CSE370 Lab 3 (KEY ATTRIBUTE)

Saturday, February 10, 2024 7:31 PM

A database can contain multiple tables.

Bank database

Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

Account

branch_name	account_number	balance
Downtown	A-101	500
Perryridge	A-102	400
Brighton	A-201	900
Mianus	A-215	700
Brighton	A-217	750
Redwood	A-222	700
Round Hill	A-305	350

Borrower

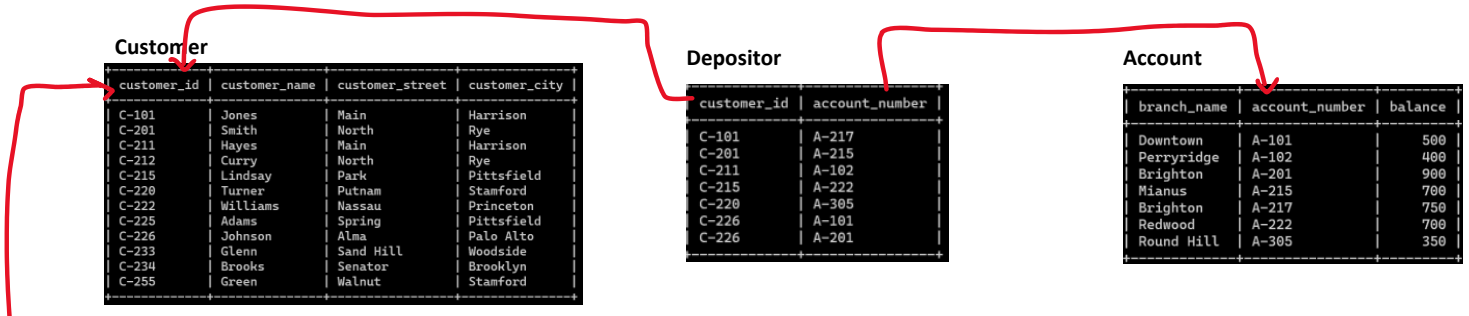
customer_id	loan_number
C-101	L-17
C-201	L-11
C-201	L-23
C-211	L-15
C-212	L-93
C-222	L-17
C-225	L-16
C-226	L-14

Loan

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Branch

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000



Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

Account

branch_name	account_number	balance
Downtown	A-101	500
Perryridge	A-102	400
Brighton	A-201	900
Mianus	A-215	700
Brighton	A-217	750
Redwood	A-222	700
Round Hill	A-305	350

Borrower

customer_id	loan_number
C-101	L-17
C-201	L-11
C-201	L-23
C-211	L-15
C-212	L-93
C-222	L-17
C-225	L-16
C-226	L-14

Loan

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Branch

branch_name	branch_city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Customer

How to identify each customer uniquely?

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

KEY ATTRIBUTE!!!

There can be multiple key attributes customer_id, NID, phone number, email etc

One key attribute that we will select --> **PRIMARY KEY**

Let us select customer_id as primary key for this Customer table

SQL Query:

```
create table customer (
customer_id varchar(10),
customer_name varchar(20) ,
customer_street varchar(30),
customer_city varchar(30),
primary key (customer_id)
);
```

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

Primary Key???

No attribute can uniquely identify each row in this table. So we will use multiple attributes as primary key!!

SQL Query:

```
create table depositor (
customer_id varchar(10),
account_number varchar(10),
primary key (customer_id,account_number),
);
```

Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

primary Key

Account

branch_name	account_number	balance
Downtown	A-101	500
Perryridge	A-102	400
Brighton	A-201	900
Mianus	A-215	700
Brighton	A-217	750
Redwood	A-222	700
Round Hill	A-305	350

primary Key

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

primary Key primary Key

SQL Query:

```
create table depositor (
customer_id varchar(10),
account_number varchar(10),
primary key (customer_id,account_number),
foreign key (customer_id) references
customer(customer_id),
foreign key (account_number) references
account(account_number)
);
```

Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

primary Key

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

primary Key primary Key

Foreign Key

What if we delete customer_id **C-101** from the **Customer Table**? What happens to the values connected to customer_id = 'C-101'?

