**Database Project   
Santa Clara University**

**Title**: Ebook Repository

In this application, customers can search the ebooks in the database using search criteria like Author name, Book title, publisher name, Genre, Price, availability to rent. The database will provide search results based on the criteria and display the results to the customer. If the customer wants to rent a book, the database will check if it is available for rent. If the book is available for rent then the book will be issued to the customer.

**Entities:**

1. **Books:**

Attributes: Book\_id, Title, Genre, Price, Author\_id, Publisher\_name

1. **Author:**

Attributes: Author\_id, Author\_name, Author\_address, Email

1. **Customers:**

Attributes: Custome\_id, Customer\_name, Email\_id, Address

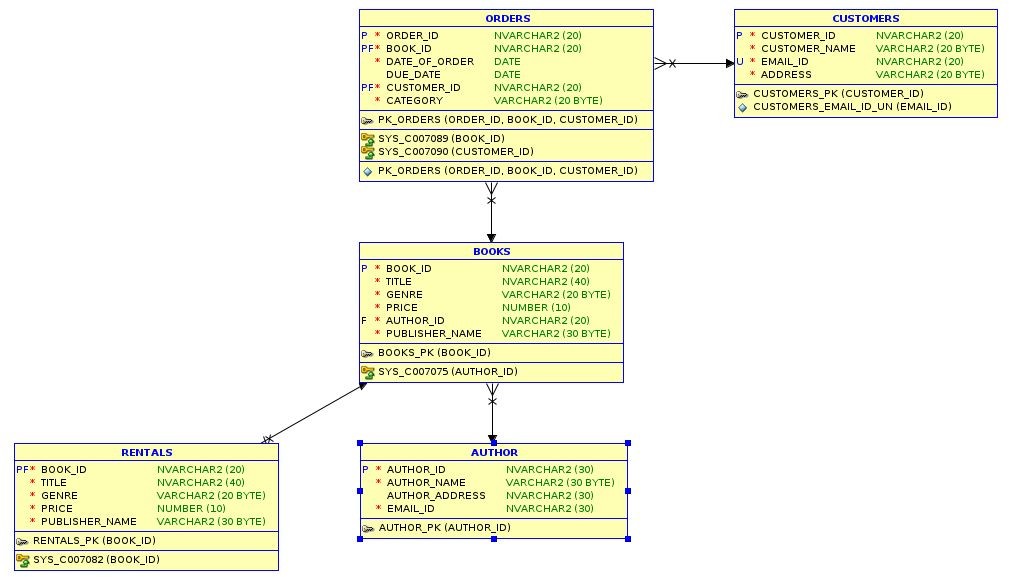
1. **Rentals:**

Attributes: Book\_id, Title, Genre, Price, Publisher\_name

1. **Orders:**

Attributes: Order\_id, Book\_id, Date\_of\_order, Due\_date, Customer\_id, Cat

**ER Diagram:**



**// Table Customers**

CREATE TABLE **customers** (

customer\_id NVARCHAR2(20) NOT NULL,

customer\_name VARCHAR2(20) NOT NULL,

email\_id NVARCHAR2(20) NOT NULL,

address VARCHAR2(20) NOT NULL,

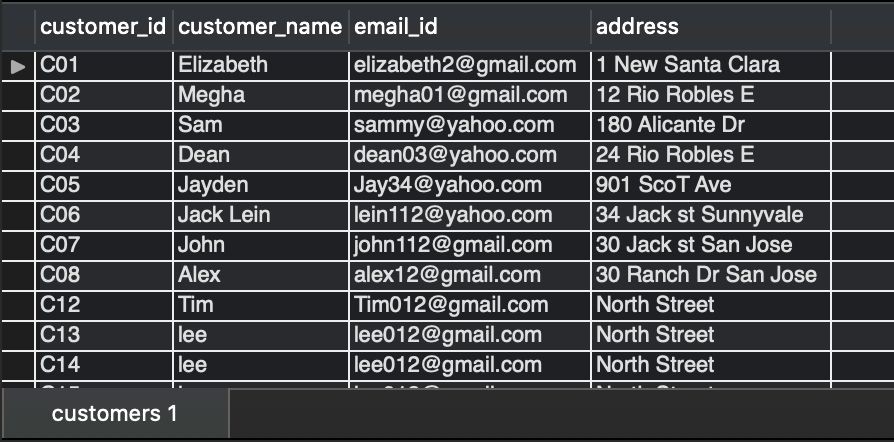
Primary key(customer\_id),

UNIQUE(email\_id)

);

INSERT INTO customers VALUES ('C01','Elizabeth','eliza2@gmail.com','1 New Santa Clara'); INSERT INTO customers VALUES ('C02','Megha','megha01@gmail.com','12 Rio Robles E'); INSERT INTO customers VALUES ('C03','Sam','sammy@yahoo.com','180 Alicante Dr'); INSERT INTO customers VALUES ('C04','Dean','dean03@yahoo.com','24 Rio Robles E'); INSERT INTO customers VALUES('C05','Jayden','Jay34@yahoo.com','901 ScoT Ave');

INSERT INTO customers VALUES('C06','Jack Lein','lein112@yahoo.com','34 Jack st Sunnyvale'); INSERT INTO customers VALUES('C07','John','john112@gmail.com','30 Jack st San Jose'); INSERT INTO customers VALUES('C08','Alex','alex12@gmail.com','30 Ranch Dr San Jose');



