과목 명: 데이터베이스시스템

담당 교수 명: 박 석

<<Project 2>>

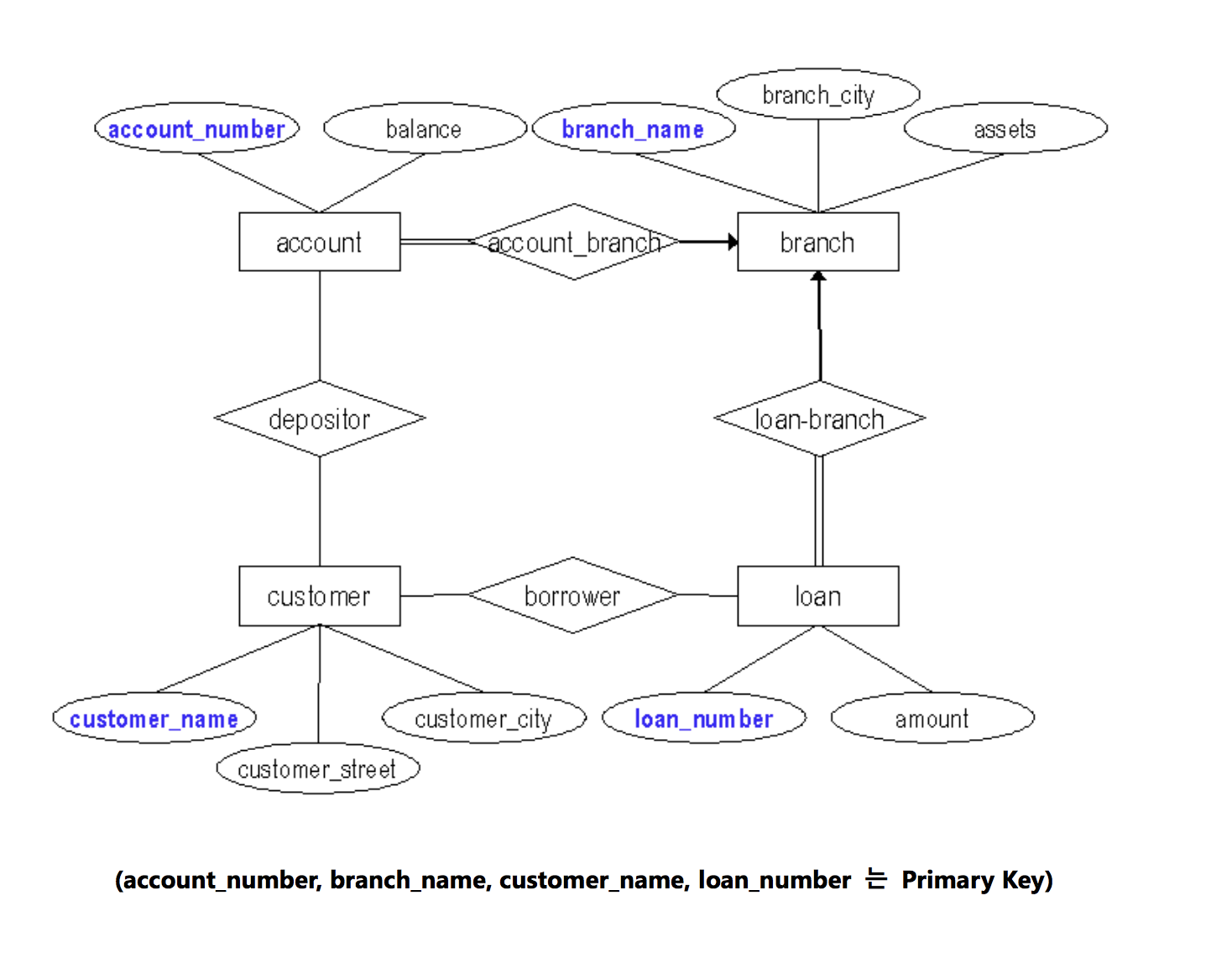
서강대학교컴퓨터공학과

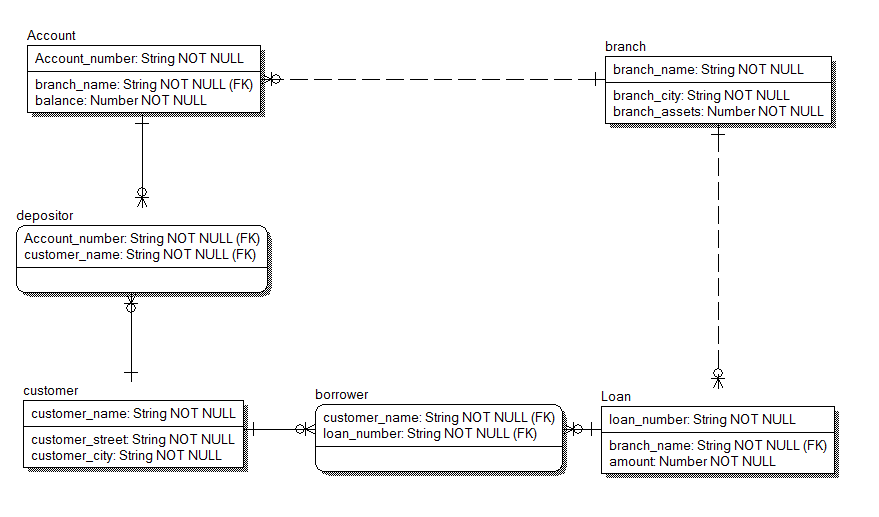
**[20121635]**

**[**장종석**]**

1 문제 정의

아래 ER 다이어그램에 맞는 데이터베이스를 설계하고 입력 데이터를 바탕으로 속성별 타입 과 제약 사항을 설정한 후, ER-win을 이용한 모드 별 설계 내용과 ER-win에서 생성된 데이터 베이스 스키마와 설계에 대한 세부 요구 사항 지정 및 설명 등에 대한 내용을 작성해 제출해 야 한다. 또한 실제 데이터베이스 서버와 연동한 후 스키마를 생성하고, 주어진 질의를 수행 할 수 있는 SQL문을 작성해 그에 따른 결과를 SQL문과 함께 보고서에 추가하여 제출한다.