Three ways to handle this situation!!!

1. Cascade
2. Set Null
3. Restrict

Cascade

SQL Query:

```
create table depositor (
customer_id varchar(10),
account_number varchar(10),
primary key (customer_id,account_number),
foreign key (customer_id) references
customer(customer_id) ON UPDATE CASCADE ON
DELETE CASCADE,
foreign key (account_number) references
account(account_number)
);
```

ON UPDATE CASCADE --> If you update the primary table, it will update the connected tables accordingly

ON DELETE CASCADE --> If you delete any value from the primary table, same values will get deleted from the connected table

Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

primary Key

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

primary Key primary Key

Foreign Key

SET NULL

SQL Query:

```
create table depositor (
customer_id varchar(10),
account_number varchar(10),
primary key (customer_id,account_number),
foreign key (customer_id) references
customer(customer_id) ON UPDATE SET NULL ON
DELETE SET NULL,
foreign key (account_number) references
```

ON UPDATE SET NULL or ON DELETE SET NULL--> the values will be set to NULL in the connected tables

account(account_number)
);

Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

primary Key

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

primary Key primary Key

Foreign Key

Restrict

SQL Query:

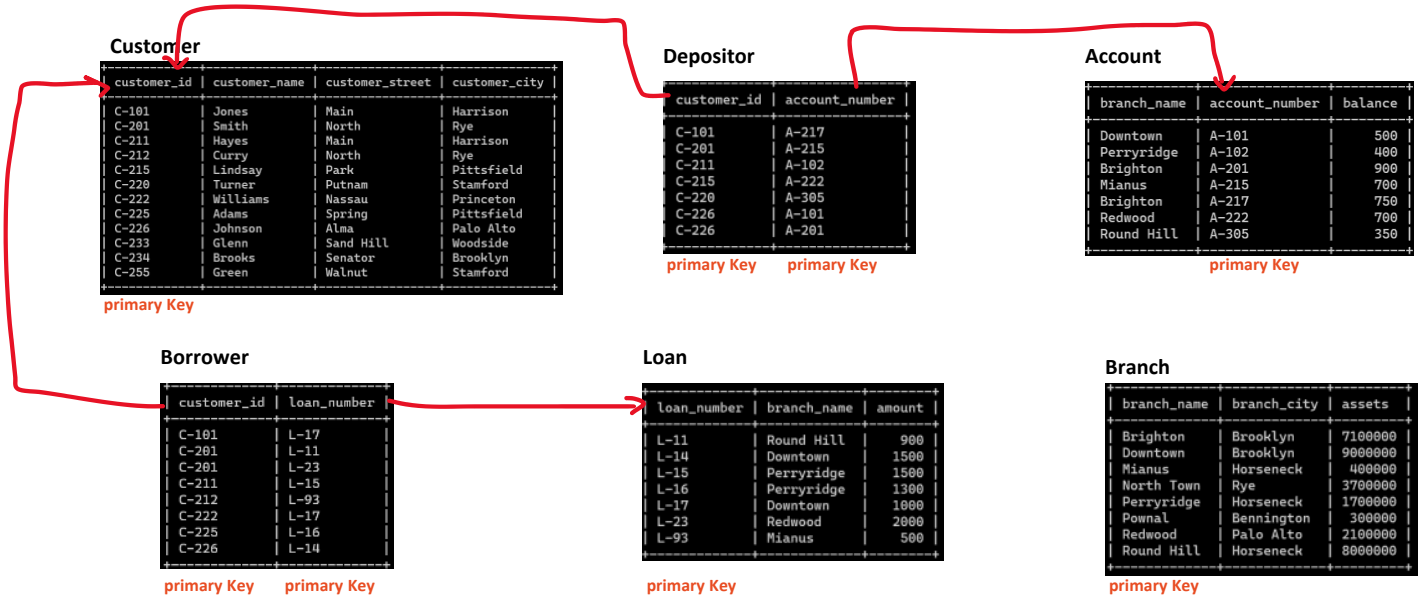
```
create table depositor (
customer_id varchar(10),
account_number varchar(10),
primary key (customer_id,account_number),
foreign key (customer_id) references
customer(customer_id),
foreign key (account_number) references
account(account_number)
);
```

If we do not mention anything, by default they are 'Restricted'.

