6, 'Pride and Prejudice', '1813-01-28', 1, 1),
121 (7, 'The Great Gatsby', '1925-04-10', 2, 1),
122 (8, 'Moby Dick', '1851-10-18', 3, 1),
123 (9, 'Crime and Punishment', '1866-01-01', 4, 1),
124 (10, 'The Hitchhikers Guide to the Galaxy', '1979-10-12', 5, 3),
125 (11, '1984', '1949-06-08', 1, 5),
126 (12, 'Animal Farm', '1945-08-17', 2, 5),
127 (13, 'The Haunting of Hill House', '1959-10-17', 3, 4),
128 (14, 'Brave New World', '1932-08-01', 4, 5),
129 (15, 'The Chronicles of Narnia: The Lion, the Witch and the Wardrobe', '1950-10-16', 5, 3),
130 (16, 'The Adventures of Huckleberry Finn', '1884-12-10', 6, 1),
131 (17, 'The Catch-22', '1961-10-11', 7, 1),
132 (18, 'The Picture of Dorian Gray', '1890-07-01', 8, 1),
133 (19, 'The Call of Cthulhu', '1928-02-01', 9, 4),
134 (20, 'Harry Potter and the Philosophers Stone', '1997-06-26', 10, 3),
135 (21, 'Frankenstein', '1818-01-01', 6, 4),
136 (22, 'A Tale of Two Cities', '1859-04-30', 7, 1),
137 (23, 'The Iliad', '1750-01-01', 8, 6),
138 (24, 'The Odyssey', '1725-01-01', 9, 6),
139 (25, 'The Brothers Karamazov', '1880-01-01', 10, 1),
140 (26, 'The Divine Comedy', '1320-01-01', 6, 6),
141 (27, 'The Grapes of Wrath', '1939-04-14', 7, 1),
142 (28, 'The Old Man and the Sea', '1952-09-01', 8, 1),
143 (29, 'The Count of Monte Cristo', '1844-01-01', 9, 1),
144 (30, 'A Midsummer Nights Dream', '1596-01-01', 10, 7),
145 (31, 'The Tricky Book', '1888-01-01', 10, 7);
```

The query is executed successfully, returning 136 rows in 136 msec. The status bar at the bottom indicates 'Query complete 00:00:00.136'.

The following image shows the SQL query of inserting values for the Member Entity.

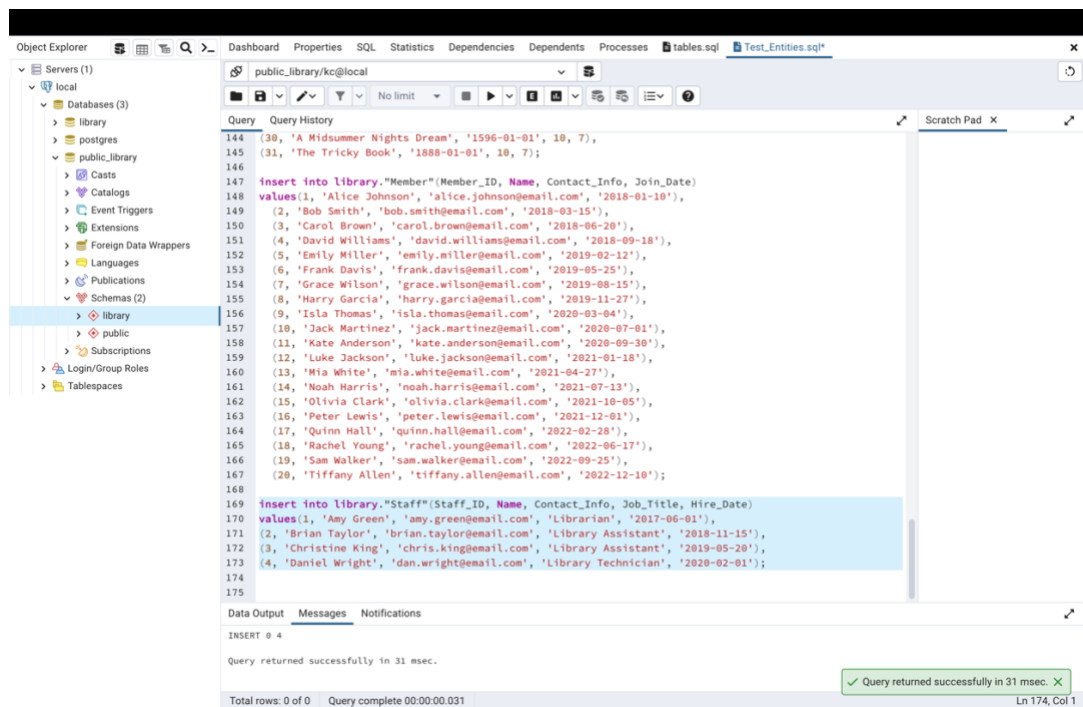


The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'public_library' database selected. The right pane shows the 'Query' window with the following SQL query:

```
138 (24, 'The Odyssey', '1725-01-01', 9, 6),
139 (25, 'The Brothers Karamazov', '1880-01-01', 10, 1),
140 (26, 'The Divine Comedy', '1320-01-01', 6, 6),
141 (27, 'The Grapes of Wrath', '1939-04-14', 7, 1),
142 (28, 'The Old Man and the Sea', '1952-09-01', 8, 1),
143 (29, 'The Count of Monte Cristo', '1844-01-01', 9, 1),
144 (30, 'A Midsummer Nights Dream', '1596-01-01', 10, 7),
145 (31, 'The Tricky Book', '1888-01-01', 10, 7);
146
147 insert into library."Member" (Member_ID, Name, Contact_Info, Join_Date)
148 values(1, 'Alice Johnson', 'alice.johnson@email.com', '2018-01-10'),
149 (2, 'Bob Smith', 'bob.smith@email.com', '2018-03-15'),
150 (3, 'Carol Brown', 'carol.brown@email.com', '2018-06-20'),
151 (4, 'David Williams', 'david.williams@email.com', '2018-09-18'),
152 (5, 'Emily Miller', 'emily.miller@email.com', '2019-02-12'),
153 (6, 'Frank Davis', 'frank.davis@email.com', '2019-05-25'),
154 (7, 'Grace Wilson', 'grace.wilson@email.com', '2019-08-15'),
155 (8, 'Harry Garcia', 'harry.garcia@email.com', '2019-11-27'),
156 (9, 'Isla Thomas', 'isla.thomas@email.com', '2020-03-04'),
157 (10, 'Jack Martinez', 'jack.martinez@email.com', '2020-07-01'),
158 (11, 'Kate Anderson', 'kate.anderson@email.com', '2020-09-30'),
159 (12, 'Luke Jackson', 'luke.jackson@email.com', '2021-01-18'),
160 (13, 'Mia White', 'mia.white@email.com', '2021-04-27'),
161 (14, 'Noah Harris', 'noah.harris@email.com', '2021-07-13'),
162 (15, 'Olivia Clark', 'olivia.clark@email.com', '2021-10-05'),
163 (16, 'Peter Lewis', 'peter.lewis@email.com', '2021-12-01'),
164 (17, 'Quinn Hall', 'quinn.hall@email.com', '2022-02-28'),
165 (18, 'Rachel Young', 'rachel.young@email.com', '2022-06-17'),
166 (19, 'Sam Walker', 'sam.walker@email.com', '2022-09-25'),
167 (20, 'Tiffany Allen', 'tiffany.allen@email.com', '2022-12-10');
168
169
```

The query is executed successfully, returning 46 rows in 46 msec. The status bar at the bottom indicates 'Query complete 00:00:00.046'.

The following image shows the SQL query of inserting values for the Staff Entity.

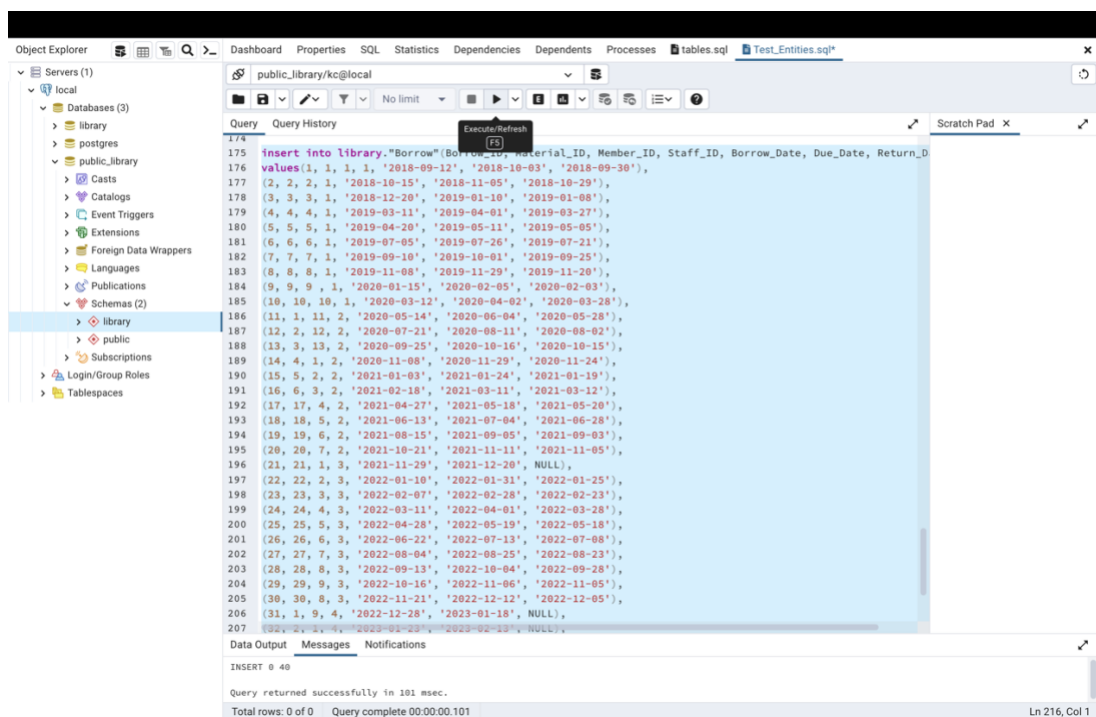


The screenshot shows a database management tool interface. On the left, the 'Object Explorer' pane displays a tree view of the database structure, including 'Servers (1)', 'local', 'Databases (3)', 'library', 'postgres', 'public_library', 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas (2)', 'library', 'public', 'Subscriptions', 'Login/Group Roles', and 'Tablespaces'. The 'library' database is selected. The main pane shows the 'Query' tab with the following SQL query:

```
144 (30, 'A Midsummer Nights Dream', '1596-01-01', 10, 7),
145 (31, 'The Tricky Book', '1888-01-01', 10, 7);
146
147 insert into library."Member"(Member_ID, Name, Contact_Info, Join_Date)
148 values(1, 'Alice Johnson', 'alice.johnson@email.com', '2018-01-10'),
149 (2, 'Bob Smith', 'bob.smith@email.com', '2018-03-15'),
150 (3, 'Carol Brown', 'carol.brown@email.com', '2018-06-20'),
151 (4, 'David Williams', 'david.williams@email.com', '2018-09-18'),
152 (5, 'Emily Miller', 'emily.miller@email.com', '2019-02-12'),
153 (6, 'Frank Davis', 'frank.davis@email.com', '2019-05-25'),
154 (7, 'Grace Wilson', 'grace.wilson@email.com', '2019-08-15'),
155 (8, 'Harry Garcia', 'harry.garcia@email.com', '2019-11-27'),
156 (9, 'Isla Thomas', 'isla.thomas@email.com', '2020-03-04'),
157 (10, 'Jack Martinez', 'jack.martinez@email.com', '2020-07-01'),
158 (11, 'Kate Anderson', 'kate.anderson@email.com', '2020-09-30'),
159 (12, 'Luke Jackson', 'luke.jackson@email.com', '2021-01-18'),
160 (13, 'Mia White', 'mia.white@email.com', '2021-04-27'),
161 (14, 'Noah Harris', 'noah.harris@email.com', '2021-07-13'),
162 (15, 'Olivia Clark', 'olivia.clark@email.com', '2021-10-05'),
163 (16, 'Peter Lewis', 'peter.lewis@email.com', '2021-12-01'),
164 (17, 'Quinn Hall', 'quinn.hall@email.com', '2022-02-28'),
165 (18, 'Rachel Young', 'rachel.young@email.com', '2022-06-17'),
166 (19, 'Sam Walker', 'sam.walker@email.com', '2022-09-25'),
167 (20, 'Tiffany Allen', 'tiffany.allen@email.com', '2022-12-10');
168
169 insert into library."Staff"(Staff_ID, Name, Contact_Info, Job_Title, Hire_Date)
170 values(1, 'Amy Green', 'amy.green@email.com', 'Librarian', '2017-06-01'),
171 (2, 'Brian Taylor', 'brian.taylor@email.com', 'Library Assistant', '2018-11-15'),
172 (3, 'Christine King', 'christine.king@email.com', 'Library Assistant', '2019-05-20'),
173 (4, 'Daniel Wright', 'dan.wright@email.com', 'Library Technician', '2020-02-01');
174
175
```

The 'Data Output' tab shows the result of the query: 'INSERT 0 4'. A message at the bottom states: 'Query returned successfully in 31 msec.' A green checkmark icon and the text 'Query returned successfully in 31 msec.' are visible in the bottom right corner. The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.031'.

The following image shows the SQL query of inserting values for the Borrow Entity.

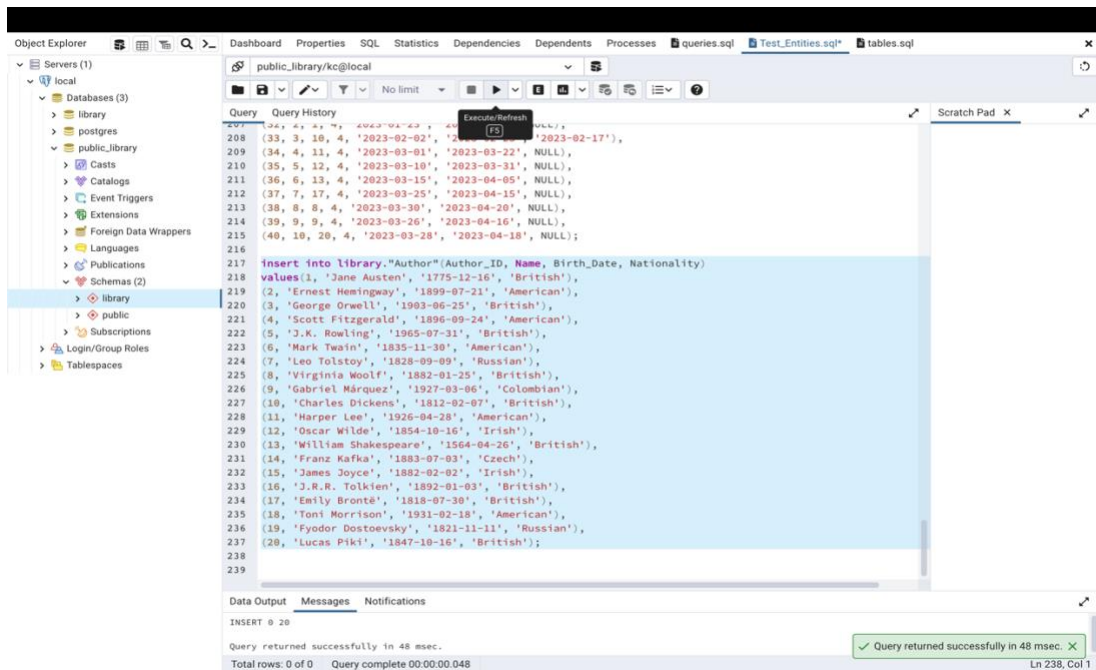


The screenshot shows the same database management tool interface. The 'Query' tab displays the following SQL query:

```
174
175
176 insert into library."Borrow"(Borrow_ID, Material_ID, Member_ID, Staff_ID, Borrow_Date, Due_Date, Return_D
177 values(1, 1, 1, 1, '2018-09-12', '2018-10-03', '2018-09-30'),
178 (2, 2, 2, 1, '2018-10-15', '2018-11-05', '2018-10-29'),
179 (3, 3, 3, 1, '2018-12-20', '2019-01-10', '2019-01-08'),
180 (4, 4, 4, 1, '2019-03-11', '2019-04-01', '2019-03-27'),
181 (5, 5, 5, 1, '2019-04-20', '2019-05-11', '2019-05-05'),
182 (6, 6, 6, 1, '2019-07-05', '2019-07-26', '2019-07-21'),
183 (7, 7, 7, 1, '2019-09-10', '2019-10-01', '2019-09-25'),
184 (8, 8, 8, 1, '2019-11-08', '2019-11-29', '2019-11-20'),
185 (9, 9, 9, 1, '2020-01-15', '2020-02-05', '2020-02-03'),
186 (10, 10, 10, 1, '2020-03-12', '2020-04-02', '2020-03-28'),
187 (11, 1, 11, 2, '2020-05-14', '2020-06-04', '2020-05-28'),
188 (12, 2, 12, 2, '2020-07-21', '2020-08-11', '2020-08-02'),
189 (13, 3, 13, 2, '2020-09-25', '2020-10-16', '2020-10-15'),
190 (14, 4, 1, 2, '2020-11-08', '2020-11-29', '2020-11-24'),
191 (15, 5, 2, 2, '2021-01-03', '2021-01-24', '2021-01-19'),
192 (16, 6, 3, 2, '2021-02-18', '2021-03-11', '2021-03-12'),
193 (17, 17, 4, 2, '2021-04-27', '2021-05-18', '2021-05-20'),
194 (18, 18, 5, 2, '2021-06-13', '2021-07-04', '2021-06-28'),
195 (19, 19, 6, 2, '2021-08-15', '2021-09-05', '2021-09-03'),
196 (20, 20, 7, 2, '2021-10-21', '2021-11-11', '2021-11-05'),
197 (21, 21, 1, 3, '2021-11-29', '2021-12-20', NULL),
198 (22, 22, 2, 3, '2022-01-10', '2022-01-31', '2022-01-25'),
199 (23, 23, 3, 3, '2022-02-07', '2022-02-28', '2022-02-23'),
200 (24, 24, 4, 3, '2022-03-11', '2022-04-01', '2022-03-28'),
201 (25, 25, 5, 3, '2022-04-28', '2022-05-19', '2022-05-18'),
202 (26, 26, 6, 3, '2022-06-22', '2022-07-13', '2022-07-08'),
203 (27, 27, 7, 3, '2022-08-04', '2022-08-25', '2022-08-23'),
204 (28, 28, 8, 3, '2022-09-13', '2022-10-04', '2022-09-28'),
205 (29, 29, 9, 3, '2022-10-16', '2022-11-06', '2022-11-05'),
206 (30, 30, 8, 3, '2022-11-21', '2022-12-12', '2022-12-05'),
207 (31, 1, 9, 4, '2022-12-28', '2023-01-18', NULL),
208 (32, 2, 1, 4, '2023-01-29', '2023-02-19', NULL);
209
```

The 'Data Output' tab shows the result of the query: 'INSERT 0 40'. A message at the bottom states: 'Query returned successfully in 101 msec.' The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.101'.

The following image shows the SQL query of inserting values for the Author Entity.

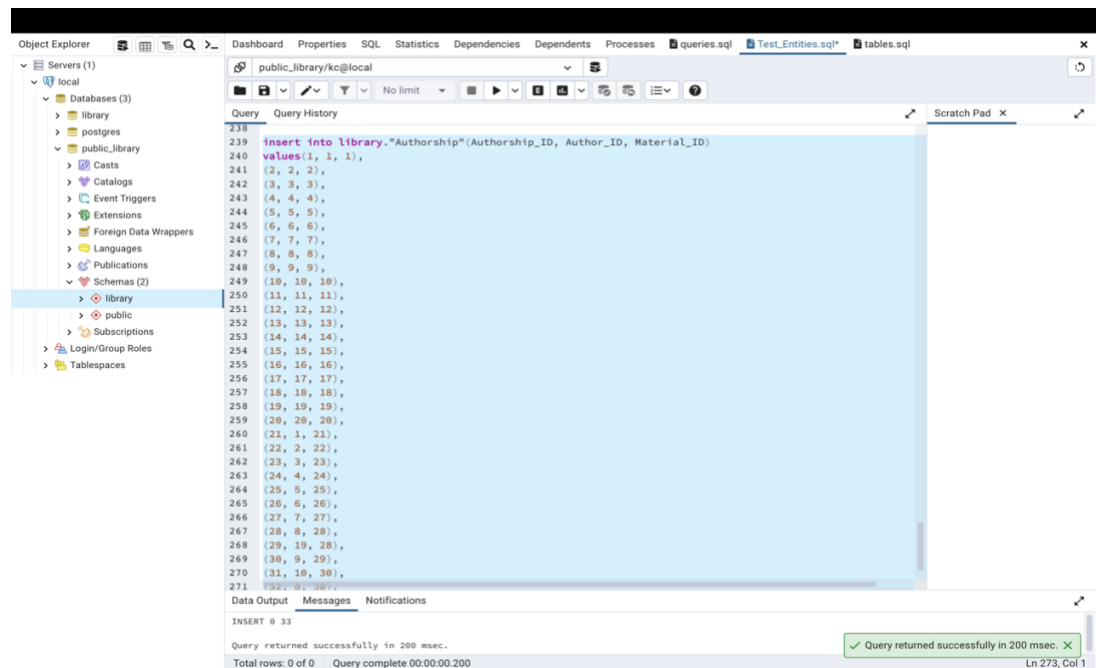


The screenshot shows the SQL Developer interface with the 'public_library/kc@local' connection. The 'Object Explorer' on the left shows the database structure, including the 'library' schema. The main window displays an SQL query to insert data into the 'Author' table. The query is as follows:

```
insert into library."Author"(Author_ID, Name, Birth_Date, Nationality)
values(1, 'Jane Austen', '1775-12-16', 'British'),
(2, 'Ernest Hemingway', '1899-07-21', 'American'),
(3, 'George Orwell', '1903-06-25', 'British'),
(4, 'Scott Fitzgerald', '1896-09-24', 'American'),
(5, 'J.K. Rowling', '1965-07-31', 'British'),
(6, 'Mark Twain', '1835-11-30', 'American'),
(7, 'Leo Tolstoy', '1828-09-09', 'Russian'),
(8, 'Virginia Woolf', '1882-01-25', 'British'),
(9, 'Gabriel Márquez', '1927-03-06', 'Colombian'),
(10, 'Charles Dickens', '1812-02-07', 'British'),
(11, 'Harper Lee', '1926-04-28', 'American'),
(12, 'Oscar Wilde', '1854-10-16', 'Irish'),
(13, 'William Shakespeare', '1564-04-26', 'British'),
(14, 'Franz Kafka', '1883-07-03', 'Czech'),
(15, 'James Joyce', '1882-02-02', 'Irish'),
(16, 'J.R.R. Tolkien', '1892-01-03', 'British'),
(17, 'Emily Brontë', '1818-07-30', 'British'),
(18, 'Toni Morrison', '1931-02-18', 'American'),
(19, 'Fyodor Dostoevsky', '1821-11-11', 'Russian'),
(20, 'Lucas Piki', '1847-10-16', 'British');
```

The 'Data Output' tab at the bottom shows the execution results: 'INSERT 0 20', 'Query returned successfully in 48 msec.', and 'Total rows: 0 of 0'. A green status bar at the bottom right indicates 'Query returned successfully in 48 msec.'.

The following image shows the SQL query of inserting values for the Authorship Entity.



The screenshot shows the SQL Developer interface with the 'public_library/kc@local' connection. The 'Object Explorer' on the left shows the database structure, including the 'library' schema. The main window displays an SQL query to insert data into the 'Authorship' table. The query is as follows:

```
insert into library."Authorship"(Authorship_ID, Author_ID, Material_ID)
values(1, 1, 1),
(2, 2, 2),
(3, 3, 3),
(4, 4, 4),
(5, 5, 5),
(6, 6, 6),
(7, 7, 7),
(8, 8, 8),
(9, 9, 9),
(10, 10, 10),
(11, 11, 11),
(12, 12, 12),
(13, 13, 13),
(14, 14, 14),
(15, 15, 15),
(16, 16, 16),
(17, 17, 17),
(18, 18, 18),
(19, 19, 19),
(20, 20, 20),
(21, 1, 21),
(22, 2, 22),
(23, 3, 23),
(24, 4, 24),
(25, 5, 25),
(26, 6, 26),
(27, 7, 27),
(28, 8, 28),
(29, 9, 29),
(30, 9, 29),
(31, 10, 30),
(32, 8, 30);
```

The 'Data Output' tab at the bottom shows the execution results: 'INSERT 0 33', 'Query returned successfully in 200 msec.', and 'Total rows: 0 of 0'. A green status bar at the bottom right indicates 'Query returned successfully in 200 msec.'.

Queries:

1. Which materials are currently available in the library?

The following image shows the execution of the query for retrieving the distinct materials that are currently available in the library along with the title of the material. Here using join the material and borrow tables are joined on material_id.

The screenshot shows a database query editor with the following query:

```
--1. Which materials are currently available in the library?  
2  
3 select distinct(m.material_id, m.title from library."Material" m  
4 inner join library."Borrow" b on b.material_id=m.material_id  
5 where b.return_date is not null  
6 order by m.material_id;
```

The Data Output tab shows the following results:

material_id [PK] integer	title character varying
1	The Catcher in the Rye
2	To Kill a Mockingbird
3	The Da Vinci Code
4	The Hobbit
5	The Shining
6	Pride and Prejudice
7	The Great Gatsby
8	Moby Dick
9	Crime and Punishment
10	The Hitchhikers Guide to the Galaxy
11	The Catch-22
12	The Call of Cthulhu
13	Harry Potter and the Philosophers Stone
14	A Tale of Two Cities
15	The Iliad
16	The Odyssey
17	The Divine Comedy
18	The Grapes of Wrath
19	The Old Man and the Sea
20	The Count of Monte Cristo
21	A Midsummer Nights Dream

Total rows: 21 of 21 Query complete 00:00:00.103 Ln 37, Col 1

2. Which materials are currently overdue? (Suppose today is 04/01/2023, and show the borrow date and due date of each material)

The screenshot shows a database query editor with the following query:

```
6 order by m.material_id;  
7  
8 2. Which materials are currently overdue?  
9 (Suppose today is 04/01/2023, and show the borrow date and due date of each material)  
10  
11 select br.borrow_date, br.due_date, br.return_date, mr.title as material_title  
12 from library."Borrow" br  
13 inner join library."Material" mr on mr.material_id = br.material_id  
14 where br.due_date < '2023-04-01' and br.return_date is NULL;  
15  
16 3. What are the top 10 most borrowed materials in the library?  
17 (Show the title of each material and order them based on their available counts)
```

The Data Output tab shows the following results:

borrow_date character varying	due_date character varying	return_date character varying	material_title character varying
2021-11-29	2021-12-05	[null]	Frankenstein
2022-12-28	2023-01-18	[null]	The Catcher in the Rye
2023-01-23	2023-02-13	[null]	To Kill a Mockingbird
2023-03-01	2023-03-22	[null]	The Hobbit
2023-03-10	2023-03-31	[null]	The Shining

Total rows: 5 of 5 Query complete 00:00:00.045 Ln 14, Col 61

In this query the considered overdue date is 2023-04-01 and the materials which are overdue are displayed. With the help of join the material and borrow tables are joined on material_id where showing the materials that are due for 2023-04-01.

- What are the top 10 most borrowed materials in the library?
(Show the title of each material and order them based on their available counts)

The top 10 most borrowed materials in the library are retrieved from the borrow table showing the material count and title using group by command.

The screenshot shows the SQL Developer interface with a query window containing the following SQL code:

```

3. What are the top 10 most borrowed materials in the library?
(Show the title of each material and order them based on their available counts)

select mt.material_id, mt.title as material_title, count(mt.material_id) as material_count from library."B
join library."Material" mt on mt.material_id=material_count
where bw.return_date is NOT NULL
group by mt.material_id
order by material_count desc
limit 10;

```

The Data Output pane displays the results of the query:

material_id	material_title	material_count
3	The Da Vinci Code	3
5	The Shining	2
2	To Kill a Mockingbird	2
4	The Hobbit	2
1	The Catcher in the Rye	2
6	Pride and Prejudice	2
10	The Hitchhikers Guide to the Galaxy	1
20	Harry Potter and the Philosophers Stone	1
7	The Great Gatsby	1
29	The Count of Monte Cristo	1

Total rows: 10 of 10 Query complete 00:00:00.051 Ln 26, Col 1

- How many books has the author Lucas Piki written?

The screenshot shows the SQL Developer interface with a query window containing the following SQL code:

```

4.How many books has the author Lucas Piki written?

select m.title, au.name, count(a.authorship_id) as books_count from library."Authorship" a
join library."Material" m on m.material_id=a.material_id
join library."Author" au on au.author_id=a.author_id
where au.name='Lucas Piki'
group by m.title, au.name;

```

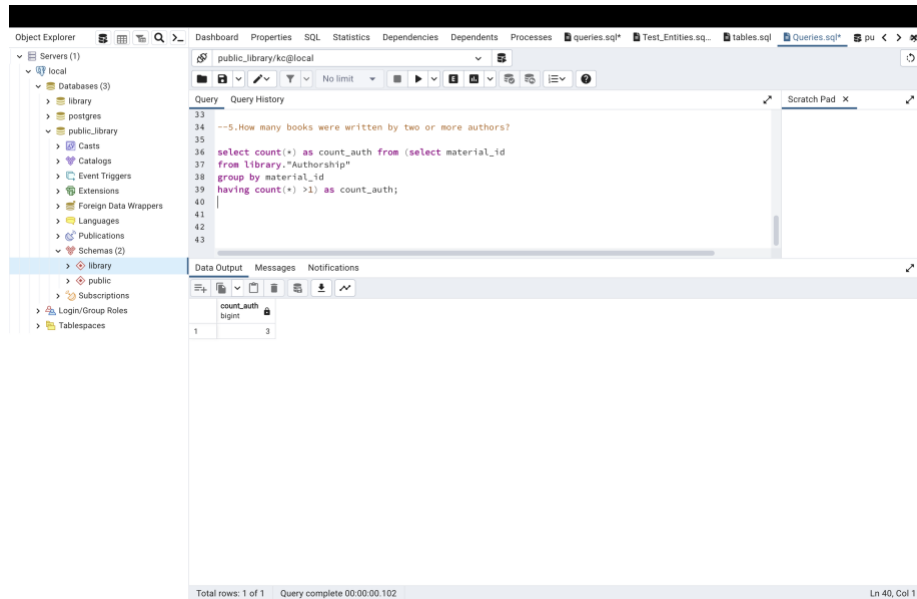
The Data Output pane displays the results of the query:

title	name	books_count
Harry Potter and the Philosophers Stone	Lucas Piki	1

Total rows: 1 of 1 Query complete 00:00:00.037 Ln 33, Col 1

In the above query we can see that the author Lucas Piki written only one book which is Harry Potter and the Philosophers stone which is obtained by the count of the authorship_id from the authorship table.

5. How many books were written by two or more authors?



The screenshot shows a database IDE with a query editor and a results pane. The query is a nested query that counts the number of authors for each material_id from the authorship table, where the count is greater than or equal to 2. The results pane shows a single row with the count of authors (3) for a specific material_id.

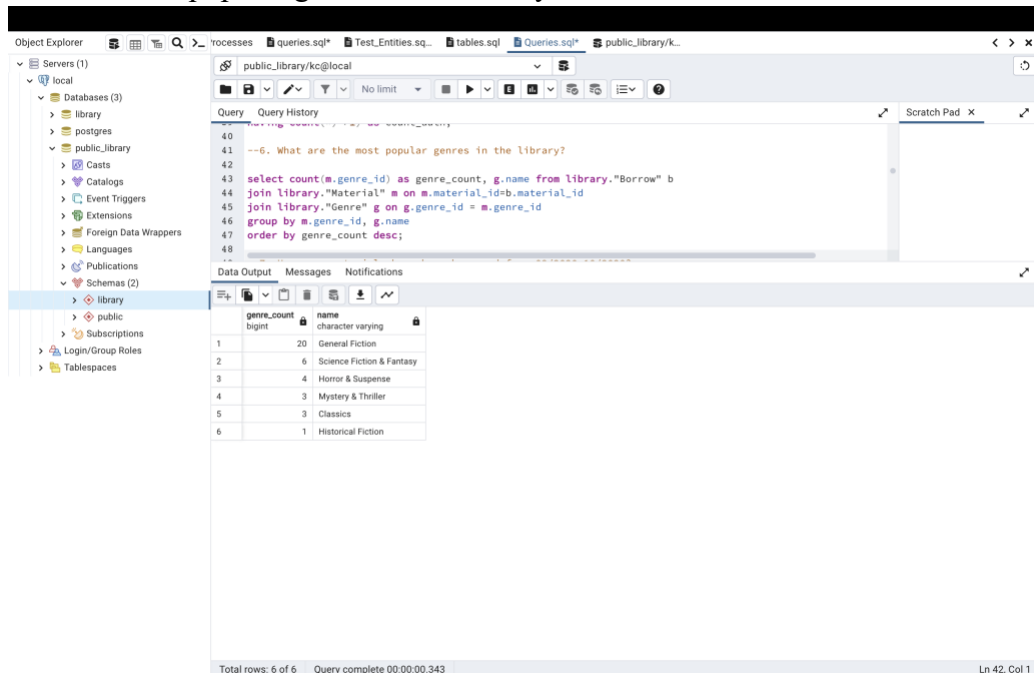
```
33
34 --5.How many books were written by two or more authors?