**//Table Author**

CREATE TABLE **author** (

author\_id NVARCHAR2(30) NOT NULL,

author\_name VARCHAR2(30) NOT NULL,

author\_address NVARCHAR2(30),

email\_id NVARCHAR2(30) NOT NULL,

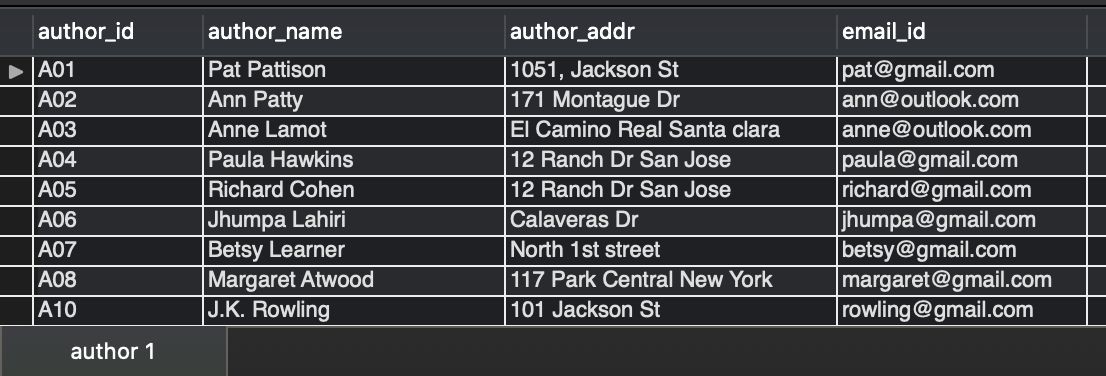
PRIMARY KEY(author\_id)

);

INSERT INTO author VALUES ('A01','Pat Pattison','1051, Jackson St.','pat@gmail.com'); INSERT INTO author VALUES ('A02','Ann Patty','171 Montague Dr','ann@outlook.com');

INSERT INTO author VALUES ('A03','Anne Lamot','El Camino Real Santa clara','anne@outlook.com'); INSERT INTO author VALUES ('A04','Paula Hawkins','12 Ranch Dr San Jose','paula@gmail.com'); INSERT INTO author VALUES ('A05','Richard Cohen','12 Ranch Dr San Jose','richard@gmail.com'); INSERT INTO author VALUES('A06','Jhumpa Lahiri','Calaveras Dr','jhumpa@gmail.com');

INSERT INTO author VALUES('A07','Betsy Learner','North 1st street','betsy@gmail.com'); INSERT INTO author VALUES('A08','Margaret Atwood','117 Park Central New York','t');



**//Table books**

CREATE TABLE **books** (

book\_id NVARCHAR2(20) NOT NULL,   
title NVARCHAR2(40) NOT NULL,   
genre VARCHAR2(20) NOT NULL,   
price NUMBER(10,0) NOT NULL,

author\_id NVARCHAR2(20) NOT NULL,   
publisher\_name VARCHAR2(30) NOT NULL,   
PRIMARY KEY(book\_id),

FOREIGN KEY(author\_id) references author(author\_id) ON DELETE CASCADE

);

INSERT INTO books VALUES ('B01','Nineteen Eighty Four','Literature','16','A01','Harvill Secker'); INSERT INTO books VALUES ('B02','Animal Farm','Literature','14','A02','Harvill Secker,');

INSERT INTO books VALUES ('B03','The Future of Life','Science', '22','A03','Hache Te Livre');

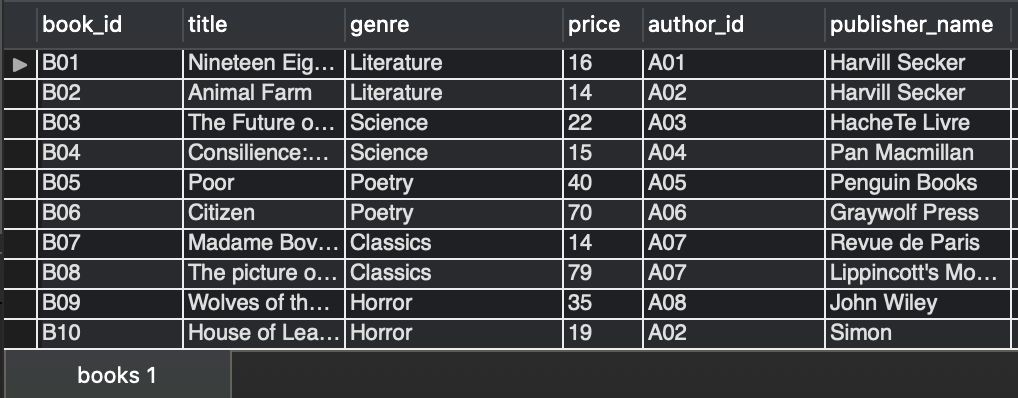
INSERT INTO books VALUES ('B04','Consilience: The Unity of Knowledge','Science','15','A03','Pan Macmillan'); INSERT INTO books VALUES ('B05','Poor','Poetry','40','A05','Penguin Books');

INSERT INTO books VALUES ('B06','Citizen','Poetry','70','A06','Graywolf Press');

INSERT INTO books VALUES ('B07','Madame Bovary','Classics’','14','A07','Revue de Paris');

INSERT INTO books VALUES ('B08','The picture of Dorain Gray','Classics’','79','A07','Lippincotts Monthly Magazine'); INSERT INTO books VALUES ('B09','Wolves of the Calla','Horror','35','A08','John Wiley');

INSERT INTO books VALUES ('B10','House of Leaves','Horror','19','A02','c');



