Bank

Project Members

Joe Cruz, 50272651

Colton Van Voorhis, 50299054

Group ID: 5

Step 1: Create an imaginary scenario. Your scenario should satisfy following conditions:

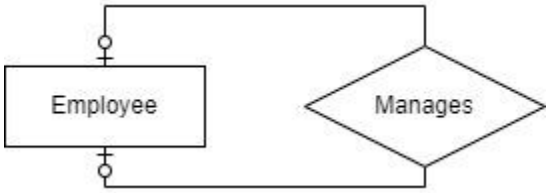
The Sigma Bank Co is headquartered in Commerce, Texas and has multiple branches throughout the USA. Each branch oversees its own customers and the employees at each branch location. A customer can have accounts and their own loans.

a. Include at least two one-to-many binary relationships.

b. Include at least two many-to-many binary relationships.

c. A valid scenario should include at least one intersection data (on many-to-many relationships).

d. A valid scenario should include at least one one-to-one unary relationships.



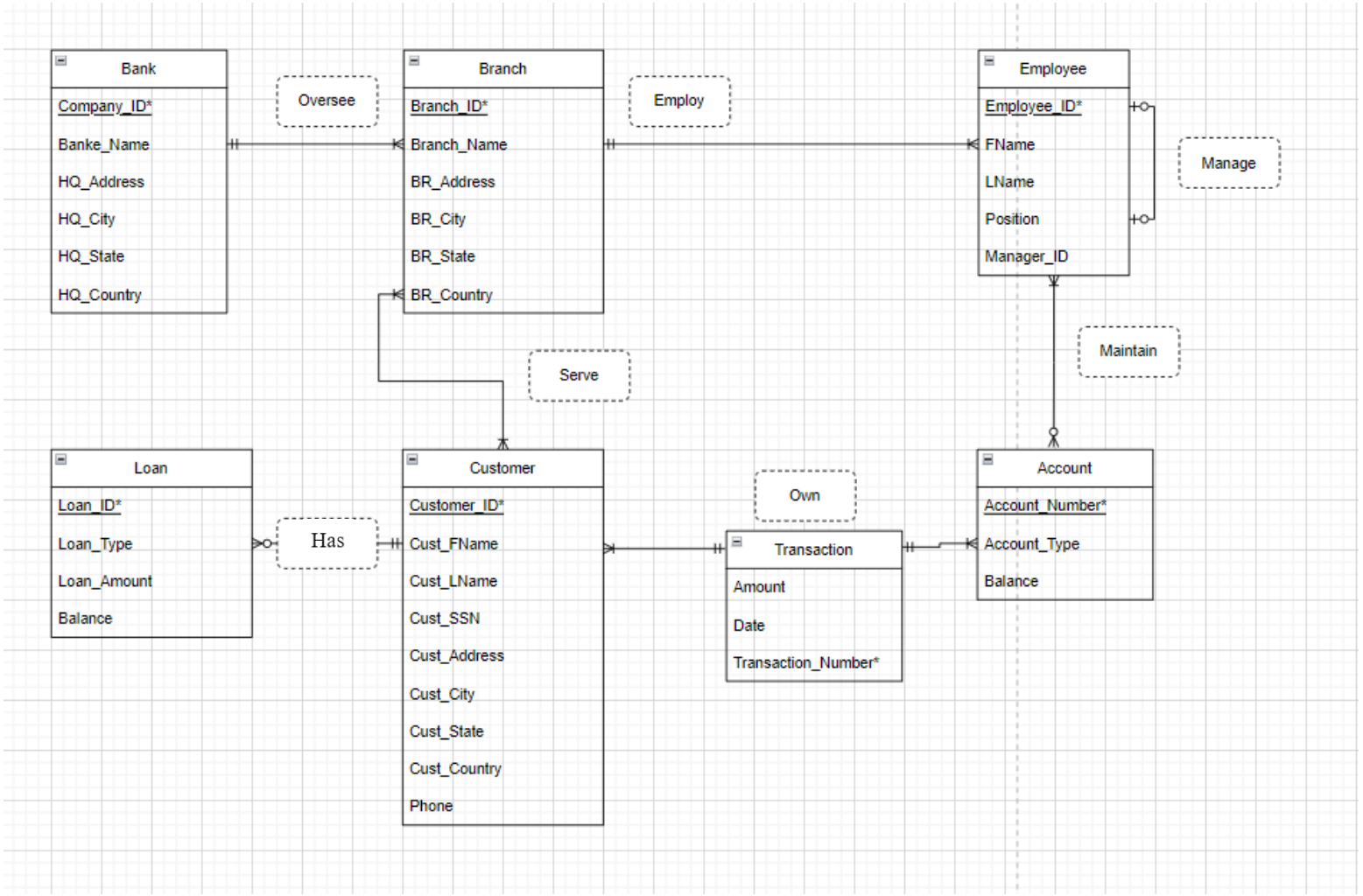
Step 2: Explain the story behind the scenario, and all your assumptions, which are required to support relationships given above.