전체 ER 다이어그래프를 보게되면 두가지 타입을 사용하고있다. String, int 가 그 두가지 이다. 아래장에서 제공하는 input 데이터를 보게 되면 String의 경우 20자리를 넘어가지 않는 것을 볼 수 있다. 이를 표현하기 위해서 2가지 도메인을 사용한다.

Entity의 경우 총 6가지를 사용하게 되는데, Account, branch, customer, loan, depositor, borrower이다. account-branch와 loan-branch의 경우 각각 account와, loan에 brahcn\_name을 외부키로 사용하면 굳이 필요가 없기 떄문이다.

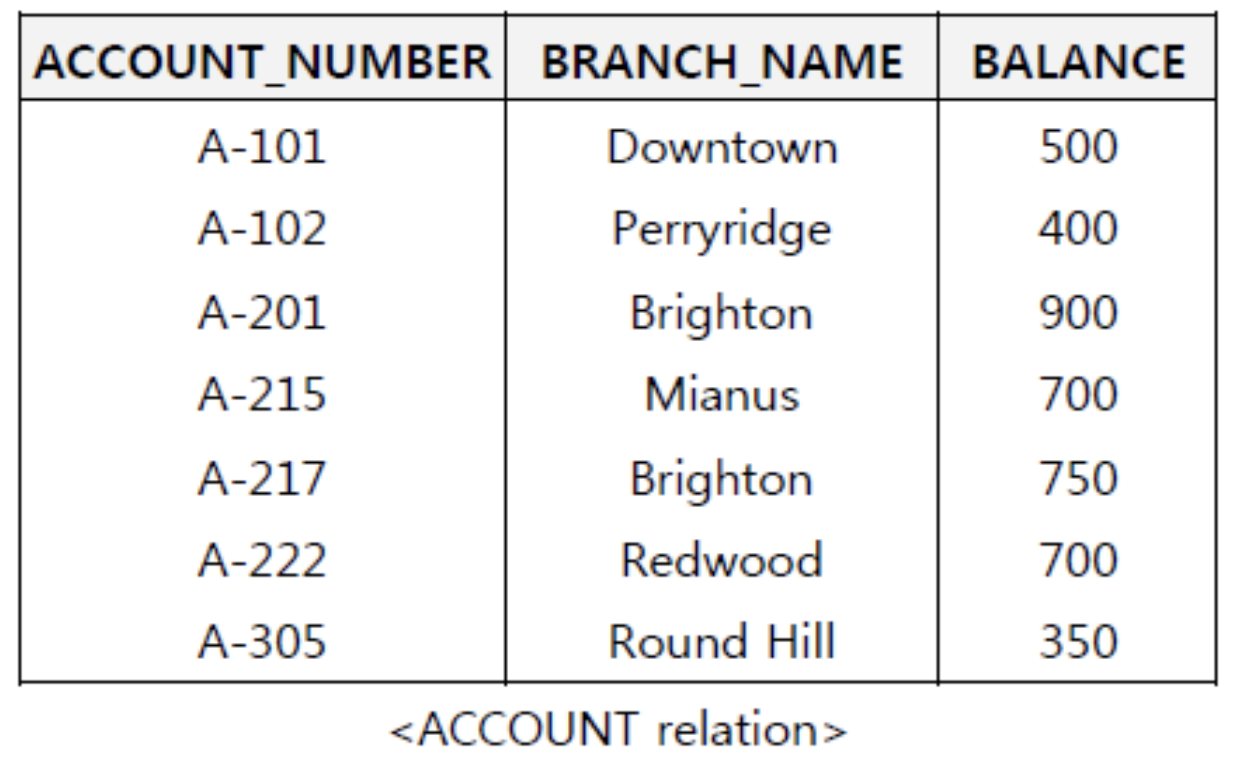
각각의 제한사항은 모두 NOT NULL인 상황으로 나타낼 수 있다.