That means, if user tries to update or delete any value inthe primary table, it won't allow the user to update and will show an error message.

CSE370 Lab 3 (JOINS)

Saturday, February 10, 2024 8:38 PM



1. Show the customer name and their corresponding account number.
2. Show the customer name and their corresponding account number of those who has their account in 'Brighton' branch

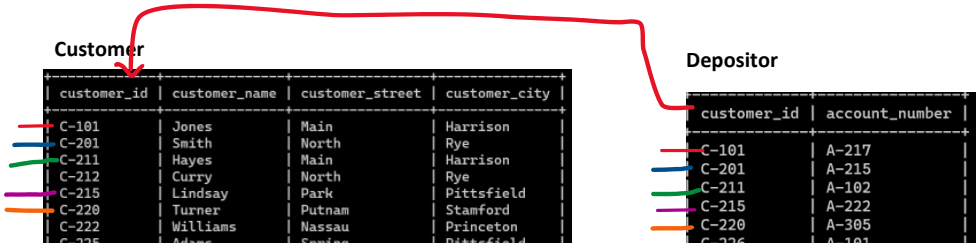
1. Need to join Customer table and Depositor table
2. Need to join Customer table, Depositor table and account table

JOINS

There are 4 types of join:

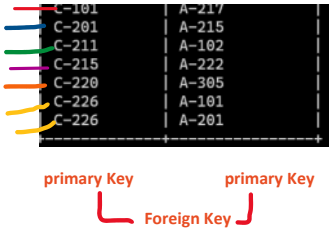
1. **Inner join**
2. **Left Join**
3. **Right Join**
4. **Full outer join (not supported by mysql)**

Inner Join



C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

primary Key



C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

primary Key primary Key

Foreign Key

Only Common values from both tables!!!!

SQL Query:

```
SELECT customer_name, account_number FROM
customer INNER JOIN depositor ON
customer.customer_id = depositor.customer_id;
```

Specifying the table name →

SQL Query:

```
SELECT customer.customer_id, customer_name,
account_number FROM customer INNER JOIN depositor
ON customer.customer_id = depositor.customer_id;
```

Using aliasing for tables →

SQL Query:

```
SELECT C.customer_id, C.customer_name,
D.account_number FROM customer C INNER JOIN
depositor D ON C.customer_id = D.customer_id;
```

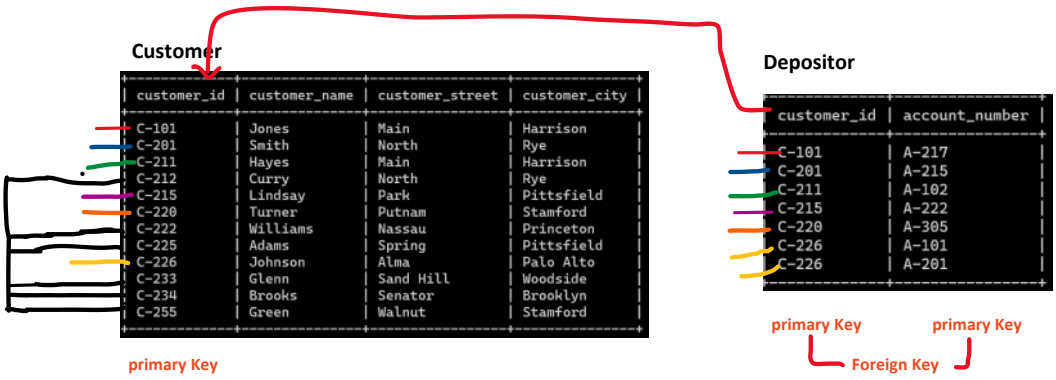
I want to see the information only for customer_name 'Johnson' →

SQL Query:

```
SELECT C.customer_id, C.customer_name,
D.account_number FROM customer C INNER JOIN
depositor D ON C.customer_id = D.customer_id WHERE
C.customer_name = 'Johnson';
```

What will happen if we select *?