The Sigma Bank Co is headquartered in Sigmaville, Texas and has multiple branches throughout the country. Each branch oversees its own customers and the employees at every branch location. Customers can bank at multiple branches while employees belong exclusively to a single branch location. A customer can also have multiple accounts and multiple loans, with joint ownership of accounts being possible. In order to change the amount in their account employees must complete transactions (intersection data). To be a customer, a person must have at least one account, but they aren't required to have any active loans. Employees are responsible for managing accounts assigned to them and bank managers organize which employees are responsible for what accounts. Branch managers and employees come in pairs.

Step 3: Create a table that shows the the entities and their relationship in relation to their attributes

Entity	**Attributes**
Bank	Bank_Name, Bank_ID, HQ_Address, HQ_City, HQ_State, HQ_Country
Branch	Branch_ID, Branch_Name, BR_Address, BR_City, BR_State, BR_Country
Customer	Customer_ID, Cust_FName, CustL_LName, SSN, Cust_Address, Cust_City, Cust_State, Cust_Country, Phone
Account	Account_Number, Account_Type, Account_Balance
Loan	Loan_ID, Loan_Type, Loan_Amount, Balance
Employee	Employee_ID, FName, LName, Position, Manager_ID
Transaction	Amount, Date, Transaction_Number, Account_number

Step 4: Show the Entity Relationship Diagram



Side note: The "Balance" is how much money the customer has left to pay back for the loan.

Step 5: Display the primary keys and foreign key relations.

FK/PK Pairs

Loan Customer_ID (FK) → Customer Customer_ID (PK)

Account Customer_ID (FK) → Customer Customer_ID (PK)

Transaction Customer_ID (FK) → Customer Customer_ID (PK)

Transaction Account_number (FK) → Account Account_number(PK)

Employee Manager_ID (FK) → Employee Employee_ID (PK)

Employee Branch_ID (FK) → Branch Branch_ID (PK)

Customer Branch_ID (FK) → Branch Branch_ID (PK)

Step 6: Display the entities within the database.

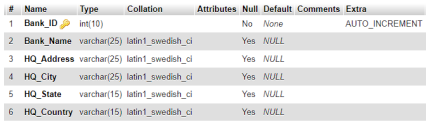
Account

Account_Number	Account_Type	Account_Balance	Customer_ID
1	Savings	3.92	1
2	Savings	420.69	3
3	Checking	67.99	4
4	Savings	999999.99	6
5	Checking	23.84	7
6	Savings	787634.23	10
7	Checking	23400.46	11
8	Checking	2382.19	16
9	Checking	455.66	17
10	Savings	734984.91	18

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Account_Number	int(10)			No	None		AUTO_INCREMENT
2	Account_Type	varchar(25)	latin1_swedish_ci		Yes	NULL		
3	Account_Balance	double(99,2)			Yes	NULL		
4	Customer_ID	int(11)			No	None		

Bank:

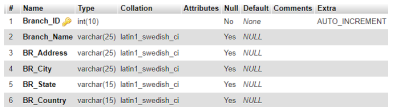
Bank_ID	Bank_Name	HQ_Address	HQ_City	HQ_State	HQ_Country
1	Sigma Bank Co.	123 Drivelane	Sigmaville	Texas	USA



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Bank_ID	int(10)			No	None		AUTO_INCREMENT
2	Bank_Name	varchar(25)	latin1_swedish_ci		Yes	NULL		
3	HQ_Address	varchar(25)	latin1_swedish_ci		Yes	NULL		
4	HQ_City	varchar(25)	latin1_swedish_ci		Yes	NULL		
5	HQ_State	varchar(15)	latin1_swedish_ci		Yes	NULL		
6	HQ_Country	varchar(15)	latin1_swedish_ci		Yes	NULL		