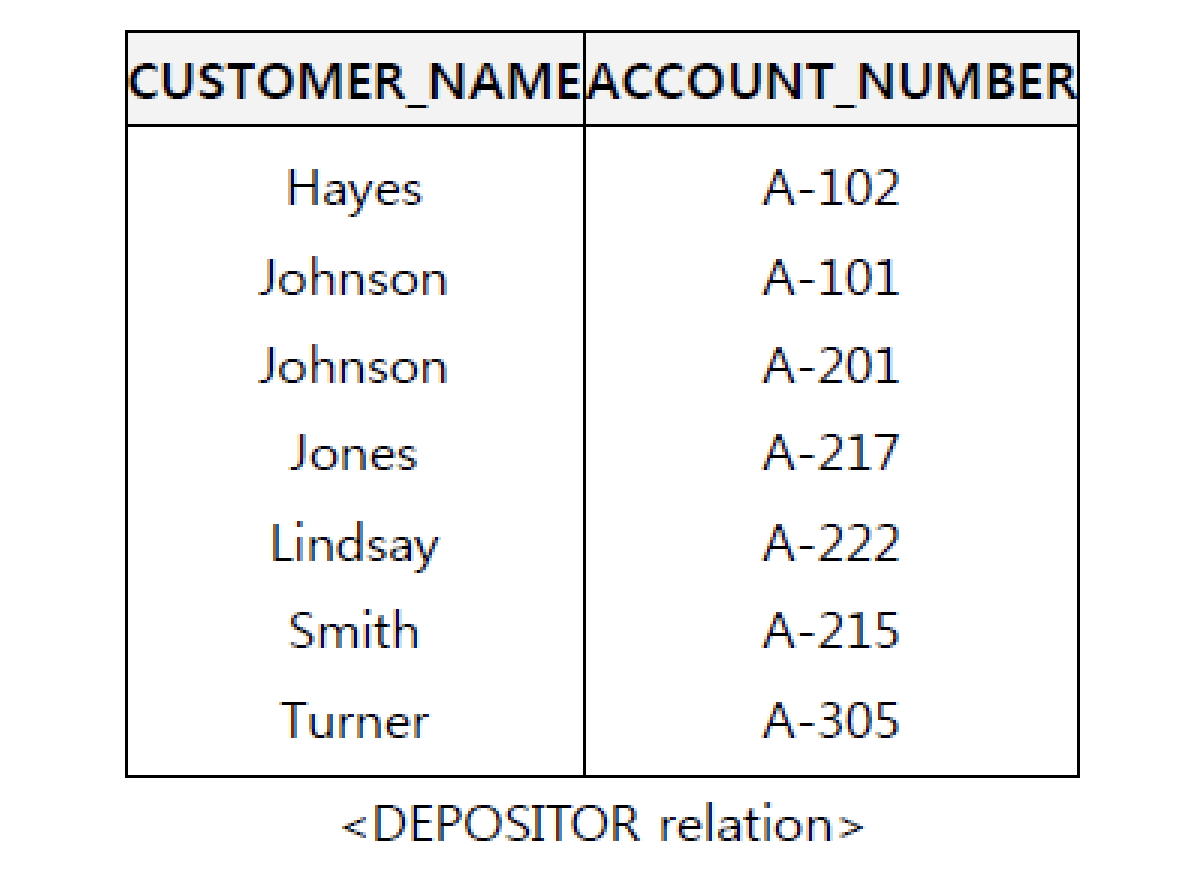
branch table과 account table, loan table은 1:N cardinality를 갖는 relation이다. account table과 loan table에서 branch table의 primary key인 branch\_name을 foreign key로 갖기 때문에 branch table을 이 두 table보다 먼저 생성했다. 또한 account table과 customer table은 N:N cardinality를 갖는 relation이기 때문에 이 두 table사이에 depositor table을 생성해주어야 하며, 이 table은 두 table의 primary key인 account\_number과 customer\_name을 foreign key로 갖는다. 마찬가지로 account table과 loan table도 N:N cardinality를 갖는 relation이기 이기때문에 위와 같이 두 table 사이에 borrower table을 생성해주어야 하며, 이 table은 두 table의 primary key인 loan\_number과 customer\_name을 foreign key로 갖는다.