Left Join



Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

All rows from left table and common rows from the right table!!!

SQL Query:

```
SELECT customer_name, account_number FROM
customer LEFT JOIN depositor ON
customer.customer_id = depositor.customer_id;
```

Right Join

Customer

customer_id	customer_name	customer_street	customer_city
C-101	Jones	Main	Harrison
C-201	Smith	North	Rye
C-211	Hayes	Main	Harrison
C-212	Curry	North	Rye
C-215	Lindsay	Park	Pittsfield
C-220	Turner	Putnam	Stamford
C-222	Williams	Nassau	Princeton
C-225	Adams	Spring	Pittsfield
C-226	Johnson	Alma	Palo Alto
C-233	Glenn	Sand Hill	Woodside
C-234	Brooks	Senator	Brooklyn
C-255	Green	Walnut	Stamford

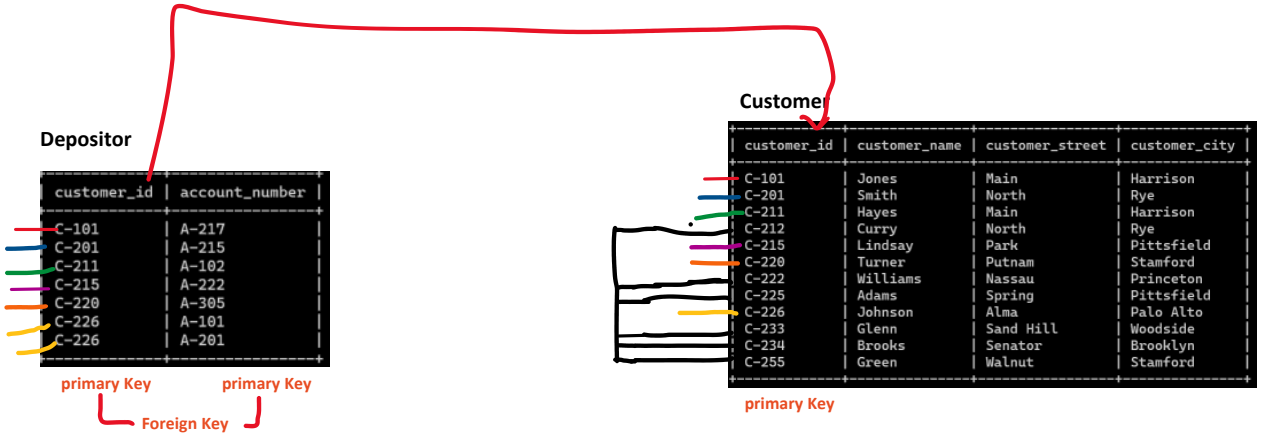
Depositor

customer_id	account_number
C-101	A-217
C-201	A-215
C-211	A-102
C-215	A-222
C-220	A-305
C-226	A-101
C-226	A-201

All rows from right table and common rows from the right table!!!

SQL Query:

```
SELECT customer_name, account_number FROM
customer RIGHT JOIN depositor ON
customer.customer_id = depositor.customer_id;
```



