

```

SELECT FirstName,
       LastName,
       Email
FROM Customers
ORDER BY LastName; -
- Create Table to store InformationCREATE TABLE AnniversaryAttendees (
    [CUSTOMER ID] INT,
    [Party Size] INT
); -- Create MenuSELECT *
FROM DISHES
ORDER BY PRICE;SELECT *
FROM Dishes
WHERE Type = "Appetizer" OR
       Type = "Beverage"
ORDER BY PRICE; -
- Sign a customer for loyalty ProgramBEGIN TRANSACTION;INSERT INTO Customers (
    FirstName,
    LastName,
    Email,
    Address,
    City,
    State,
    Phone,
    Birthday
)
VALUES (
    "Eccel",
    "Gunawan",
    "eccelwei@gmail.com",
    "jalan Jalan",
    "Memphis",
    "TN",
    "+62910293829",
    "1938-12-12"
);END TRANSACTION;SELECT *
FROM Customers; -- Update cust. InformationBEGIN TRANSACTION;UPDATE Customers
SET address = 'SALAHBRO'
WHERE CustomerID = 101;END TRANSACTION; -
- Remove cust. record BEGIN TRANSACTION;DELETE FROM Customers
WHERE CustomerID = 101;END TRANSACTION; -
- Log cust responses INSERT INTO AnniversaryAttendees (
    [CUSTOMER ID],
    [Party Size]
)
VALUES (
    (
        SELECT CustomerID
        FROM Customers
        WHERE Email = "smatticci2c@rouxacademy.o
rg"
    ),
    "4"
);SELECT *
FROM AnniversaryAttendees; -- Look reservationSELECT Customers.FirstName,
Customers.LastName,
Reservations.Date,
Reservations.PartySize
FROM Reservations
JOIN
Customers ON Customers.CustomerID = Reservations.CustomerID
WHERE Customers.LastName LIKE "STE%"; -- Take reservationSELECT *
FROM Customers;BEGIN TRANSACTION;INSERT INTO Customers (
    FirstName,
    LastName,
    Email,
    Phone
)

```

```

VALUES (
    "SAM",
    "McAdams",
    "smac@rouxacademy.com",
    "(555)555-1212"
);END TRANSACTION ;BEGIN TRANSACTION;INSERT INTO Reservation
s (
    CustomerID,
    Date,
    PartySize
)
VALUES (
    "102",
    "2020-06-14 18:00:00",
    '5'
);END TRANSACTION ;SELECT *
FROM Reservations
WHERE CustomerID = "102";SELECT *
FROM Orders
WHERE CustomerID = "70";BEGIN TRANSACTION;INSERT INTO Orders (
    CustomerID,
    OrderDate
)
VALUES (
    70,
    "2020-03-20 14:00:00"
);END TRANSACTION ;-- Track Favorite Dishes SELECT DishID
FROM Dishes
WHERE Name = "Quinoa Salmon Salad";SELECT *
FROM Customers
WHERE FirstName = "Cleo";UPDATE Customers
SET FavoriteDish = "9"
WHERE CustomerID = "42";-- Check book availabilitySELECT COUNT(Books.Title)
FROM Loans
JOIN
BOOKS ON Loans.BookID = Books.BookID
WHERE Books.Title = "Dracula" AND
Loans.ReturnedDate IS NULL;-- Add new bookSELECT *
FROM Books;BEGIN TRANSACTION;INSERT INTO Books (
    Title,
    Author,
    Published,
    BarCode
)
VALUES (
    "Dracula",
    "Bram Stoker",
    "1897",
    "4819277482"
),
(
    " Gulliver's Travels",
    "Jonathan Swift",
    "1729",
    "4899254401"
);END TRANSACTION ;-- Check book due INSERT INTO Loans (
    BookID,
    PatronID,
    LoanDate,
    DueDate
)
VALUES (
    (
        SELECT BookID
        FROM Books
        WHERE Barcode = "2855934983"
    ),

```

```

        (
            SELECT PatronID
            FROM Patrons
            WHERE Email = "jvaan@wisdompets.com"
        ),
        "2020-08-25",
        "2020-09-08"
    ); -- Return book
SELECT Loans.DueDate,
Books.Title,
Patrons.Email,
Patrons.FirstName
FROM Loans
JOIN
Books ON Loans.BookID = Books.BookID
JOIN
Patrons ON Loans.PatronID = Patrons.PatronID
WHERE Loans.DueDate = "2020-07-13" AND
Loans.ReturnedDate IS NULL;
UPDATE Loans
SET ReturnedDate = "2020-07-05"
WHERE BookID = (
    SELECT BookID
    FROM Books
    WHERE Barcode = "8730298424"
)
AND
ReturnedDate IS NULL;
SELECT COUNT(Loans.LoanID) AS loancount,
Patrons.FirstName,
Patrons.Email
FROM Loans
JOIN
Patrons ON Loans.PatronID = Patrons.PatronID
GROUP BY Loans.PatronID
LIMIT 10; -- Book to feature
SELECT Books.Title,
Books.Author,
Books.Published
FROM Books
JOIN
Loans ON Books.BookID = Loans.BookID
WHERE Books.Published > 1889 AND
Books.Published < 1900 AND
LOANS.ReturnedDate IS NOT NULL
GROUP BY Books.BookID
ORDER BY Books.Title; -- Book statistics
SELECT Published,
COUNT(DISTINCT (Title) )
FROM Books
GROUP BY Published;

```