**//Table Rentals(books eligible for rent)**

CREATE TABLE **rentals** (

book\_id NVARCHAR2(20) NOT NULL,   
title NVARCHAR2(40) NOT NULL,   
genre VARCHAR2(20) NOT NULL,   
price NUMBER(10,0) NOT NULL,

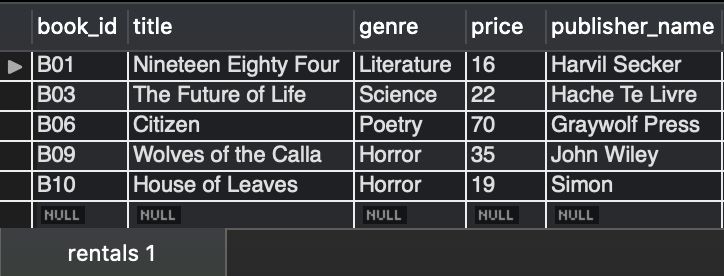
publisher\_name VARCHAR2(30) NOT NULL, PRIMARY KEY (book\_id),

FOREIGN KEY (book\_id) REFERENCES books(book\_id) ON DELETE CASCADE

);

INSERT INTO rentals VALUES ('B01', 'Nineteen Eighty Four','Literature','16','Harvill Secker'); INSERT INTO rentals VALUES ('B03','The Future of Life','Science', '22','Hache Te Livre'); INSERT INTO rentals VALUES ('B06','Citizen','Poetry','70','Graywolf Press');

INSERT INTO rentals VALUES ('B09','Wolves of the Calla','Horror','35','John Wiley'); INSERT INTO rentals VALUES ('B10','House of Leaves','Horror','19','Simon');



**//Table orders**

CREATE TABLE **orders** (

order\_id NVARCHAR2(20) NOT NULL, book\_id NVARCHAR2(20) NOT NULL,

date\_of\_order DATE NOT NULL,

due\_date DATE,

customer\_id NVARCHAR2(20) NOT NULL,   
 category VARCHAR2(20) NOT NULL,

CONSTRAINT PK\_ORDERS PRIMARY KEY(order\_id, book\_id, customer\_id), FOREIGN KEY (book\_id) REFERENCES BOOKS(book\_id) ON DELETE CASCADE,

FOREIGN KEY (customer\_id) REFERENCES CUSTOMERS(customer\_id) ON DELETE CASCADE

);

INSERT INTO orders VALUES ('1001','B01','23-05-2020', '23-11-2020','C01', 'rented');

INSERT INTO orders VALUES ('1002','B02','20-07-2020', '','C02', 'purchased');

INSERT INTO orders VALUES ('1003','B03','05-07-2020','05-09-2020','C03', 'rented');

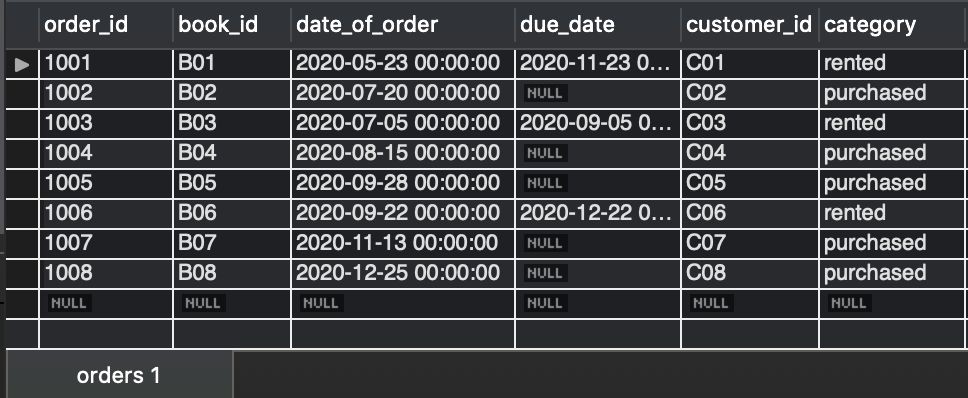
INSERT INTO orders VALUES ('1004','B04','15-08-2020', '', 'C04', 'purchased');

INSERT INTO orders VALUES ('1005','B05','28-09-2020', '', 'C05', 'purchased');

INSERT INTO orders VALUES ('1006','B06','22-09-2020','22-12-2020','C06', 'rented');

INSERT INTO orders VALUES ('1007','B07','13-11-2020','','C07', 'purchased');

INSERT INTO orders VALUES ('1008','B08','25-12-2020', '','C08', 'purchased');



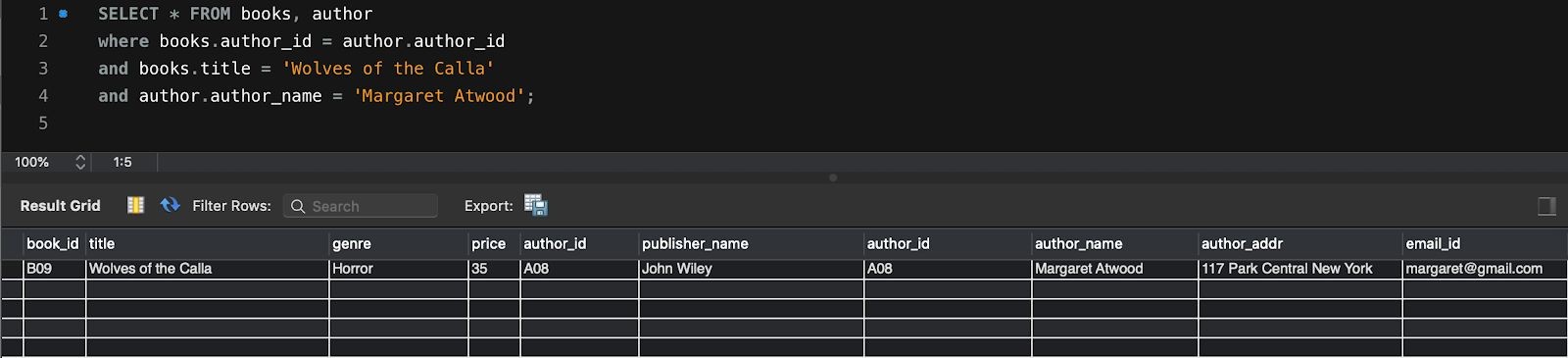
Queries :