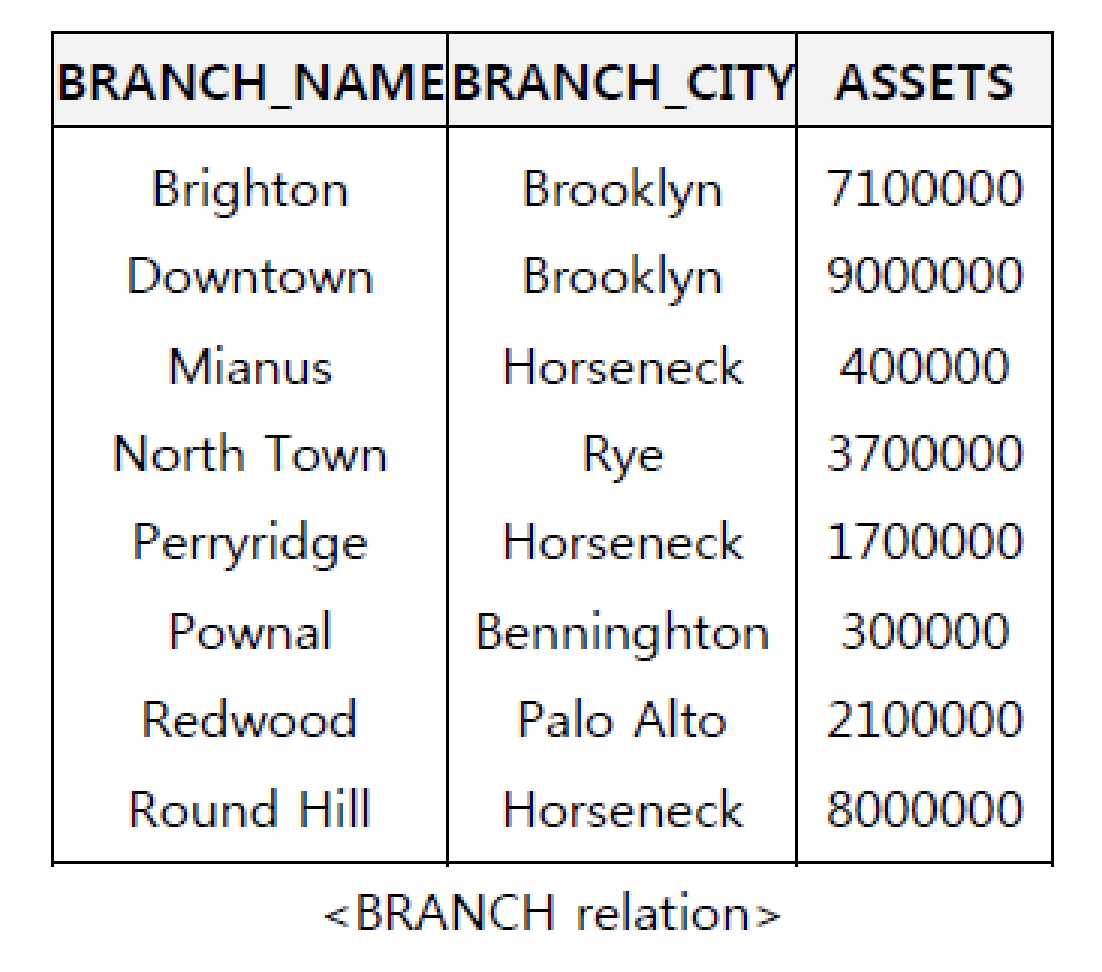
2 요구사항

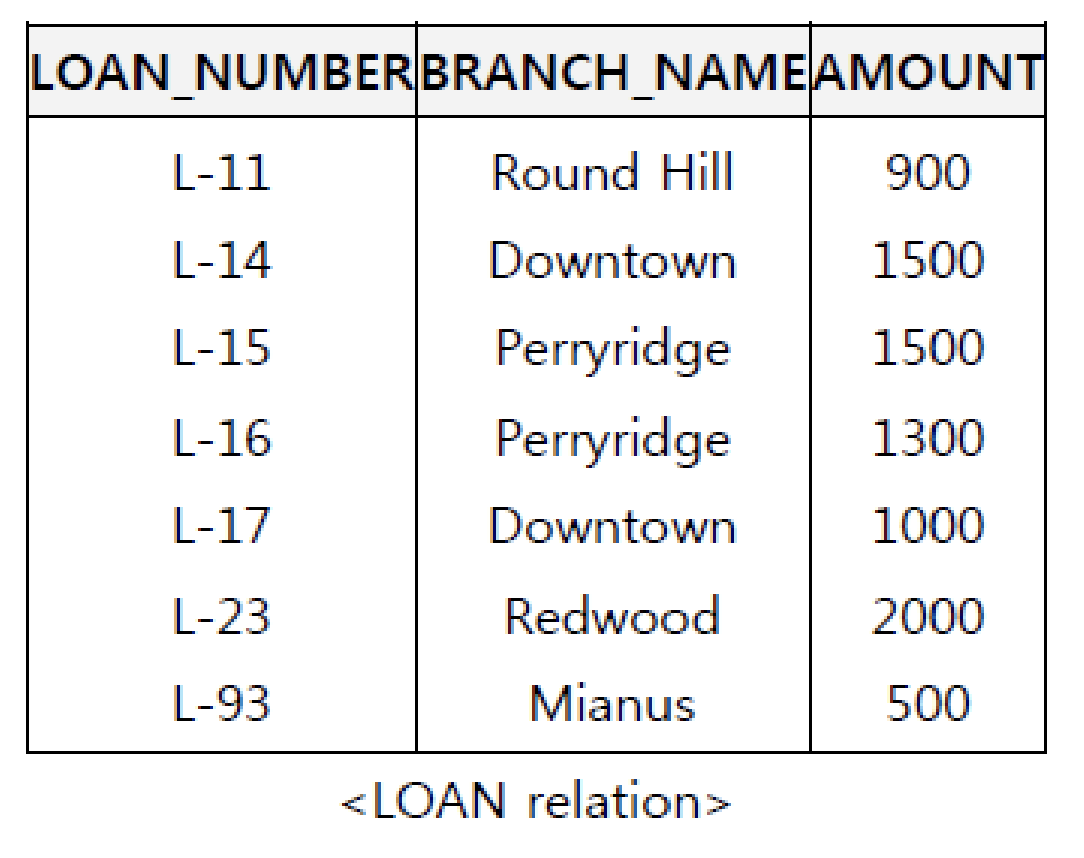