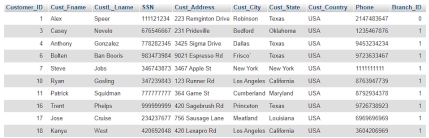
Branch:

Branch_ID	Branch_Name	BR_Address	BR_City	BR_State	BR_Country
1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA
2	Sigma Bank	578 Brown Sugar Dr	Dallas	Texas	USA
3	Beta Bank	123 Ram Rd	Cumberland	Maryland	USA
4	Lambda Bank	546 Long Ln	New York	New York	USA
5	Zeta Bank	547 Feet Av	Omega	Georgia	USA
6	Gamma Bank	378 Meatland Av	Sausage	Louisiana	USA
7	Theta Bank	195 Oak Tree	Bedford	Oklahoma	USA
8	Omicron Bank	876 Smith St	Los Angeles	California	USA
9	Chi Bank	765 Schlong Av	Frisco	Texas	USA
10	Ligma Bank	356 Perry St	Commerce	Texas	USA

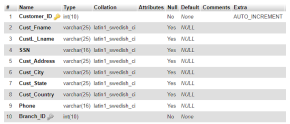


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Branch_ID	int(10)			No	None		AUTO_INCREMENT
2	Branch_Name	varchar(25)	latin1_swedish_ci		Yes	NULL		
3	BR_Address	varchar(25)	latin1_swedish_ci		Yes	NULL		
4	BR_City	varchar(25)	latin1_swedish_ci		Yes	NULL		
5	BR_State	varchar(15)	latin1_swedish_ci		Yes	NULL		
6	BR_Country	varchar(15)	latin1_swedish_ci		Yes	NULL		

Customer:



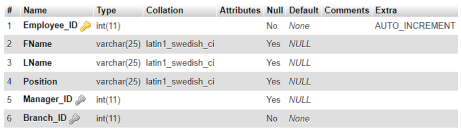
Customer_ID	Cust_Fname	Cust_Lname	SSN	Cust_Address	Cust_City	Cust_State	Cust_Country	Phone	Branch_ID
1	Alex	Speer	111121234	223 Remington Drive	Robinson	Texas	USA	2147483647	8
3	Casey	Nevelo	676546667	231 Prideville	Bedford	Oklahoma	USA	1235467876	1
4	Anthony	Gonzalez	778282345	3425 Sigma Drive	Dallas	Texas	USA	9453234234	1
6	Bolton	Ben Boorlo	983473964	9021 Espresso Rd	Frisco	Texas	USA	9723633467	1
7	Steve	Jobs	346743873	3467 Apple St	New York	New York	USA	1111111111	1
10	Ryan	Gosling	347239843	123 Runner Rd	Los Angeles	California	USA	8763947739	1
11	Patrick	Squidman	777777777	364 Game St	Cumberland	Maryland	USA	8792934378	1
16	Trent	Phelps	999999999	420 Sagebrush Rd	Princeton	Texas	USA	9726738923	1
17	Jose	Cruise	234237677	756 Sausage Lane	Meatland	Louisiana	USA	6969696969	1
18	Kanye	West	420692048	420 Lunapro Rd	Los Angeles	California	USA	3634200969	1



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Customer_ID	int(10)			No	None		AUTO_INCREMENT
2	Cust_Fname	varchar(25)	latin1_swedish_ci		Yes	NULL		
3	Cust_Lname	varchar(25)	latin1_swedish_ci		Yes	NULL		
4	SSN	varchar(16)	latin1_swedish_ci		Yes	NULL		
5	Cust_Address	varchar(25)	latin1_swedish_ci		Yes	NULL		
6	Cust_City	varchar(25)	latin1_swedish_ci		Yes	NULL		
7	Cust_State	varchar(25)	latin1_swedish_ci		Yes	NULL		
8	Cust_Country	varchar(25)	latin1_swedish_ci		Yes	NULL		
9	Phone	varchar(16)	latin1_swedish_ci		Yes	NULL		
10	Branch_ID	int(10)			No	None		

Employee:

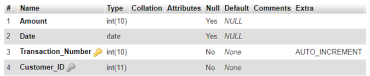
Employee_ID	FName	LName	Position	Manager_ID	Branch_ID
5	Lou	Bloom	Branch Manager	NULL	1
6	Rick	Ahmed	Teller	5	1
7	Park	Squiga	Security	5	1
12	Paul	Allen	Branch Manager	NULL	3
13	Blake	Barnes	Teller	12	3
14	Lukas	Morehead	Security	12	3
15	Kraven	Morehead	Branch Manager	NULL	9
16	Edith	Branch	Teller	15	9
17	Max	Bomba	Security	15	9
18	Yale	Du	Branch Manager	NULL	6
19	Cavender	Campbell	Teller	18	6
20	Carrie	Butts	Security	18	6
21	Gina	Fouke	Branch Manager	NULL	4
22	James	Evans	Teller	21	4
23	Timothy	Stone	Security	21	4
24	Morris	Chambers	Branch Manager	NULL	10
25	Ana	Espino	Teller	24	10
26	Kayla	Reagan	Security	24	10
27	Adrian	Fowler	Branch Manager	NULL	8
28	Donald	Glover	Teller	27	8
29	Doctor	Morbius	Security	27	8
30	Steven	Strange	Branch Manager	NULL	2
31	Tony	Stark	Teller	30	2
32	Mental	Breakdown	Security	30	2
33	Shack	Wes	Branch Manager	NULL	7



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Employee_ID	int(11)			No	None		AUTO_INCREMENT
2	FName	varchar(25)	latin1_swedish_ci		Yes	NULL		
3	LName	varchar(25)	latin1_swedish_ci		Yes	NULL		
4	Position	varchar(25)	latin1_swedish_ci		Yes	NULL		
5	Manager_ID	int(11)			Yes	NULL		
6	Branch_ID	int(11)			No	None		

Loan:

Loan_ID	Loan_Type	Loan_Amount	Balance	Customer_ID
1	Student	8600	4000	1
2	Personal	8000	3400	3
3	Auto	2000	400	4
4	Home-Equity	3700	230	16
5	Credit-Builder	300	80	7
6	Personal	8000	6000	10
7	Student	25000	24000	11
8	Auto	1200	500	16
9	Mortgage	3500	1600	18
10	Home-Equity	8500	3259	18



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Amount	int(10)			Yes	NULL		
2	Date	date			Yes	NULL		
3	Transaction_Number	int(10)			No	None		AUTO_INCREMENT
4	Customer_ID	int(11)			No	None		

Transaction:

Amount	Date	Transaction_Number	Customer_ID	Account_Number
-500.00	2021-12-27	1	18	10
23.00	2021-10-13	29	10	6
2173.00	2022-05-01	30	7	5
6409.42	2022-02-28	31	1	1
1367.00	2022-01-06	32	10	6
6927.00	2021-03-16	33	10	6
-9.00	2021-05-19	34	3	2
82.95	2022-03-24	35	7	5
49.00	2022-02-15	36	10	6
-2.00	2021-12-14	37	16	8