1. **SELECT**:

# Select e-books with title = Wolves of the Calla and author name = Margaret Atwood

SELECT \* from books B, author A   
 where B.author\_id = a.author\_id and

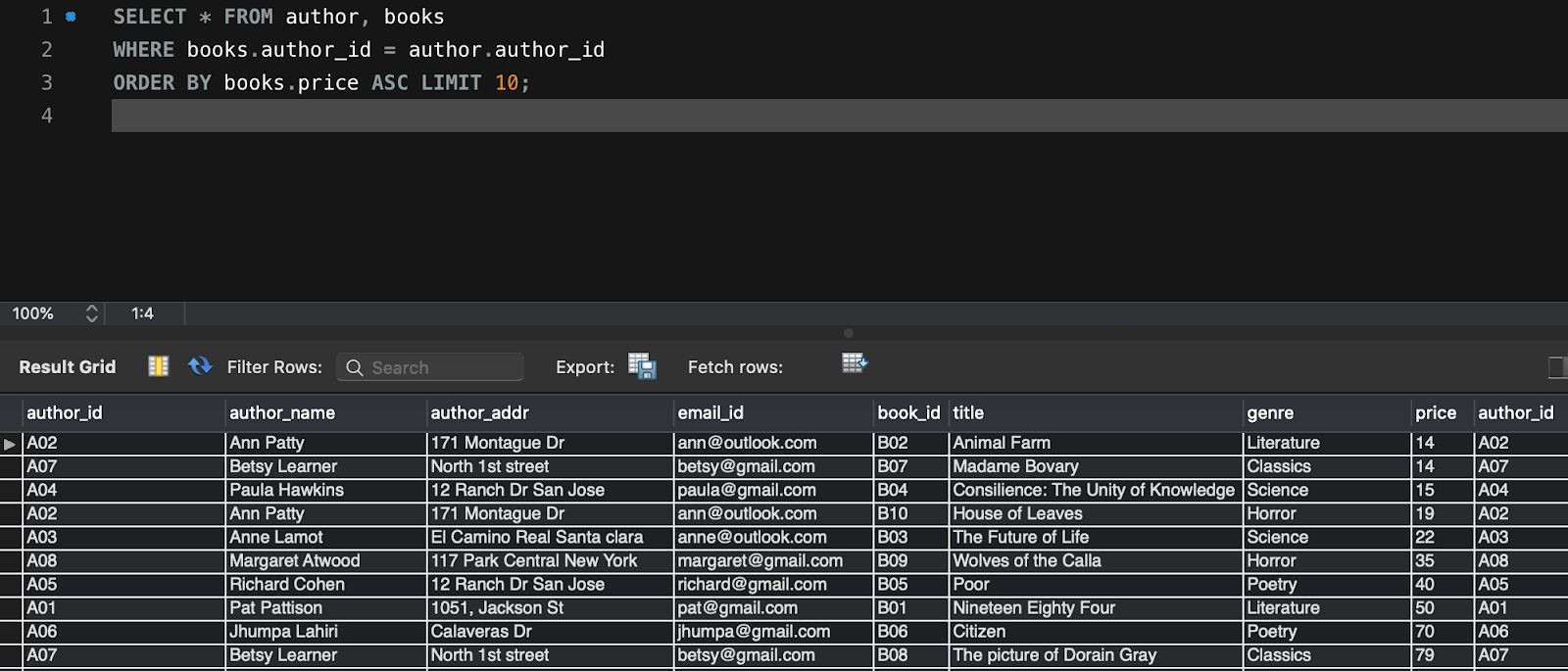
B.title = 'Wolves of the Calla' and A.author\_name = 'Margaret Atwood'; (1)



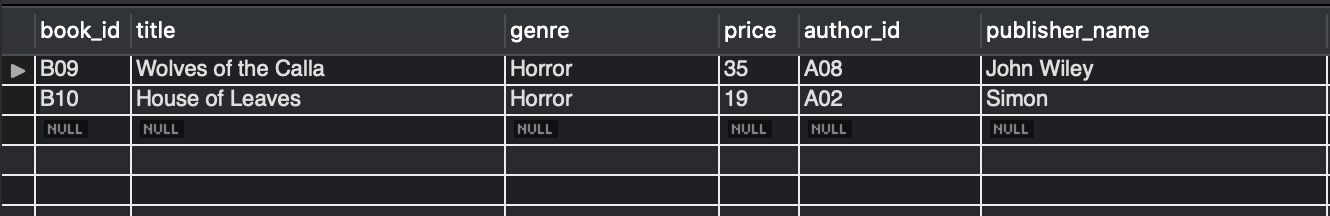
# Display top 10 cheapest ebooks

SELECT \*   
FROM author, books

WHERE books.author\_id = author.author\_id ORDER BY books.price ASC LIMIT 10;



