

CSE370 LAB 3

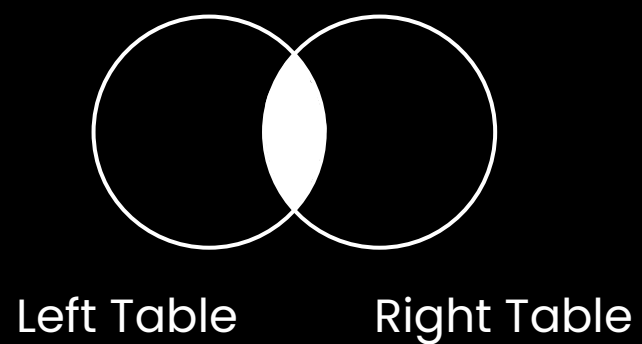
SQL JOINS AND CONSTRAINTS

Prepared By: M. Shafiul Alam [FUL/SFA]



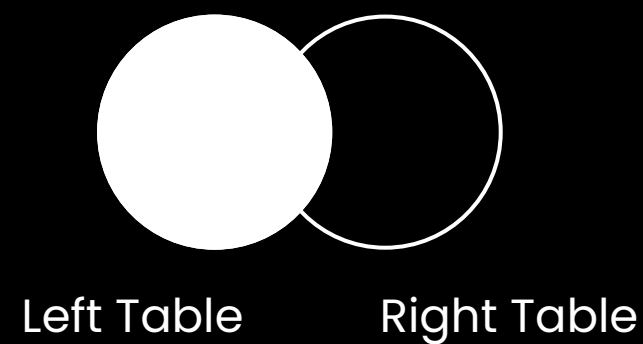
SQL JOINS

1. Inner Join



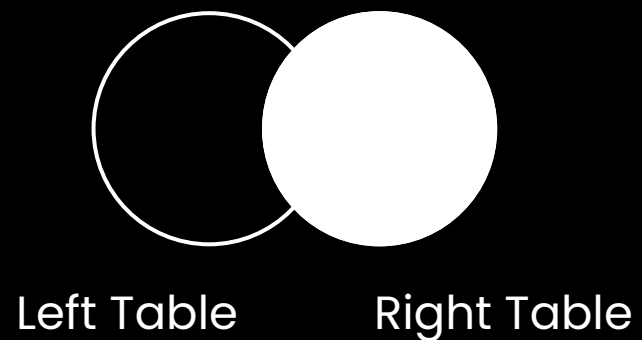
$$\text{Inner Join} = \text{Left Table} \cap \text{Right Table}$$

2. Left Join



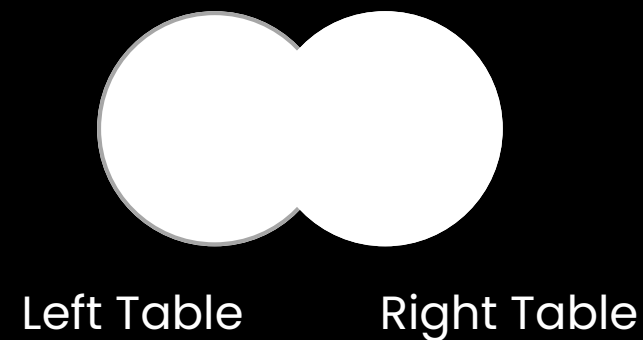
$$\text{Left Join} = \text{Left Table} \cup (\text{Left Table} \cap \text{Right Table})$$

3. Right Join



$$\text{Right Join} = \text{Right Table} \cup (\text{Right Table} \cap \text{Left Table})$$

4. Full Join



$$\text{Full Join} = \text{Left Table} \cup \text{Right Table}$$

SQL JOINS

id	name
1	Abrar
2	Bashir
3	Cassie
4	Dylan
5	Erin
6	NULL

Students

id	hobby
2	Fishing
1	Singing
3	Gardening
1	Gaming
5	Sleeping
9	NULL

Hobbies

1. Inner Join:

```
MariaDB [jointest]> select t1.name, t2.hobby from students t1 join hobbies t2 on t1.id = t2.id;
```

name	hobby
Bashir	Fishing
Abrar	Singing
Cassie	Gardening
Abrar	Gaming
Erin	Sleeping