다음의 질의를 작성하고 질의문과 결과를 보고서에 기입하시오.

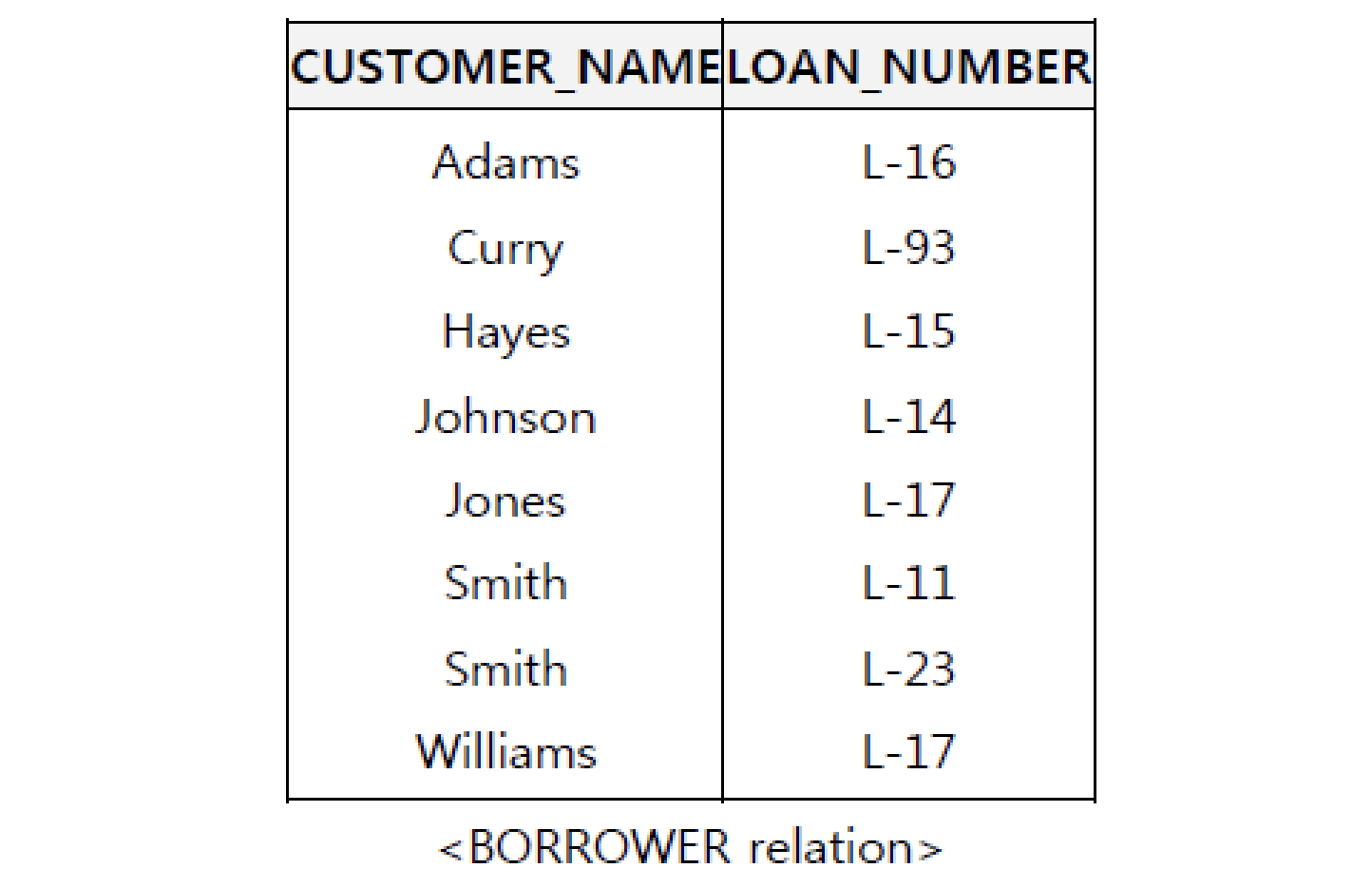
1) 다음 데이터를 입력하시오 (insert)