# Select books based on genre

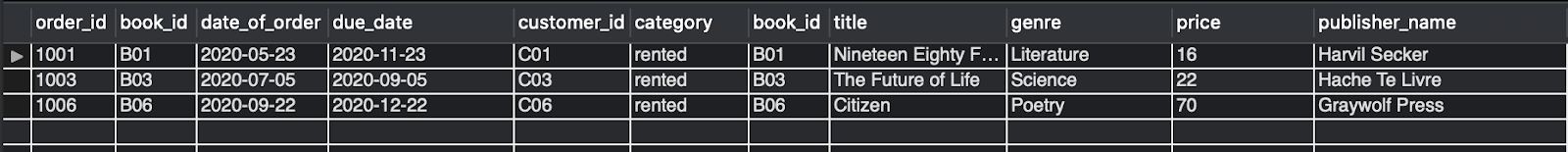
SELECT \* FROM BOOKS   
WHERE GENRE = 'Horror'; (2)

1. **JOIN**:

# Joining of two tables with all the columns being displayed

SELECT \* FROM orders JOIN rentals

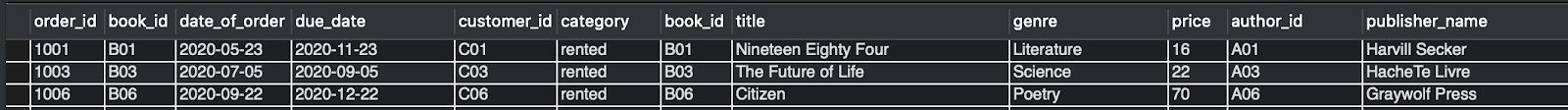
ON orders.book\_id = rentals.book\_id;



# joining orders and books table where category = rented

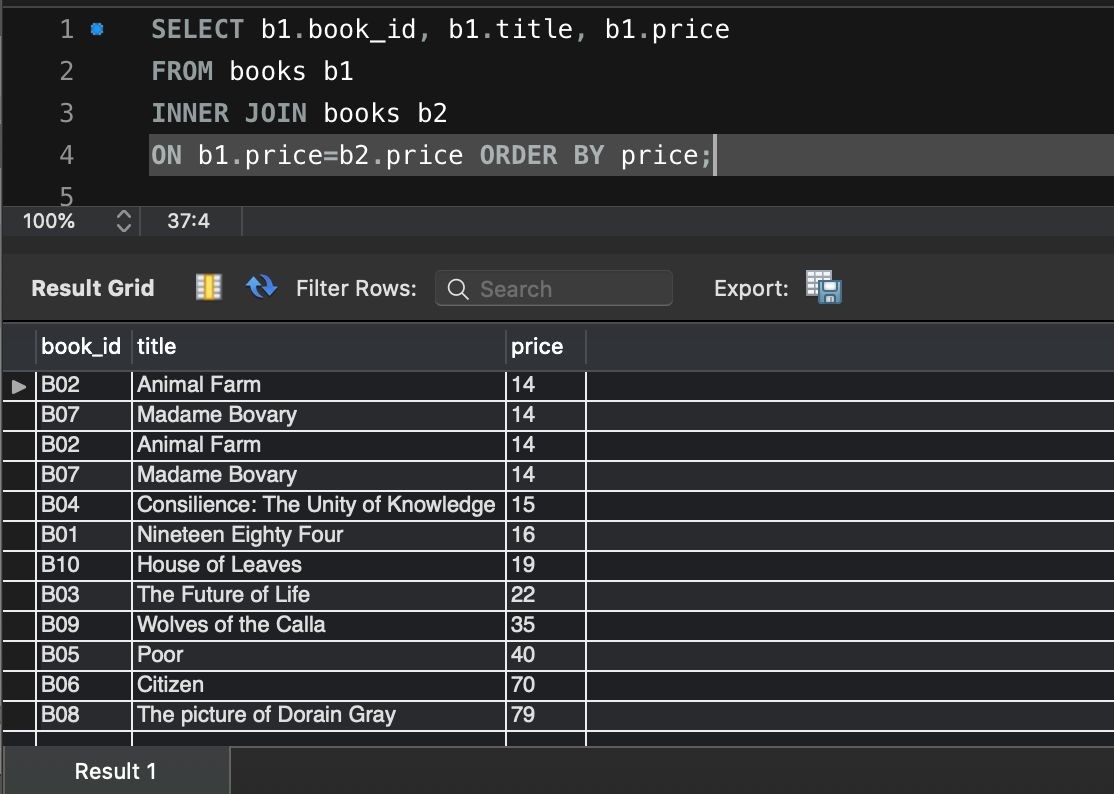
SELECT \* FROM orders O  
 JOIN books B

ON O.book\_id = B.book\_id   
WHERE O.category = 'rented';