5 rows in set (0.026 sec)

2. Left Join:

```
MariaDB [jointest]> select t1.name, t2.hobby from students t1 left join hobbies t2 on t1.id = t2.id;
```

name	hobby
Bashir	Fishing
Abrar	Singing
Cassie	Gardening
Abrar	Gaming
Erin	Sleeping
Dylan	NULL
NULL	NULL

7 rows in set (0.000 sec)

SQL JOINS

id	name
1	Abrar
2	Bashir
3	Cassie
4	Dylan
5	Erin
6	NULL

Students

id	hobby
2	Fishing
1	Singing
3	Gardening
1	Gaming
5	Sleeping
9	NULL

Hobbies

3. Right Join:

```
MariaDB [jointest]> select t1.name, t2.hobby from students t1 right join hobbies t2
on t1.id = t2.id;
+-----+-----+
| name  | hobby |
+-----+-----+
| Abrar  | Singing |
| Abrar  | Gaming |
| Bashir  | Fishing |
| Cassie  | Gardening |
| Erin   | Sleeping |
| NULL   | NULL |
+-----+-----+
5 rows in set (0.000 sec)
```

4. Full Join:

```
MariaDB [jointest]> select t1.name, t2.hobby from students t1 left join hobbies t2 on t1.id = t2.id
-> union
-> select t1.name, t2.hobby from students t1 right join hobbies t2 on t1.id = t2.id;
+-----+-----+
| name  | hobby |
+-----+-----+
| Bashir | Fishing |
| Abrar  | Singing |
| Cassie  | Gardening |
| Abrar  | Gaming |
| Erin   | Sleeping |
| Dylan  | NULL |
| NULL   | NULL |
+-----+-----+
7 rows in set (0.001 sec)
```

SQL JOINS

Task 3:

Retrieve all customer's id, name, city and account number using

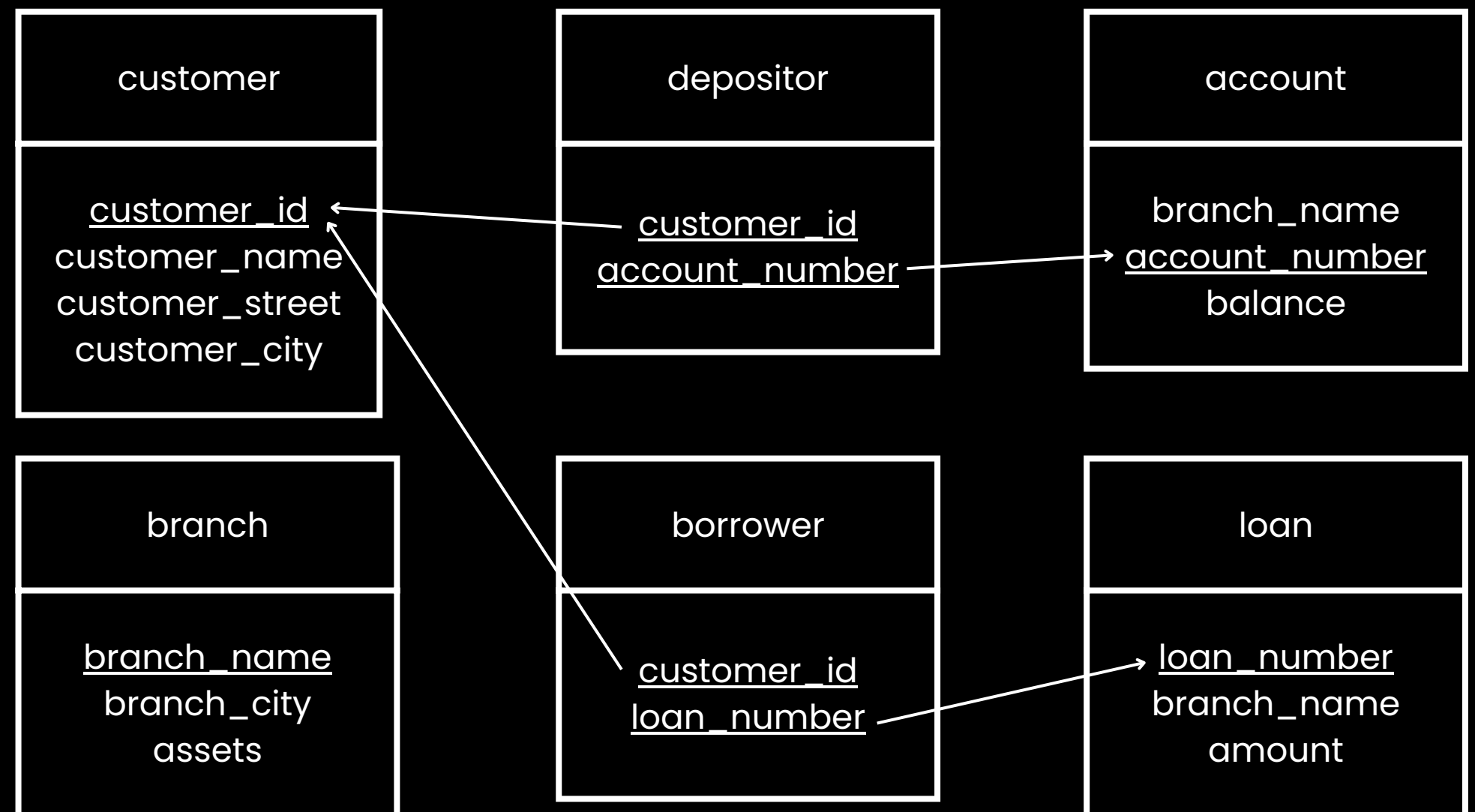
1. Inner Join
2. Left Join
3. Right Join
4. Full Join

SQL JOINS

Task 3:

Retrieve all customer's id, name, city and account number using

1. Inner Join
2. Left Join
3. Right Join
4. Full Join



SQL JOINS - INNER JOIN

Task 3:

Retrieve all customer's id, name, city and account number using

1. Inner Join
2. Left Join
3. Right Join
4. Full Join

1.1 select c.customer_id, c.customer_name,
c.customer_city, d.account_number from
customer c inner join depositor d on
c.customer_id = d.customer_id;

1.2 select c.customer_id, c.customer_name,
c.customer_city, d.account_number from
customer c join depositor d on c.customer_id =
d.customer_id;

customer_id	customer_name	customer_city	account_number
C-101	Jones	Harrison	A-217
C-201	Smith	Rye	A-215
C-211	Hayes	Harrison	A-102
C-215	Lindsay	Pittsfield	A-222
C-220	Turner	Stamford	A-305
C-226	Johnson	Palo Alto	A-101
C-226	Johnson	Palo Alto	A-201

7 rows in set (0.001 sec)

SQL JOINS - INNER JOIN

Task 3:

Retrieve all customer's id, name, city and account number using

1. Inner Join
2. Left Join
3. Right Join
4. Full Join

1.3 select c.customer_id, c.customer_name,
c.customer_city, d.account_number from
customer c, depositor d where c.customer_id =
d.customer_id;

customer_id	customer_name	customer_city	account_number
C-101	Jones	Harrison	A-217
C-201	Smith	Rye	A-215
C-211	Hayes	Harrison	A-102
C-215	Lindsay	Pittsfield	A-222
C-220	Turner	Stamford	A-305
C-226	Johnson	Palo Alto	A-101
C-226	Johnson	Palo Alto	A-201

7 rows in set (0.001 sec)

SQL JOINS - LEFT JOIN

Task 3:

Retrieve all customer's id, name, city and account number using

1. Inner Join
2. Left Join
3. Right Join
4. Full Join

2. select c.customer_id, c.customer_name,
c.customer_city, d.account_number from
customer c left join depositor d on c.customer_id
= d.customer_id;

customer_id	customer_name	customer_city	account_number
C-101	Jones	Harrison	A-217
C-201	Smith	Rye	A-215
C-211	Hayes	Harrison	A-102
C-212	Curry	Rye	NULL
C-215	Lindsay	Pittsfield	A-222
C-220	Turner	Stamford	A-305
C-222	Williams	Princeton	NULL
C-225	Adams	Pittsfield	NULL
C-226	Johnson	Palo Alto	A-101
C-226	Johnson	Palo Alto	A-201
C-233	Glenn	Woodside	NULL
C-234	Brooks	Brooklyn	NULL
C-255	Green	Stamford	NULL

SQL JOINS - RIGHT JOIN

Task 3:

Retrieve all customer's id, name, city and account number using

1. Inner Join
2. Left Join
3. Right Join
4. Full Join

3. select c.customer_id, c.customer_name,
c.customer_city, d.account_number from
customer c right join depositor d on
c.customer_id = d.customer_id;

customer_id	customer_name	customer_city	account_number
C-101	Jones	Harrison	A-217
C-201	Smith	Rye	A-215
C-211	Hayes	Harrison	A-102
C-215	Lindsay	Pittsfield	A-222
C-220	Turner	Stamford	A-305
C-226	Johnson	Palo Alto	A-101
C-226	Johnson	Palo Alto	A-201

SQL JOINS - FULL JOIN

Task 3:

Retrieve all customer's id, name, city and account number using

1. Inner Join
2. Left Join
3. Right Join
4. Full Join

```
4. select c.customer_id, c.customer_name,  
c.customer_city, d.account_number from customer c  
left join depositor d on c.customer_id = d.customer_id  
union  
select c.customer_id, c.customer_name,  
c.customer_city, d.account_number from customer c  
right join depositor d on c.customer_id =  
d.customer_id;
```

customer_id	customer_name	customer_city	account_number
C-101	Jones	Harrison	A-217
C-201	Smith	Rye	A-215
C-211	Hayes	Harrison	A-102
C-212	Curry	Rye	NULL
C-215	Lindsay	Pittsfield	A-222
C-220	Turner	Stamford	A-305
C-222	Williams	Princeton	NULL
C-225	Adams	Pittsfield	NULL
C-226	Johnson	Palo Alto	A-101
C-226	Johnson	Palo Alto	A-201
C-233	Glenn	Woodside	NULL
C-234	Brooks	Brooklyn	NULL
C-255	Green	Stamford	NULL