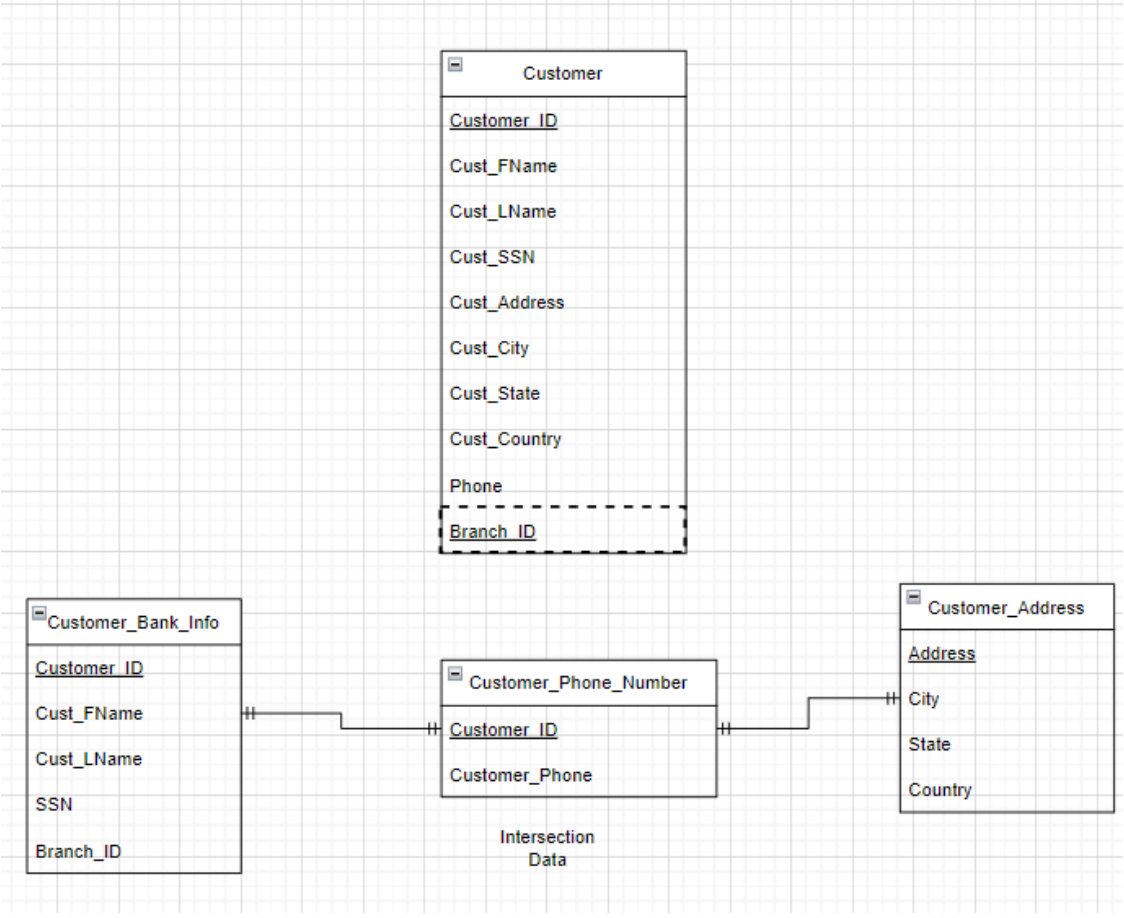
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Amount	double(99,2)			Yes	NULL		
2	Date	date			Yes	NULL		
3	Transaction_Number	int(10)			No	None		AUTO_INCREMENT
4	Customer_ID	int(11)			No	None		
5	Account_Number	int(10)			No	None		

Step 7:

1NF - All of the tables have the same number of atomic attributes. The transaction table is an example of a data set that emerged from multiple overlaps between Customer and Account.

2NF - Most tables obey 2NF rules with non-key elements depending solely on their associated primary key.

3NF - Most tables do obey 3NF rules except for the Customer table. The reason we only normalize the Customer table is because multiple customers could live at the same address. The reason we don't have to normalize the Bank and Branch tables into specific address tables is because no other building would be in the same address as the Bank/Branch building.



Step 8: Display the SQL statement for each attribute.

BANK(Bank_Name (PK), Company_ID, HQ_Address, HQ_City, HQ_State, HQ_Country)

BRANCH(Branch_ID (PK), Branch_Name, BR_Address, BR_City, BR_State, BR_Country)

CUSTOMER(Customer_ID (PK), Cust_FName, CustL_LName, Cust_SSN, Cust_Address, Cust_City, Cust_State, Cust_Coutry, Phone, Branch_ID (FK))

LOAN(Loan_ID (PK), Loan_Type, Loan_Amount, Balance, Customer_ID (FK))

ACCOUNT(Account_Number (PK), Account_Type, Balance, Customer_ID (FK))

EMPLOYEE(Employee_ID (PK), FName, LName, Position, Manager_ID (FK), Branch_ID(FK))

TRANSACTION(Transaction_Number(PK), Amount, Date, Customer_ID(FK), Branch_ID(FK))

Step 9: Display the use of the INSERT statement.

INSERT INTO `Bank` (`Bank_ID`, `Bank_Name`, `HQ_Address`, `HQ_City`, `HQ_State`, `HQ_Country`) VALUES ('1', 'Sigma Bank Co.', '123 Drivelane', 'Sigmaville', 'Texas', 'USA')

INSERT INTO `Customer` (`Customer_ID`, `Cust_Fname`, `CustL_Lname`, `SSN`, `Cust_Address`, `Cust_City`, `Cust_State`, `Cust_Country`, `Phone`) VALUES ('1', 'Alex', 'Speer', '111121234', '223 Remginton Drive', 'Robinson', 'Texas', 'USA', '2541236452')

Step 10: Display the use of the DELETE statement.

```
DELETE FROM `Employee` WHERE  ‘Employee_ID’ = 1

DELETE FROM `Customer` WHERE `Customer_ID` = 1
```

Step 11: Display the use of the UPDATE statement.

```
UPDATE `Branch` SET Branch_Name = 'Alpha Land', BR_City= 'Hillsboro' WHERE BranchID = 1;


UPDATE `Customer` SET Cust_City= 'Houston’, Phone= '1235431678’ WHERE BranchID = 1;
```

Step 12: Display the use of the SELECT statement.