1. SELF JOIN:
   1. SELECT b1.book\_id, b1.title, b1.price   
      FROM books b1

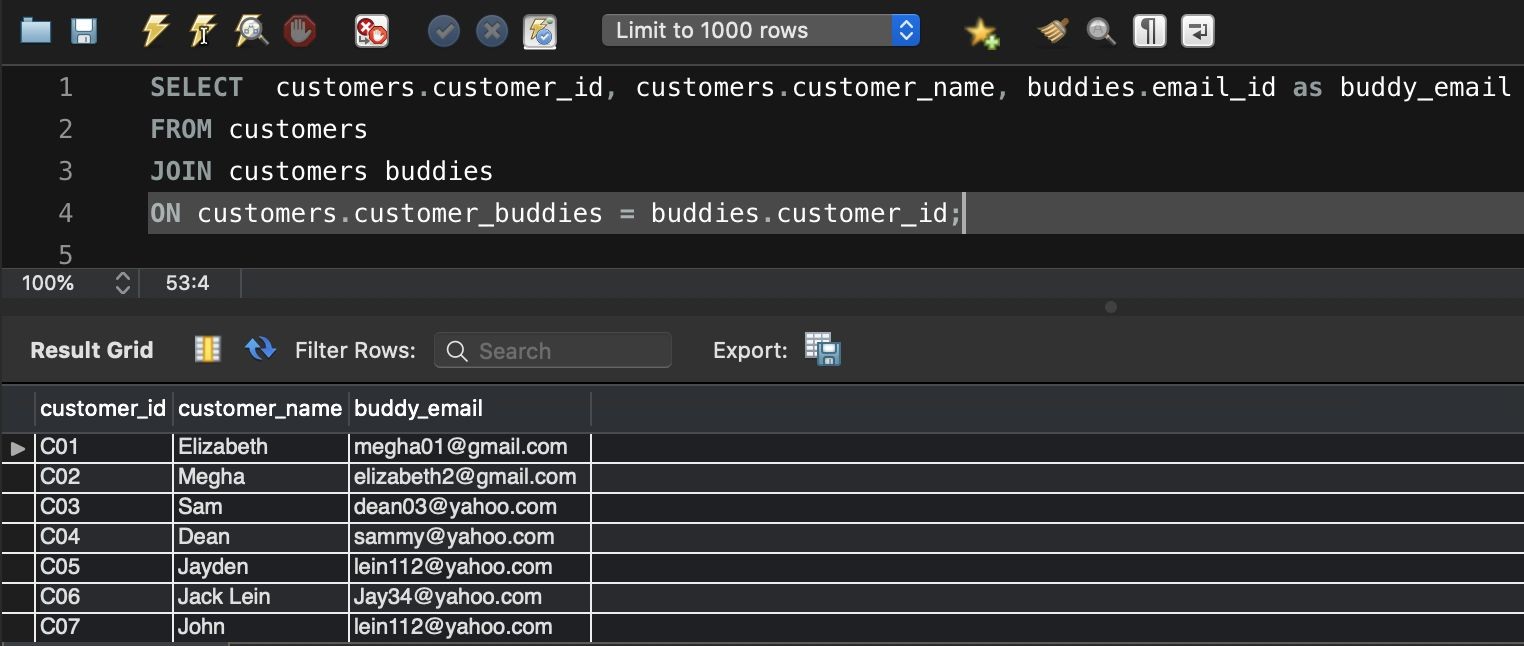
INNER JOIN books b2 ON b1.price = b2.price   
ORDER BY price;



* 1. SELECT customers.customer\_id, customers.customer\_name, buddies.email\_id as buddy\_email FROM customers

JOIN customers buddies

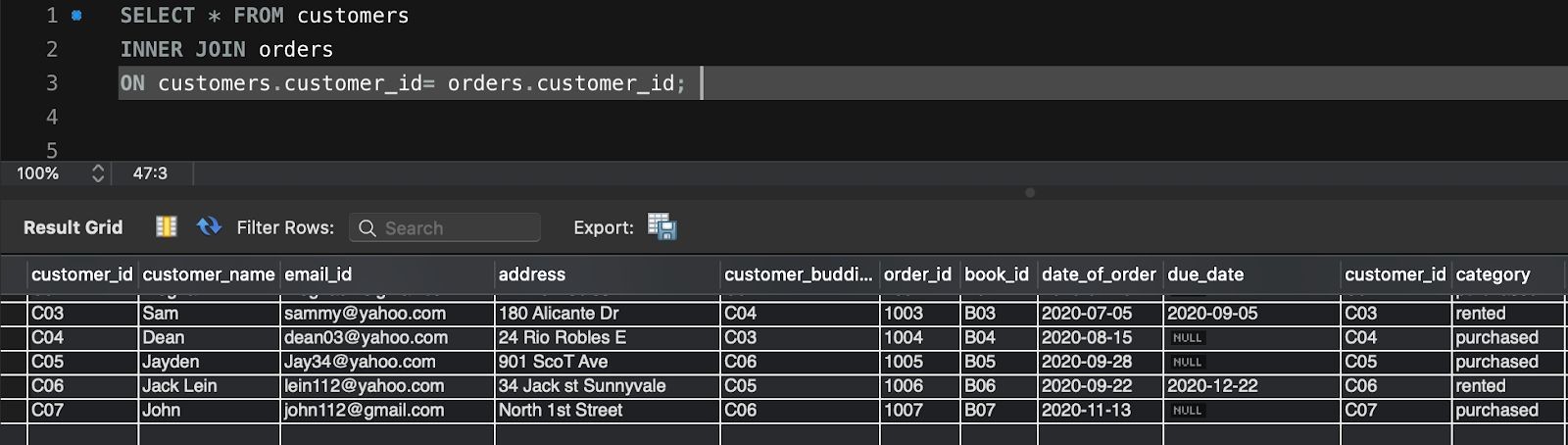
ON customers.CUSTOMER\_BUDDY = buddies.customer\_id;