SQL JOINS

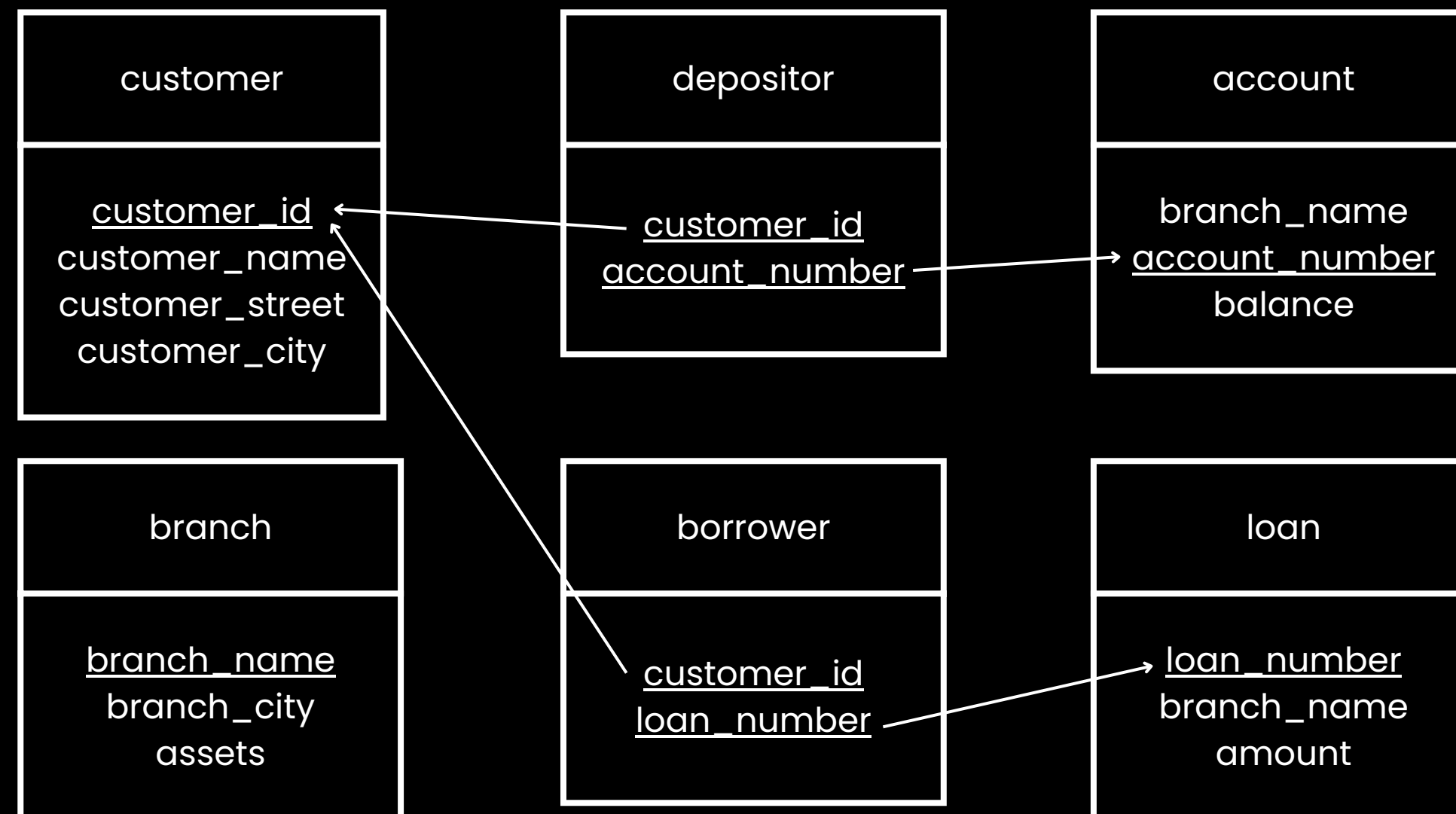
Task 4:

Retrieve the following information from your database using “join”: Customer name, city, account number, balance and branch name.

SQL JOINS

Task 4:

Retrieve the following information from your database using “join”: Customer name, city, account number, balance and branch name.



SQL JOINS

Task 4:

Retrieve the following information from your database using “join”: Customer name, city, account number, balance and branch name.

-> select c.customer_name, c.customer_city, a.account_number, a.balance, a.branch_name from customer c inner join depositor d on d.customer_id = c.customer_id inner join account a on d.account_number = a.account_number;

customer_name	customer_city	account_number	balance	branch_name
Jones	Harrison	A-217	750	Brighton
Smith	Rye	A-215	700	Mianus
Hayes	Harrison	A-102	400	Perryridge
Lindsay	Pittsfield	A-222	700	Redwood
Turner	Stamford	A-305	350	Round Hill
Johnson	Palo Alto	A-101	500	Downtown
Johnson	Palo Alto	A-201	900	Brighton

SQL JOINS

Task 5:

Retrieve the following information from your database using “join”: Customer name, city, account number, balance and branch name (without using join keyword).

SQL JOINS

Task 5:

Retrieve the following information from your database using “join”: Customer name, city, account number, balance and branch name (without using join keyword).