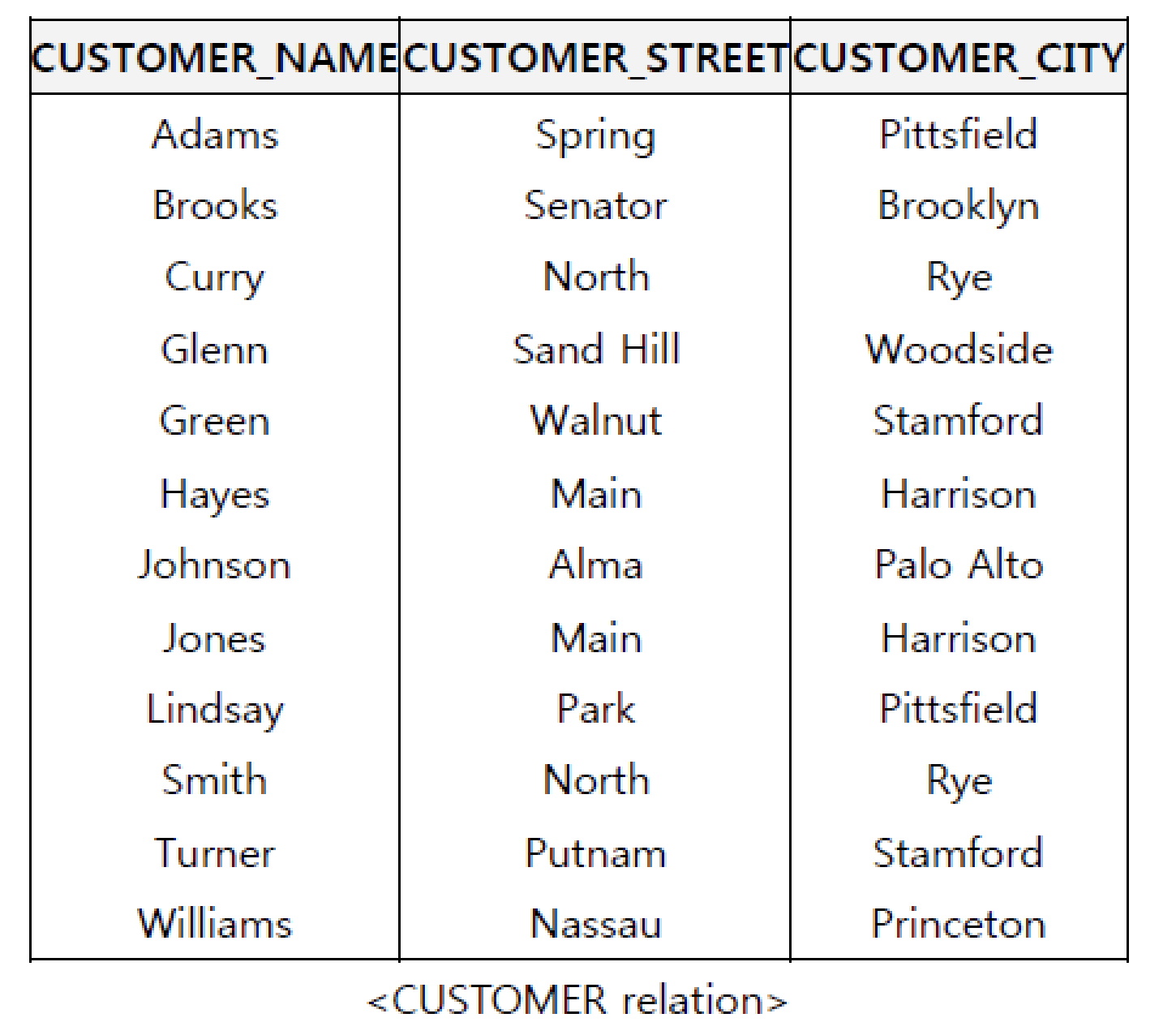












Insert SQL 문

// BRANCH Relation

insert into branch VALUES ('Brighton', 'Brooklyn', 7100000);