Get the name and customer ID of customers who live in Texas.

SELECT Cust_Fname as First_Name, CustL_Lname as Last_Name, Cust_State as Home_State, Customer_ID as Customer_ID FROM `Customer` WHERE Cust_State = 'Texas'

http://joe-cruz.infinityfreeapp.com/datawhere.php?i=1

```
<?

@$db = mysql_pconnect("sql210.epizy.com", "epiz_31407029", "N94YMcLv6u5Q");

if (!$db) {
    echo "ERROR: Could not connect to database.  Please try again later.";
    exit;
}

mysql_select_db("epiz_31407029_sigma");

$query = "SELECT Cust_Fname as First_Name, CustL_Lname as Last_Name, Cust_State as Home_State, Customer_ID as Customer_ID FROM `Customer` WHERE Cust_State = 'Texas'";

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

echo "<p>Number of Customers found: " . $num_results . "</p>";

for ($i = 0; $i < $num_results; $i++) {
    $row = mysql_fetch_array($result);
    echo "
                <tr class='data'>
                    <td>" . $row['Customer_ID'] . "</td>
                    <td>" . $row['First_Name'] . " " . $row['Last_Name'] . "</td>
                    <td>" . $row['Home_State'] . "</td>
                </tr>
            ";
}

?>
```

Get the name, employee_ID, and branch_ID of employees who hold the position branch manager.

SELECT FName as First_Name, LName as Last_Name, Employee_ID, Branch_ID FROM `Employee` WHERE Position = 'Branch Manager' ORDER BY Branch_ID

http://joe-cruz.infinityfreeapp.com/datawhere2.php

```
<?

@$db = mysql_pconnect("sql210.epizy.com", "epiz_31407029", "N94YMcLv6u5Q");

if (!$db) {
    echo "ERROR: Could not connect to database.  Please try again later.";
    exit;
}

mysql_select_db("epiz_31407029_sigma");

$query = "SELECT FName as First_Name, LName as Last_Name, Employee_ID, Branch_ID FROM `Employee` WHERE Position = 'Branch Manager' ORDER BY Branch_ID";

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

echo "<p>Number of Customers found: " . $num_results . "</p>";

for ($i = 0; $i < $num_results; $i++) {
    $row = mysql_fetch_array($result);
    echo "
                <tr class='data'>
                    <td>" . $row['First_Name'] . " " . $row['Last_Name'] . "</td>
                    <td>" . $row['Employee_ID'] . "</td>
                    <td>" . $row['Branch_ID'] . "</td>
                </tr>
            ";
}

?>
```

Step 13: Display the use of the GROUP statement.

The entire branch table is grouped by branch name and shows the number of employees that work in each branch.

SELECT * FROM `Branch` group by Branch_Name

Branch_ID	Branch_Name	BR_Address	BR_City	BR_State	BR_Country	Number_of_Employees
1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA	3
3	Beta Bank	123 Ram Rd	Cumberland	Maryland	USA	3
9	Chi Bank	765 Schlong Av	Frisco	Texas	USA	3
6	Gamma Bank	378 Meatland Av	Sausage	Louisiana	USA	3
4	Lambda Bank	546 Long Ln	New York	New York	USA	3
10	Ligma Bank	356 Perry St	Commerce	Texas	USA	3
8	Omicron Bank	876 Smitih St	Los Angeles	California	USA	3
2	Sigma Bank	578 Brown Sugar Dr	Dallas	Texas	USA	3
7	Theta Bank	195 Oak Tree	Bedford	Oklahoma	USA	3
5	Zeta Bank	547 Feet Av	Omega	Georgia	USA	3

This table joins the Loan and Customer Table and then groups them by their Customer_ID. The purpose of this table is to show how many loans each customer has. It also sums up the total amount of money their loans are.