-> select c.customer_name, c.customer_city, a.account_number, a.balance, a.branch_name from customer c, account a, depositor d where c.customer_id = d.customer_id and d.account_number = a.account_number;

customer_name	customer_city	account_number	balance	branch_name
Jones	Harrison	A-217	750	Brighton
Smith	Rye	A-215	700	Mianus
Hayes	Harrison	A-102	400	Perryridge
Lindsay	Pittsfield	A-222	700	Redwood
Turner	Stamford	A-305	350	Round Hill
Johnson	Palo Alto	A-101	500	Downtown
Johnson	Palo Alto	A-201	900	Brighton

SQL JOINS

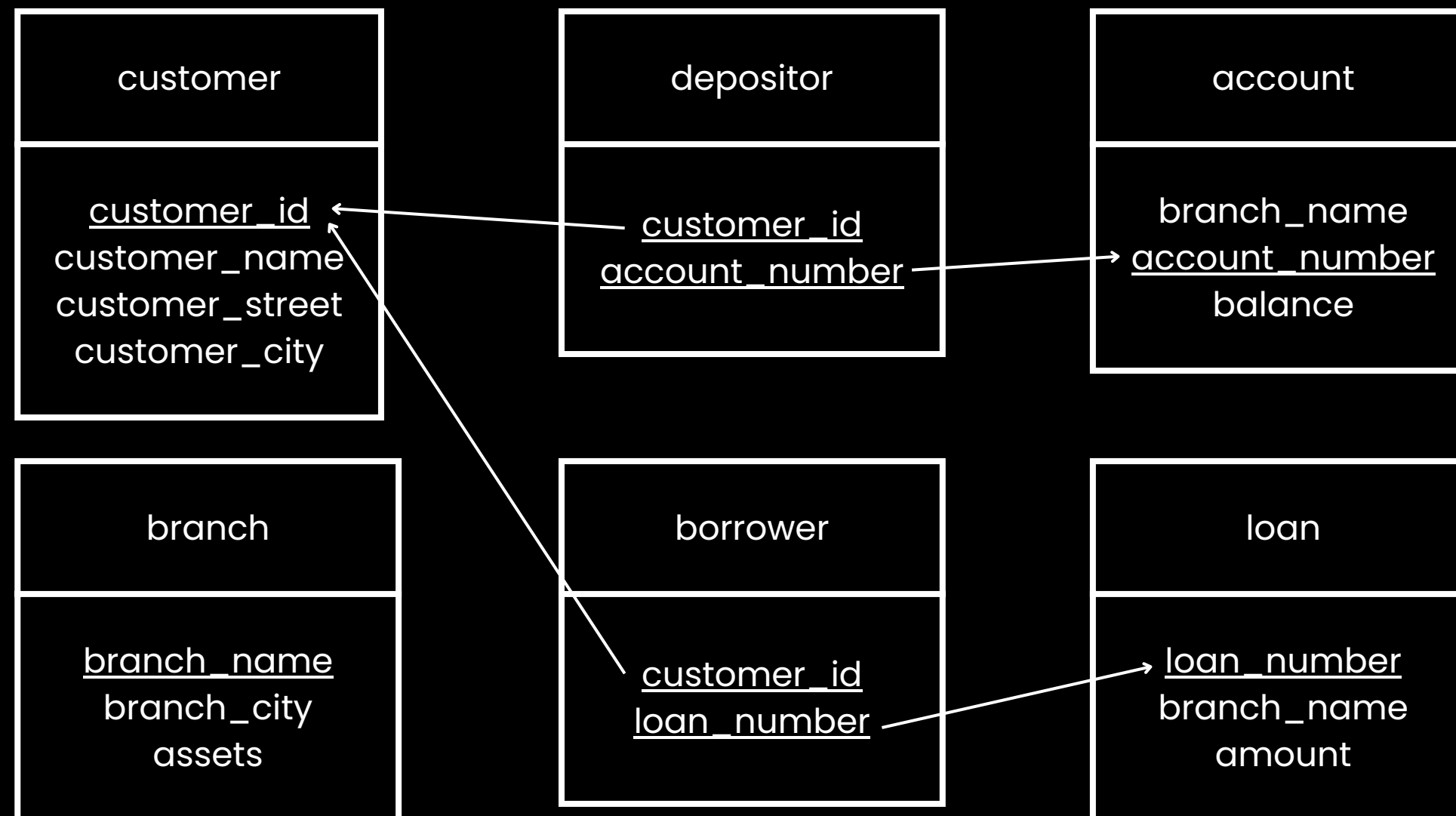
Task 6.1:

Find names and cities of customers who have a loan at Perryridge branch

SQL JOINS

Complete Task 6

Note: The following diagram is attached for reference



SQL JOINS

Task 6.1:

Find names and cities of customers who have a loan at Perryridge branch

-> select c.customer_name, c.customer_city from customer c, borrower b, loan l where c.customer_id = b.customer_id and b.loan_number = l.loan_number and l.branch_name = 'Perryridge';

customer_name	customer_city
Hayes	Harrison
Adams	Pittsfield

SQL JOINS

Task 6.2:

Find the accounts with balances between 700 and 900

SQL JOINS

Task 6.2:

Find the accounts with balances between 700 and 900

-> select account_number,balance from account where balance between 700 and 900;

account_number	balance
A-201	900
A-215	700
A-217	750
A-222	700

SQL JOINS

Task 6.3:

Find the names of customers on streets with names ending in "Hill"

SQL JOINS

Task 6.3:

Find the names of customers on streets with names ending in "Hill"

-> `select customer_name, customer_street from customer where customer_street like '%Hill';`

customer_name
Glenn

SQL JOINS

Task 6.4:

Find the names of branches whose assets are greater than the assets of some branch in Brooklyn

SQL JOINS

Task 6.4:

Find the names of branches whose assets are greater than the assets of some branch in Brooklyn

-> select branch_name from branch where assets > any (select assets from branch where branch_city = 'Brooklyn');

+	-	-	-	-	-	-	-	+
				branch_name				
+	-	-	-	-	-	-	-	+
				Downtown				
				Round Hill				
+	-	-	-	-	-	-	-	+

SQL JOINS

Task 6.5:

Find the set of names of branches whose assets are greater than the assets of all branches in Horseneck

SQL JOINS

Task 6.6:

Find the set of names of customers at Brighton branch, in alphabetical order

SQL JOINS

Task 6.6:

Find the set of names of customers at Brighton branch, in alphabetical order

-> select c.customer_name from customer c, depositor d, account a where c.customer_id = d.customer_id and a.account_number = d.account_number and a.branch_name = 'Brighton' order by c.customer_name;

customer_name
Johnson
Jones

SQL JOINS

Task 6.7:

Show the loan data, ordered by decreasing amounts, then increasing loan numbers

SQL JOINS

Task 6.7:

Show the loan data, ordered by decreasing amounts, then increasing loan numbers

-> `select * from loan order by amount desc, loan_number asc;`

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

SQL JOINS

Task 6.8:

Find the names of branches having at least one account, with average balances greater than or equal 700.

SQL JOINS

Task 6.8:

Find the names of branches having at least one account, with average balances greater than or equal 700.

```
-> select branch_name from account group by branch_name having avg(balance) >= 700;
```

```
+-----+
| branch_name |
+-----+
| Brighton    |
| Mianus      |
| Redwood     |
+-----+
```

SQL JOINS

Task 6.9:

Find the names and account number of customers who have the 3 highest balances in their accounts

SQL JOINS

Task 6.9:

Find the names and account number of customers who have the 3 highest balances in their accounts

-> select c.customer_name, a.account_number, a.balance from customer c, depositor d, account a where c.customer_id = d.customer_id and a.account_number = d.account_number order by a.balance desc limit 3;

customer_name	account_number	balance
Johnson	A-201	900
Jones	A-217	750
Smith	A-215	700

COMPLETE TASK 7 AND ASSIGNMENT-3