insert into branch VALUES ('Downtown', 'Brooklyn', 9000000);

insert into branch VALUES ('Mianus', 'Horseneck', 400000);

insert into branch VALUES ('North Town', 'Rye', 3700000);

insert into branch VALUES ('Perryridge', 'Horseneck', 1700000);

insert into branch VALUES ('Pownal', 'Benninghton', 300000);

insert into branch VALUES ('Redwood', 'Palo Alto', 2100000);

insert into branch VALUES ('Round Hill', 'Horseneck', 8000000);

// ACCOUNT Relation

insert into account VALUES ('A-101', 500, 'Downtown');

insert into account VALUES ('A-102', 400, 'Perryridge');

insert into account VALUES ('A-201', 900, 'Brighton');

insert into account VALUES ('A-215', 700, 'Mianus');

insert into account VALUES ('A-217', 750, 'Brighton');

insert into account VALUES ('A-222', 700, 'Redwood');

insert into account VALUES ('A-305', 350, 'Round Hill');

// LOAN Relation

insert into loan VALUES ('L-11', 900, 'Round Hill');

insert into loan VALUES ('L-14', 1500, 'Downtown');

insert into loan VALUES ('L-15', 1500, 'Perryridge');

insert into loan VALUES ('L-16', 1300, 'Perryridge');

insert into loan VALUES ('L-17', 1000, 'Downtown');

insert into loan VALUES ('L-23', 2000, 'Redwood');

insert into loan VALUES ('L-93', 500, 'Mianus');

// CUSTOMER Relation

insert into customer VALUES ('Adams', 'Spring', 'Pittsfield');

insert into customer VALUES ('Brooks', 'Senator', 'Brooklyn');

insert into customer VALUES ('Curry', 'North', 'Rye');

insert into customer VALUES ('Glenn', 'Sand Hill', 'Woodside');

insert into customer VALUES ('Green', 'Walnut', 'Stamford');

insert into customer VALUES ('Hayes', 'Main', 'Harrison');

insert into customer VALUES ('Johnson', 'Alma', 'Palo Alto');

insert into customer VALUES ('Jones', 'Main', 'Harrison');

insert into customer VALUES ('Lindsay', 'Park', 'Pittsfield');

insert into customer VALUES ('Smith', 'North', 'Rye');

insert into customer VALUES ('Turner', 'Putnam', 'Stamford');

insert into customer VALUES ('Williams', 'Nassau', 'Princeton');

// DEPOSITOR Relation

insert into depositor VALUES ( 'A-102','Hayes');

insert into depositor VALUES ( 'A-101','Johnson');

insert into depositor VALUES ( 'A-201','Johnson');

insert into depositor VALUES ( 'A-217','Jones');

insert into depositor VALUES ( 'A-222','Lindsay');

insert into depositor VALUES ( 'A-215','Smith');

insert into depositor VALUES ( 'A-305','Turner');

// BORROWER Relation

insert into borrower VALUES ('Adams', 'L-16');

insert into borrower VALUES ('Curry', 'L-93');

insert into borrower VALUES ('Hayes', 'L-15');

insert into borrower VALUES ('Johnson', 'L-14');

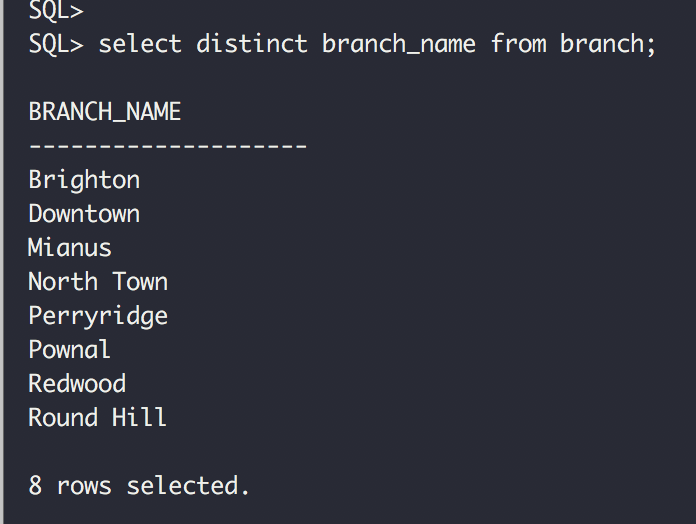
insert into borrower VALUES ('Jones', 'L-17');

insert into borrower VALUES ('Smith', 'L-11');