SELECT Customer.Cust_Fname, Customer.CustL_Lname, SUM(Loan.Loan_Amount) as Total_Loan_Amount, COUNT(Loan.Loan_ID) as Number_of_Loans FROM Loan LEFT JOIN Customer ON Loan.Customer_ID = Customer.Customer_ID GROUP by Customer.Customer_ID

Cust_Fname	CustL_Lname	Total_Loan_Amount	Number_of_Loans
Alex	Speer	8600	1
Casey	Novelo	8000	1
Anthony	Gonzalez	2000	1
Steve	Jobs	300	1
Ryan	Gosling	8000	1
Patrick	Squidman	25000	1
Trent	Phelps	4900	2
Kanye	West	12000	2

Step 14: Display the use of the “>” or “<” statement.

Get the customer_ID and remaining balance of customers with an outstanding loan balance greater than 1000.

SELECT Customer_ID, Balance from `Loan` GROUP BY Balance having Balance > 1000

Customer_ID	Balance
18	1600
18	3259
3	3400
1	4000
10	6000
11	24000

Get the name, customer_ID, and account balance of customers with an account balance less than 2000.

SELECT Customer.Cust_Fname as First_Name, Customer.CustL_Lname as Last_Name, Account.Customer_ID, Account.Account_Balance as Balance FROM Account LEFT JOIN Customer ON Account.Customer_ID = Customer.Customer_ID GROUP BY Account_Balance having Account_Balance < 2000

First_Name	Last_Name	Customer_ID	Balance
Alex	Speer	1	3.92
Steve	Jobs	7	23.84
Anthony	Gonzalez	4	67.99
Casey	Novelo	3	420.69
Jose	Cruise	17	455.66

Step 15:

This table joins the Branch table with the Customer table. The Branch table has Branch_ID as its primary key, and the Customer table uses the Branch_ID from the Branch table as a foregin key. The customer table's primary key is Customer_ID.

If I were to do a left join, then some of the customer columns would be empty because some of the branches don't have customers

SELECT Customer.Cust_Fname, Customer.CustL_Lname, Branch.* FROM Branch RIGHT JOIN Customer ON Branch.Branch_ID = Customer.Branch_ID;

Cust_Fname	CustL_Lname	Branch_ID	Branch_Name	BR_Address	BR_City	BR_State	BR_Country
Alex	Speer	2	Sigma Bank	578 Brown Sugar Dr	Dallas	Texas	USA
Casey	Novelo	1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA
Anthony	Gonzalez	1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA
Bolten	Ban Booris	1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA
Steve	Jobs	4	Lambda Bank	546 Long Ln	New York	New York	USA
Ryan	Gosling	8	Omicron Bank	876 Smitih St	Los Angeles	California	USA
Patrick	Squidman	1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA
Trent	Phelps	1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA
Jose	Cruise	7	Theta Bank	195 Oak Tree	Bedford	Oklahoma	USA
Kanye	West	1	Alpha Bank	125 Sugar Tree	Hillsboro	Texas	USA

Step 16:

Displays the total amount of loans given to a customer as well as who that customer is with the type of loan.

We left-join the Customer to the Loan table because a customer could possibly not have a loan, but a loan has to have a customer.

SELECT Customer.Cust_Fname, Customer.CustL_Lname, Loan.Loan_ID, Loan.Loan_Type FROM Loan LEFT JOIN Customer ON Loan.Customer_ID = Customer.Customer_ID;

http://joe-cruz.infinityfreeapp.com/dataleftjoin.php

```php
<?

@$db = mysql_pconnect("sql210.epizy.com", "epiz_31407029", "N94YMcLv6u5Q");

if (!$db) {
    echo "ERROR: Could not connect to database.  Please try again later.";
    exit;
}

mysql_select_db("epiz_31407029_sigma");

$query = "SELECT Customer.Cust_Fname, Customer.CustL_Lname, Loan.Loan_ID, Loan.Loan_Type FROM Loan
LEFT JOIN Customer ON Loan.Customer_ID = Customer.Customer_ID";

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

echo "<p>Number of Customers found: " . $num_results . "</p>";

for ($i = 0; $i < $num_results; $i++) {
    $row = mysql_fetch_array($result);
    echo "
                <tr class='data'>
                    <td>" . $row['Loan_ID'] . "</td>
                    <td>" . $row['Cust_Fname'] . " " . $row['CustL_Lname'] . "</td>
                    <td>" . $row['Loan_Type'] .  "</td>
                </tr>
            ";
}
```

Step 17:

We display all of the information about the customer and their corresponding account. The table is in descending order by their account balance. This also mimics a full outer join between the Customer and Account tables.