35
36 select count(*) as count_auth from (select material_id
37 from library."Authorship"
38 group by material_id
39 having count(*) >= 1) as count_auth;
40
41
42
43
```

count_auth
3

Total rows: 1 of 1 Query complete 00:00:00.102 Ln 40, Col 1

Here the count of authors is displayed where two or more authors written a material using nested query, here material_id has been considered from the authorship table.

6. What are the most popular genres in the library?



The screenshot shows a database IDE with a query editor and a results pane. The query is a join query that counts the number of books for each genre from the library table, ordered by the count in descending order. The results pane shows a table with 6 rows, listing the genre and the count of books.

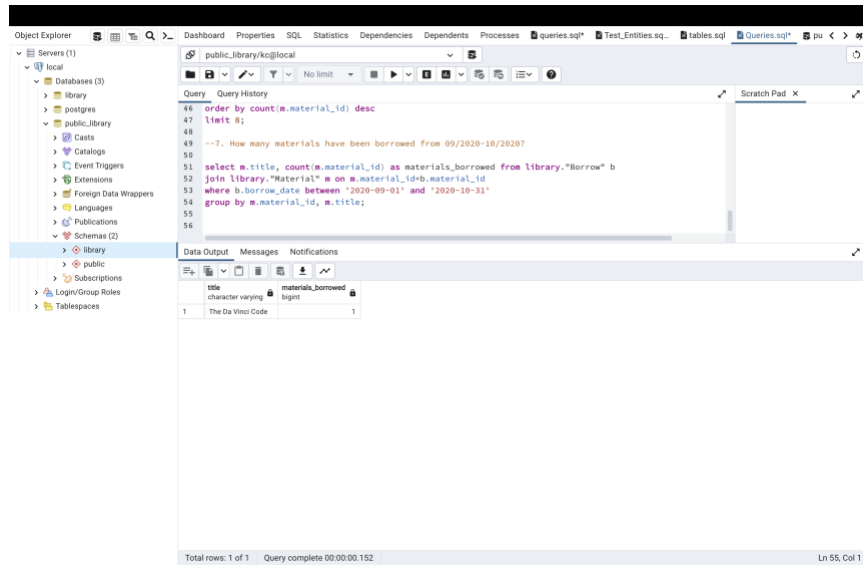
```
40
41 --6. What are the most popular genres in the library?
42
43 select count(m.genre_id) as genre_count, g.name from library."Borrow" b
44 join library."Material" m on m.material_id=b.material_id
45 join library."Genre" g on g.genre_id = m.genre_id
46 group by m.genre_id, g.name
47 order by genre_count desc;
48
```

genre_count	name
20	General Fiction
6	Science Fiction & Fantasy
4	Horror & Suspense
3	Mystery & Thriller
3	Classics
1	Historical Fiction

Total rows: 6 of 6 Query complete 00:00:00.343 Ln 42, Col 1

The most popular genres considered are retrieved from the borrow table. By using the count of genre_id we can get the top borrowed genres from the library.

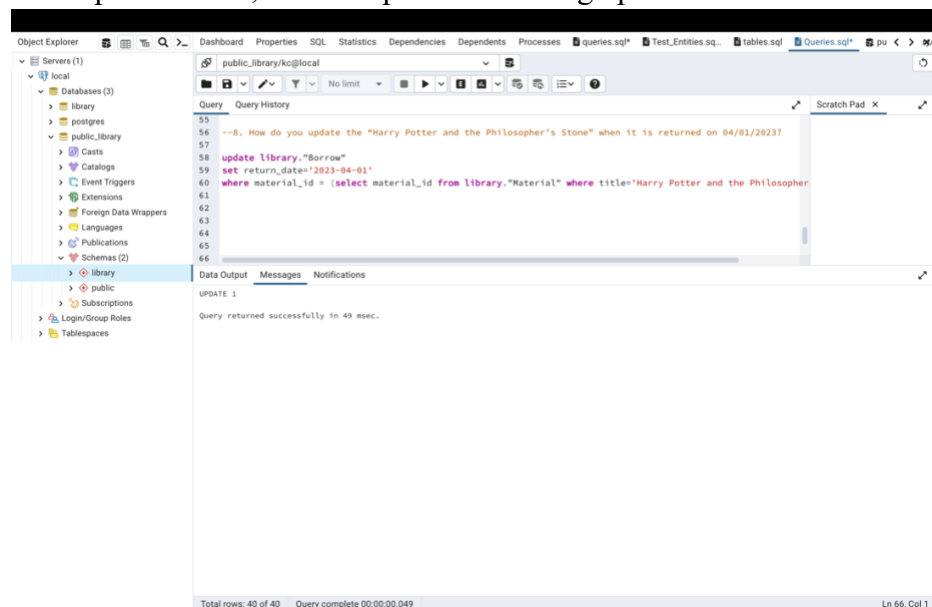
7. How many materials have been borrowed from 09/2020-10/2020?



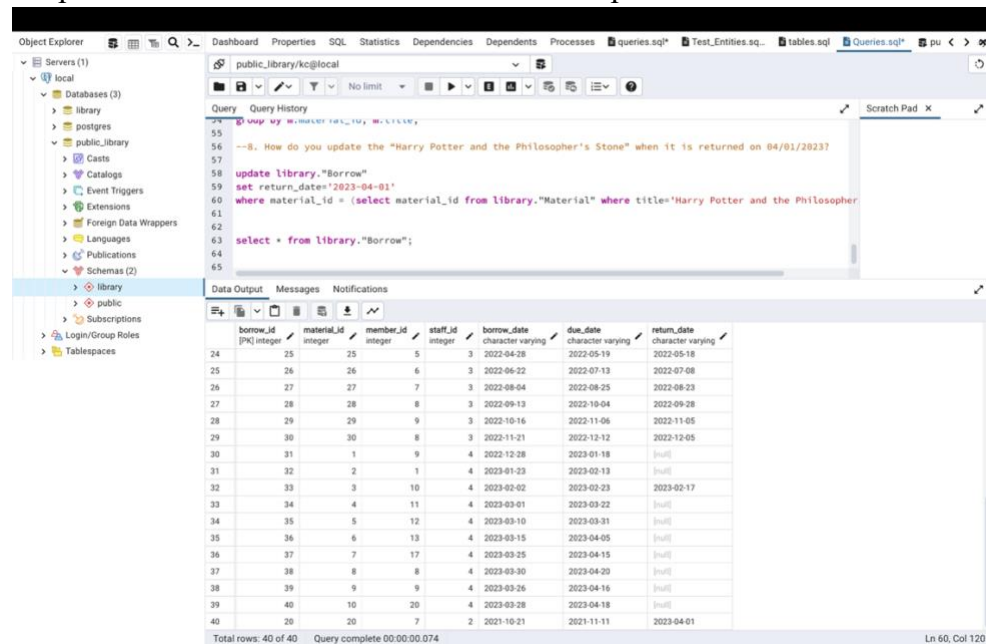
In this query the number of materials borrowed between 2020-09-01 and 2020-10-31 has been displayed using join, where and group by.

8. How do you update the “Harry Potter and the Philosopher's Stone” when it is returned on 04/01/2023?

In the following image shows the query for updating the return date of material Harry Potter and the Philosopher's Stone, which is performed using update command.



The following image shows the updated borrow table, in which the title 'Harry Potter and the Philosopher's Stone' and the return date has been updated where the material id is 20.



Query