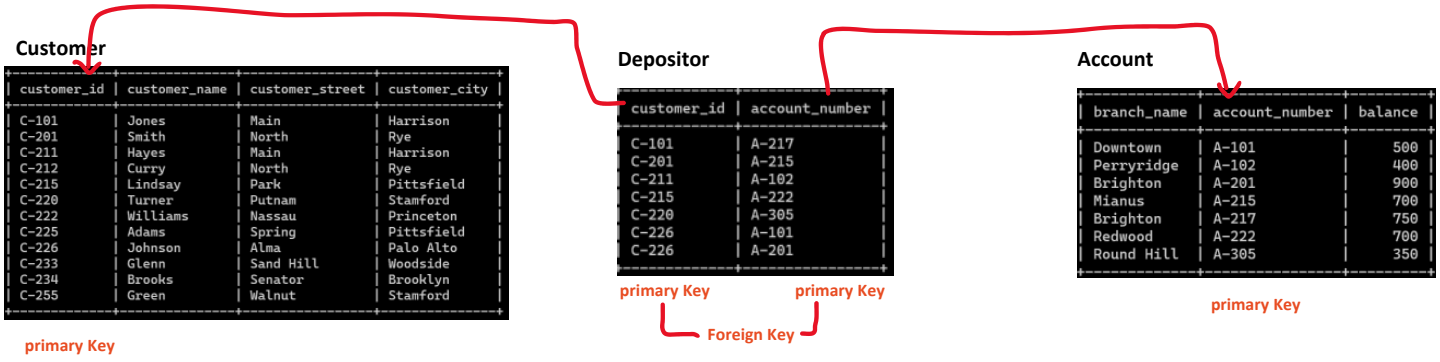
All rows from left table and common rows from the right table!!!

SQL Query:

```
SELECT customer_name, account_number FROM
depositor RIGHT JOIN customer ON
customer.customer_id = depositor.customer_id;
```

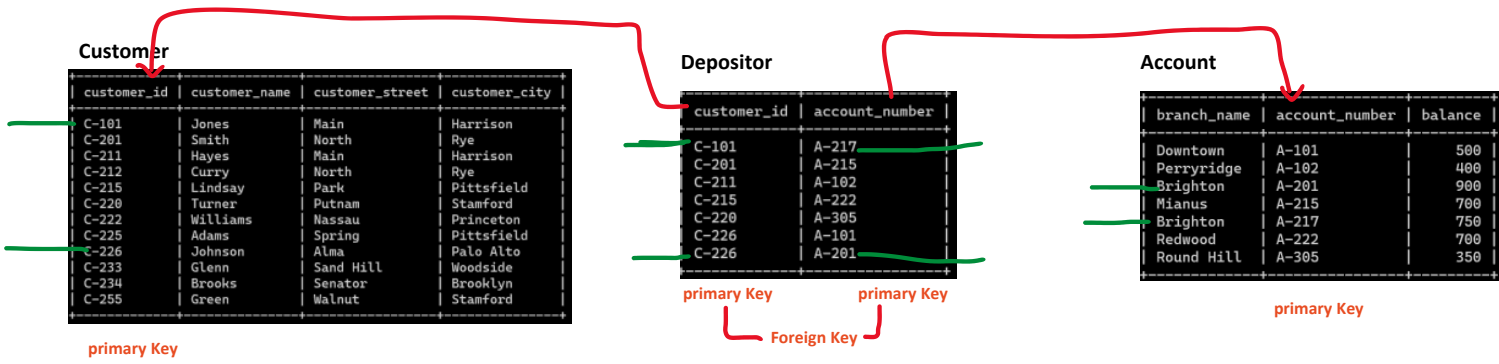
Joining More than one Table

Show the customer name, their corresponding account number and branch name of those who has their account in 'Brighton' branch



SQL Query:

```
SELECT C.customer_name, D.account_number,
A.branch_name FROM (depositor D INNER JOIN
customer C ON C.customer_id = D.customer_id) INNER
JOIN Account A ON A.account_number =
D.account_number WHERE A.branch_name  =
'Brighton';
```



Alternative way to do INNER join more than two tables

SQL Query:

```
SELECT C.customer_name, D.account_number,
A.branch_name FROM
depositor D,  customer C, account A WHERE
C.customer_id = D.customer_id and A.account_number
= D.account_number and A.branch_name  = 'Brighton';
```

Not for LEFT or Right join