SELECT * FROM Customer LEFT JOIN Account ON Customer.Customer_ID = Account.Customer_ID UNION SELECT * FROM Customer RIGHT JOIN Account ON Customer.Customer_ID = Account.Customer_ID ORDER BY Account_Balance DESC;

http://joe-cruz.infinityfreeapp.com/dataorderby.php

```php
<?

@$db = mysql_pconnect("sql210.epizy.com", "epiz_31407029", "N94YMcLv6u5Q");

if (!$db) {
    echo "ERROR: Could not connect to database.  Please try again later.";
    exit;
}

mysql_select_db("epiz_31407029_sigma");

$query = "SELECT * FROM Customer LEFT JOIN Account ON Customer.Customer_ID = Account.Customer_ID UNION SELECT * FROM Customer RIGHT JOIN
Account ON Customer.Customer_ID = Account.Customer_ID ORDER BY Account_Balance DESC";

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

echo "<p>Number of Customers found: " . $num_results . "</p>";

for ($i = 0; $i < $num_results; $i++) {
    $row = mysql_fetch_array($result);
    echo "
                <tr class='data'>
                    <td>" . $row['Customer_ID'] . "</td>
                    <td>" . $row['Cust_Fname'] . " " . $row['CustL_Lname'] . "</td>
                    <td>" . $row['Account_Number'] .  "</td>
                    <td>" . $row['Account_Type'] . "</td>
                    <td>" . $row['Account_Balance'] . "</td>
                    <td>" . $row['Cust_Address'] . " " . $row['Cust_City'] . ", " . $row['Cust_State'] . "  " . $row['Cust_Country'] . "</td>
                    <td>" . $row['Phone'] . "</td>
                    <td>" . $row['Branch_ID'] . "</td>
                </tr>
            ";
}
```

Step 18:

This statement displays any transaction that was taken on the date after December 31, 2021.

SELECT Customer.Cust_Fname, Customer.CustL_Lname, Transaction.* FROM Transaction LEFT JOIN Customer ON Transaction.Customer_ID = Customer.Customer_ID where Transaction.Date > '2021-12-31' order by date asc;

http://joe-cruz.infinityfreeapp.com/datadate.php

```php
<?

@$db = mysql_pconnect("sql210.epizy.com", "epiz_31407029", "N94YMcLv6u5Q");

if (!$db) {
    echo "ERROR: Could not connect to database.  Please try again later.";
    exit;
}

mysql_select_db("epiz_31407029_sigma");

$query = "SELECT Customer.Cust_Fname, Customer.CustL_Lname, Transaction.* FROM Transaction LEFT JOIN Customer ON
Transaction.Customer_ID = Customer.Customer_ID where Transaction.Date > '2021-12-31' order by date asc";

$result = mysql_query($query);
$num_results = mysql_num_rows($result);

echo "<p>Number of Transactions found: " . $num_results . "</p>";

for ($i = 0; $i < $num_results; $i++) {
    $row = mysql_fetch_array($result);
    echo "
                    <tr class='data'>
                        <td>" . $row['Transaction_Number'] . "</td>
                        <td>" . $row['Amount'] . "</td>
                        <td>" . $row['Cust_Fname'] . " " . $row['CustL_Lname'] . "</td>
                        <td>" . $row['Date'] .  "</td>
                    </tr>
                ";
}

?>
```

Step 19:

Creates a table view using the Customer table and Transaction table to display the transaction dates and amount, as well as who made the transaction.

CREATE VIEW CustomerTransactions as SELECT Customer.Cust_Fname, Customer.CustL_Lname, Transaction.Date, Transaction.Amount from Customer INNER JOIN Transaction on Customer.Customer_ID=Transaction.Customer_ID

	Cust_Fname	CustL_Lname	Date	Amount
Edit Copy Delete	Kanye	West	2021-12-27	-500
Edit Copy Delete	Ryan	Gosling	2021-10-13	23
Edit Copy Delete	Steve	Jobs	2022-05-01	2173
Edit Copy Delete	Alex	Speer	2022-02-28	6409
Edit Copy Delete	Ryan	Gosling	2022-01-06	1367
Edit Copy Delete	Ryan	Gosling	2021-03-16	6927
Edit Copy Delete	Casey	Novelo	2021-05-19	-9
Edit Copy Delete	Steve	Jobs	2022-03-24	82
Edit Copy Delete	Ryan	Gosling	2022-02-15	49
Edit Copy Delete	Trent	Phelps	2021-12-14	-2