1. INNER JOIN:
   1. Join customer and order table WITH SELECTED COLUMNS DISPLAYED SELECT \* FROM CUSTOMERS C

INNER JOIN ORDERS O

ON C.CUSTOMER\_ID = O.CUSTOMER\_ID; (8)

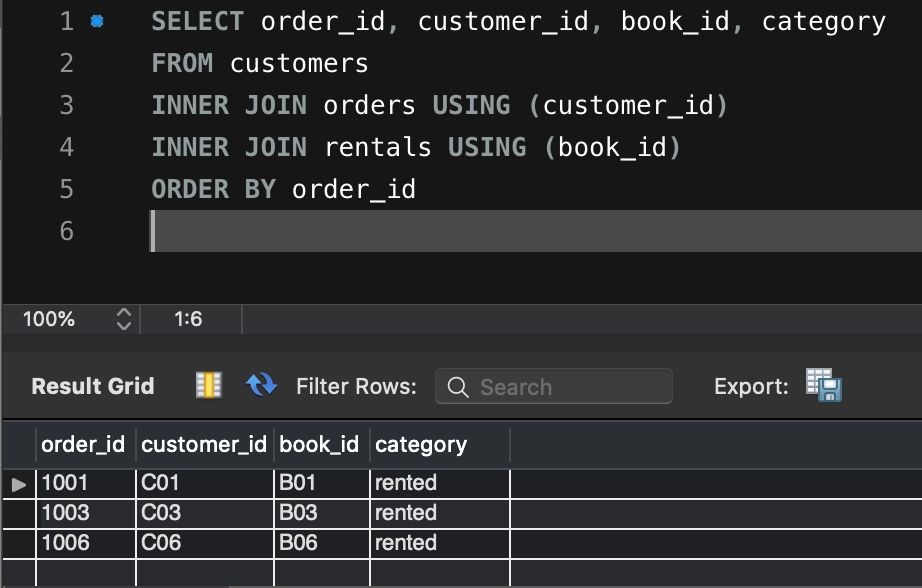


1. MULTI JOIN:
   1. Multiple join

SELECT order\_id, customer\_id, book\_id, category FROM customers

INNER JOIN orders USING (customer\_id)

INNER JOIN rentals USING (book\_id) ORDER BY order\_id;



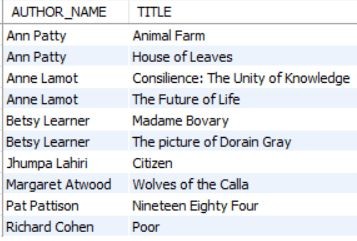
1. GROUP BY:

# group by author\_name

SELECT A.AUTHOR\_NAME, TITLE   
FROM BOOKS B, AUTHOR A   
WHERE B.AUTHOR\_ID = A.AUTHOR\_ID

GROUP BY A.AUTHOR\_NAME, B.TITLE

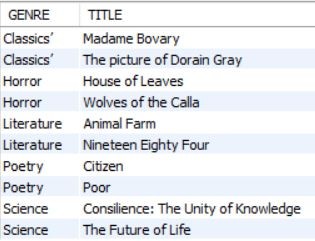
ORDER BY A.AUTHOR\_NAME;



# group titles by genre

SELECT GENRE, TITLE FROM BOOKS GROUP BY GENRE, TITLE

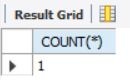
ORDER BY GENRE; (10)



1. COUNT:

# count of e-books ordered based on author name

SELECT COUNT(\*) FROM ORDERS O, AUTHOR A, BOOKS B WHERE O.BOOK\_ID = B.BOOK\_ID

AND B.AUTHOR\_ID = A.AUTHOR\_ID AND A.AUTHOR\_NAME = 'Ann Patty'; (1)

1. INSERT, UPDATE, DELETE:

# insert a row into books

INSERT INTO BOOKS VALUES ('B11','The Testaments','Suspense','15','A08','Nan A. Talese');

# update address of a customer

UPDATE CUSTOMERS

SET ADDRESS = '901 Scott Ave' WHERE ADDRESS = '901 ScoT Ave';

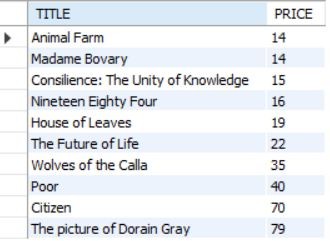
# delete row from books based on title

DELETE FROM BOOKS WHERE title = 'The Testaments'

1. ORDER BY:

# sort titles by price

SELECT TITLE, PRICE FROM BOOKS ORDER BY PRICE; (10)



# displays the author name and title of books with highest price

SELECT AUTHOR\_NAME, TITLE FROM AUTHOR A, BOOKS B WHERE B.AUTHOR\_ID = A.AUTHOR\_ID

AND ROWNUM <=1

ORDER BY B.PRICE ASC;



1. AGGREGATE :
   1. Displays counts of records in author and customers table

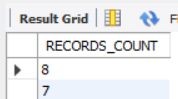
SELECT RECORDS\_COUNT.RECORDS\_COUNT FROM

(

SELECT COUNT(\*) AS RECORDS\_COUNT FROM AUTHOR UNION ALL

SELECT COUNT(\*) AS RECORDS\_COUNT FROM CUSTOMERS

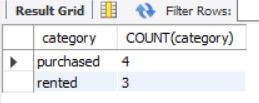
)RECORDS\_COUNT; (2)



# display number of e-books rented vs. purchased

SELECT category, COUNT(category) FROM orders GROUP BY category

ORDER BY category; (2)