```

55
56 --8. How do you update the "Harry Potter and the Philosopher's Stone" when it is returned on 04/01/2023?
57
58 update library."Borrow"
59 set return_date='2023-04-01'
60 where material_id = (select material_id from library."Material" where title='Harry Potter and the Philosopher
61
62
63 select * from library."Borrow";
64
65

```

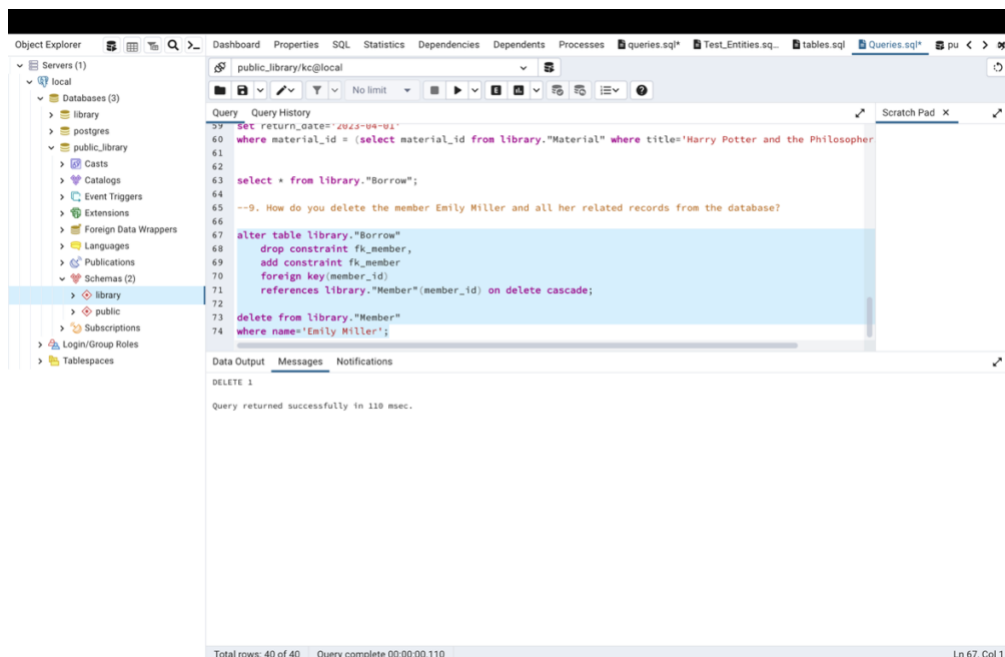
Data Output

borrow_id	material_id	member_id	staff_id	borrow_date	due_date	return_date
24	25	25	5	3	2022-04-28	2022-05-19
25	26	26	6	3	2022-06-22	2022-07-13
26	27	27	7	3	2022-08-04	2022-08-25
27	28	28	8	3	2022-09-13	2022-10-04
28	29	29	9	3	2022-10-16	2022-11-06
29	30	30	8	3	2022-11-21	2022-12-05
30	31	1	9	4	2022-12-28	2023-01-18
31	32	2	1	4	2023-01-23	2023-02-13
32	33	3	10	4	2023-02-02	2023-02-23
33	34	4	11	4	2023-03-01	2023-03-22
34	35	5	12	4	2023-03-10	2023-03-31
35	36	6	13	4	2023-03-15	2023-04-05
36	37	7	17	4	2023-03-25	2023-04-15
37	38	8	8	4	2023-03-30	2023-04-20
38	39	9	9	4	2023-03-26	2023-04-16
39	40	10	20	4	2023-03-28	2023-04-18
40	20	20	7	2	2021-10-21	2021-11-11

Total rows: 40 of 40 Query complete 00:00:00.074 Ln 60, Col 120

9. How do you delete the member Emily Miller and all her related records from the database?

The following image shows the query to delete a record from the member table named member Emily Miller. For this first altered the Borrow table using ALTER command to perform the on delete cascade, then deleted the record.



Query