# display total price of the order

SELECT SUM(PRICE) AS TOTAL\_PRICE FROM BOOKS B, ORDERS O WHERE B.BOOK\_ID = O.BOOK\_ID

AND ORDER\_ID = '1001';



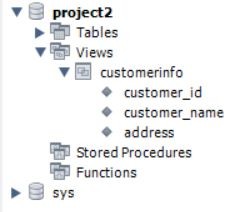
`11) VIEW:

1. Creating the view

CREATE VIEW customerinfo AS

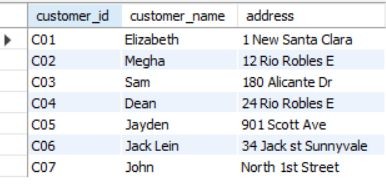
SELECT customer\_id, customer\_name, address

FROM customers;



1. Displaying the view

SELECT \* FROM customerinfo;



1. Updating the view

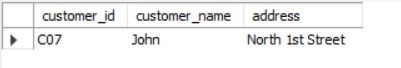
UPDATE customerinfo SET

address = 'North 1st Street' WHERE

customer\_id= 'C07';

1. Verify the updated view

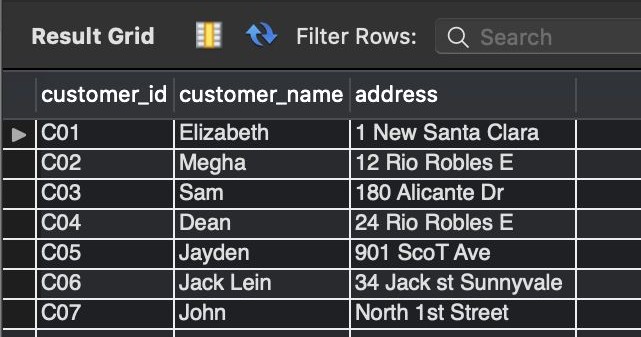
SELECT \* FROM customerinfo WHERE customer\_id='C07';



1. Delete a row from view

DELETE FROM customerinfo WHERE

customer\_id = 'C08';



1. ALTER TABLE :
   1. Add new column to Customer table

ALTER TABLE CUSTOMERS

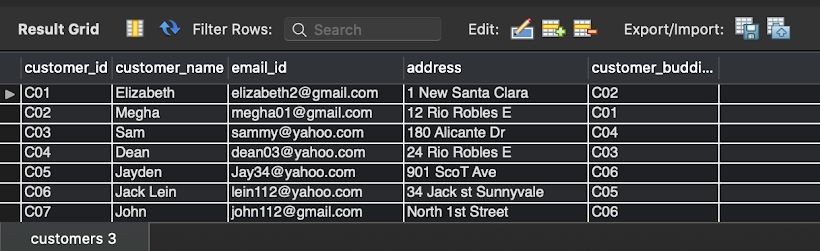
ADD COLUMN CUSTOMER\_BUDDY VARCHAR(20) ;

* 1. Delete column from customer table

ALTER TABLE customers DROP COLUMN CUSTOMER\_BUDDY;

1. Insert values into customer table newly added column :

update customers set customer\_buddy ='C01' where customer\_id = 'C02' ; update customers set customer\_buddy ='C02' where customer\_id = 'C01' ; update customers set customer\_buddy ='C03' where customer\_id = 'C04' ; update customers set customer\_buddy ='C04' where customer\_id = 'C03' ; update customers set customer\_buddy ='C05' where customer\_id = 'C06' ; update customers set customer\_buddy ='C06' where customer\_id = 'C05' ; update customers set customer\_buddy ='C06' where customer\_id = 'C07' ;



**Transactions:**

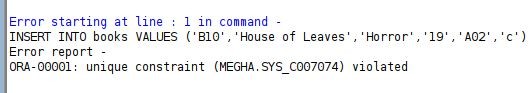
**Constraint Violation:**

1. **Primary Key Constraint Violation:**

INSERT INTO books VALUES ('B10','House of Leaves','Horror','19','A02','c');

Insert again with same book id

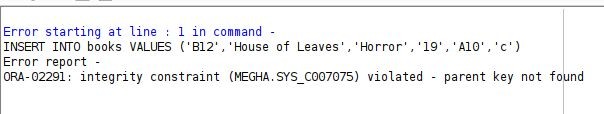
INSERT INTO books VALUES ('B10','House of Leaves','Horror','19','A02','c');



1. **Foreign Key Constraint**

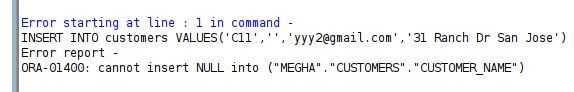
INSERT INTO books VALUES ('B12','House of Leaves','Horror','19','A10','c');

Author does not exist in Author table



1. **Not null constraint:**

INSERT INTO customers VALUES('C11','','yyy2@gmail.com','31 Ranch Dr San Jose');

Customer name cannot be null.

**Trigger:**

1) On update of book price, update price of book in rentals table to half of the original price

CREATE or replace TRIGGER rental\_book\_update AFTER UPDATE of price on books

FOR EACH ROW

BEGIN

UPDATE rentals

set rentals.price=(:new.price/2) where rentals.book\_id=(:old.book\_id); END;

**Nested Query:**

1) Details of top three most expensive books

SELECT \* FROM

(SELECT b.title, b.genre, b.price, b.publisher\_name, a.author\_name FROM books b, author a

WHERE b.author\_id=a.author\_id ORDER BY b.price desc)

where ROWNUM<4;