```

57 set return_date='2023-04-01'
60 where material_id = (select material_id from library."Material" where title='Harry Potter and the Philosopher
61
62
63 select * from library."Borrow";
64
65
66 --9. How do you delete the member Emily Miller and all her related records from the database?
67
68 alter table library."Borrow"
69 drop constraint fk_member,
70 add constraint fk_member
71 foreign key(member_id)
72 references library."Member"(member_id) on delete cascade;
73
74 delete from library."Member"
75 where name='Emily Miller';

```

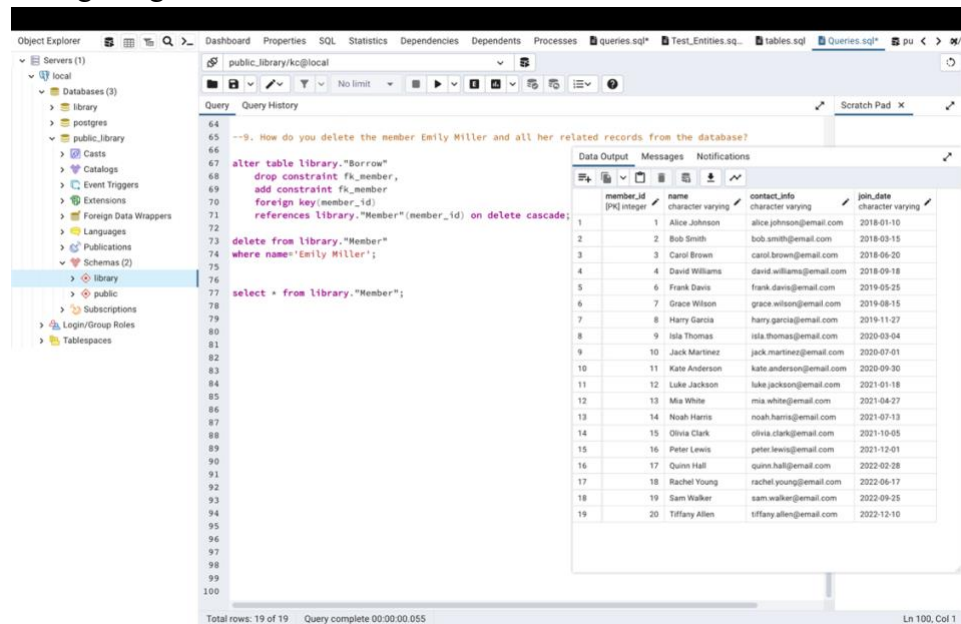
Data Output

name
Emily Miller

DELETE 1
Query returned successfully in 118 msec.

Total rows: 40 of 40 Query complete 00:00:00.110 Ln 67, Col 1

The following image shows the record that is deleted from member table.



10. How do you add the following material to the database?

Title: New book

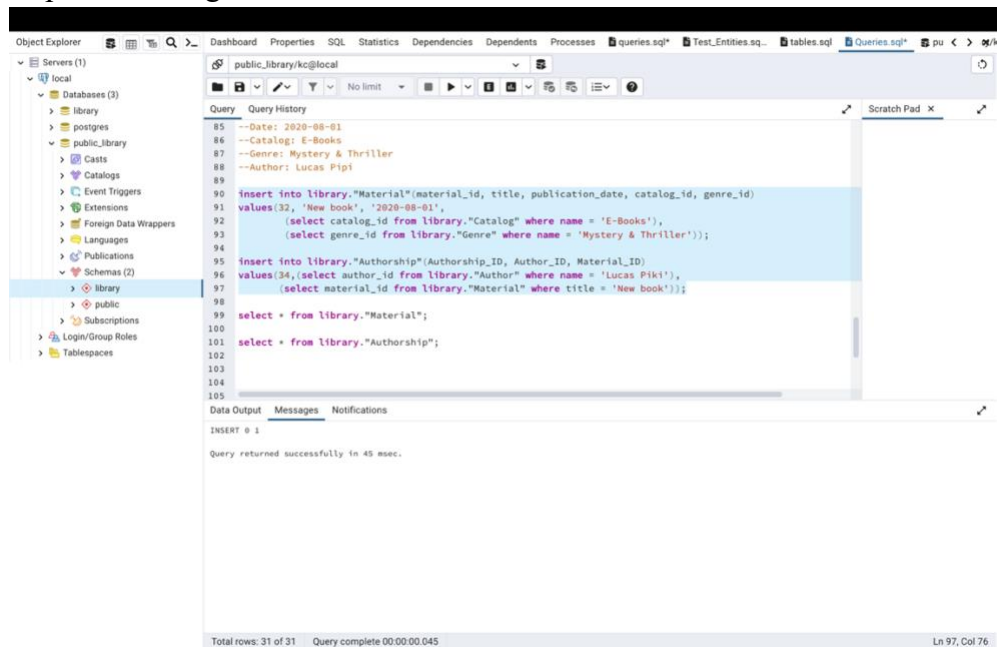
Date: 2020-08-01

Catalog: E-Books

Genre: Mystery & Thriller

Author: Lucas Piki

The following image shows the query for inserting the above values in material and authorship tables using insert command.



The following image shows the inserted values of the material title in the material table.

The screenshot shows a PostgreSQL database interface. The left sidebar displays the 'Object Explorer' with a tree view of the database structure, including 'Servers (1)', 'local', 'Databases (3)', 'library', 'postgres', 'public_library', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', and 'Schemas (2)'. The main window shows a SQL query in the 'Query' tab, which is a complex insert statement using subqueries to populate the 'library' database. The 'Data Output' tab at the bottom displays the results of the query, showing a table with columns: 'material_id', 'title', 'publication_date', 'catalog_id', and 'genre_id'. The table contains 32 rows of data, including titles like 'The Catch-22', 'The Picture of Dorian Gray', 'The Call of Cthulhu', 'Harry Potter and the Philosophers Stone', 'Frankenstein', 'A Tale of Two Cities', 'The Road', 'The Odyssey', 'The Brothers Karamazov', 'The Divine Comedy', 'The Grapes of Wrath', 'The Old Man and the Sea', 'The Count of Monte Cristo', 'A Midsummer Nights Dream', 'The Tricky Book', and 'New book'.

material_id	title	publication_date	catalog_id	genre_id
17	The Catch-22	1961-10-11	7	1
18	The Picture of Dorian Gray	1890-07-01	8	1
19	The Call of Cthulhu	1928-02-01	9	4
20	Harry Potter and the Philosophers Stone	1997-09-26	10	3
21	Frankenstein	1818-01-01	6	4
22	A Tale of Two Cities	1859-04-30	7	1
23	The Road	1950-01-01	8	6
24	The Odyssey	1725-01-01	9	6
25	The Brothers Karamazov	1880-01-01	10	1
26	The Divine Comedy	1320-01-01	6	6
27	The Grapes of Wrath	1939-04-14	7	1
28	The Old Man and the Sea	1952-09-01	8	1
29	The Count of Monte Cristo	1844-01-01	9	1
30	A Midsummer Nights Dream	1596-01-01	10	7
31	The Tricky Book	1888-01-01	10	7
32	New book	2020-08-01	3	2

The advanced querying techniques such as joins, aggregation, and subqueries are used when dealing with large set of databases with multiple tables. The data can be retrieved and analyzed efficiently in an organized way and can extract information from a large data set, reducing the amount of irrelevant data with the help of these techniques.

Joins: combines rows from two or more tables with the help of related column between them.

Aggregation: summarizes data using functions such as COUNT, SUM, AVG, MIN, and MAX.

Subqueries: used to nest one query inside another to perform a more complex operation, such as filtering data based on a condition in another table.

Design to extend the existing database system:

Part-A:

Alert staff about overdue materials on a daily-basis?

We can run a SQL query on a daily basis using Cron which is a time-based job scheduler in Unix-like operating systems, including Linux and macOS. It allows us to schedule jobs (such as running a PostgreSQL query) to run automatically at specified intervals.

The query for the overdue is

```
select b.borrow_date, b.due_date, b.return_date, m.title as material_title from library."Borrow" b
join library."Material" m on m.material_id=b.material_id
where cast(b.due_date as date) = current_date and b.return_date is NULL ;
```

Steps for PostgreSQL query to run on a daily basis using cron:

- Making sure the query for overdue that we want to run on a daily basis is saved on computer.
- Open the terminal or command prompt and type "crontab -e" which will open up the cron job list.
- Adding a new line to the cron job list that can specify the time and frequency of the SQL query. The syntax for a cron job is

```
0 0 * * * /kc/local/bin/psql -U kc -d public_library -f /Users/kc/Downloads/overdue.sql >> /
Users/kc/Downloads /overdueresult.log 2>&1
```

In the above syntax 0 0 * * * represents minutes, hours, day of month, month, day of week where * represents every day.

From the above command line, we can say that the job runs at 12 am every day, every month where the results will be saved as a logfile in the mentioned path. From this logfile we can alert the staff members by sending an email to them.

Part B:

Automatically deactivate the membership based on the member's overdue occurrence (>= three times). And reactivate the membership once the member pays the overdue fee.

Step 1:

Alter the table Members by adding new column 'active' which tells whether member is active or not. We can use below query to alter the table

```
ALTER TABLE member
ADD COLUMN active BOOLEAN DEFAULT TRUE;
```

STEP 2:

Create a new table to track the number of overdue by the members. This table gives us the information of total overdue by a member. Below is the query to create this table

```

Create table overdues(
    Overdue_id int,
    Member_id int ,
    Borrow_id int,
    Overdue_fee int,
    Overdue_fee_paid boolean,
    constraint fk_member
        foreign key(member_ID)
        references library."Member"(Member_ID),
    constraint fk_borrow
        foreign key(borrow_ID)
        references library."Borrow "( borrow_ID)
);

```

Step 3:

Insert data into overdues table whenever member past the due_date using the results from below query

```

Select borrow_id, member_id from borrow where cast(b.due_date as date) = current_date and
b.return_date is NULL;

```

Step 4:

Get the list of all members from overdues table whose overdues count is ≥ 3 using below query.

```

Select member_id, count(member_id) as overdue_count from overdues where overdue_count  $\geq$ 
3 group by member_id;

```

Step 5:

Update the 'active' column of members table for all the members obtained from step 4 using below query.

```

Update members set active = False where member_id = <member_id>

```

Step 6:

Once the overdue fee is paid by the member, we'll update overdues table and reactivate the member using below queries.

```

Update overdues set overdue_fee_paid = TRUE where member_id = <member_id>

```

```

Update members set active = TRUE where member_id = <member_id>

```