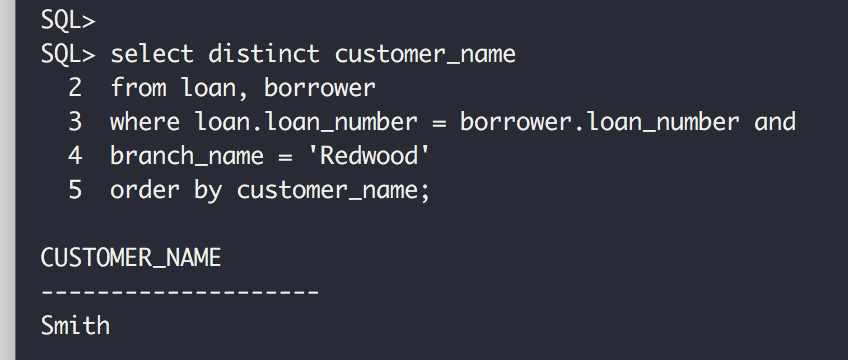
insert into borrower VALUES ('Smith', 'L-23');

insert into borrower VALUES ('Williams', 'L-17');

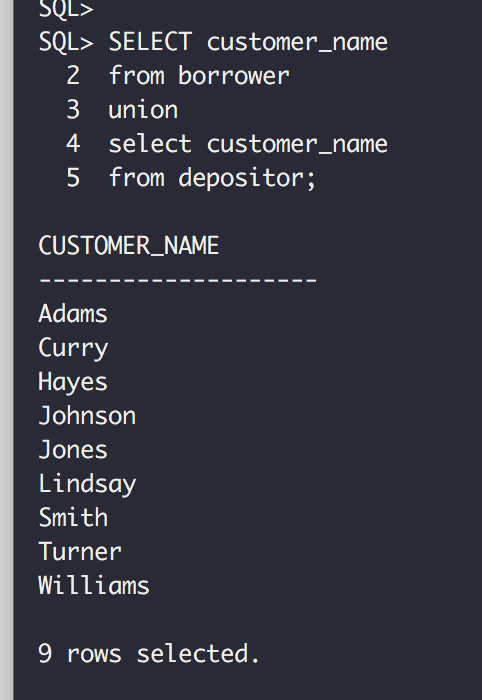
2) 중복되지 않은 모든 지점들의 이름을 구하라.



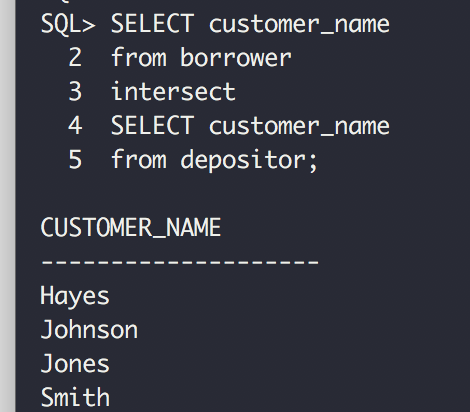
3) Redwood 지점의 대출을 가진 모든 고객들을 알파벳 순서로 나열하라.



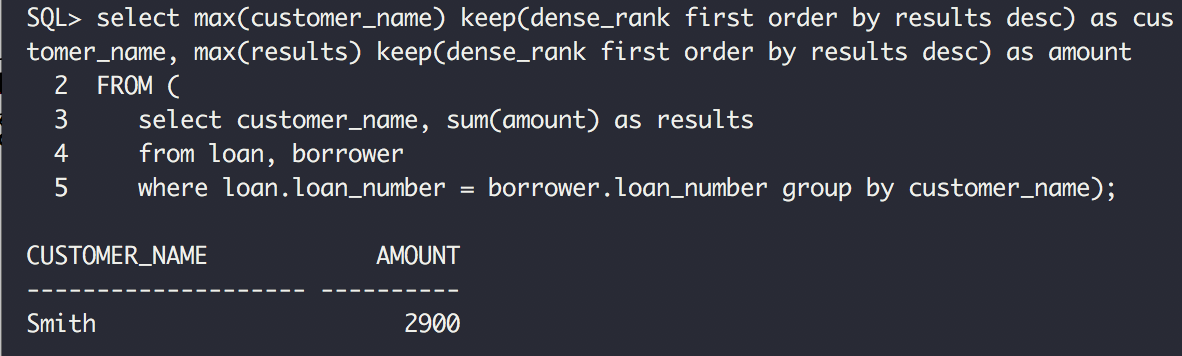
4) 은행에서 대출, 계좌 혹은 둘 다를 가진 모든 고객을 나열하라.



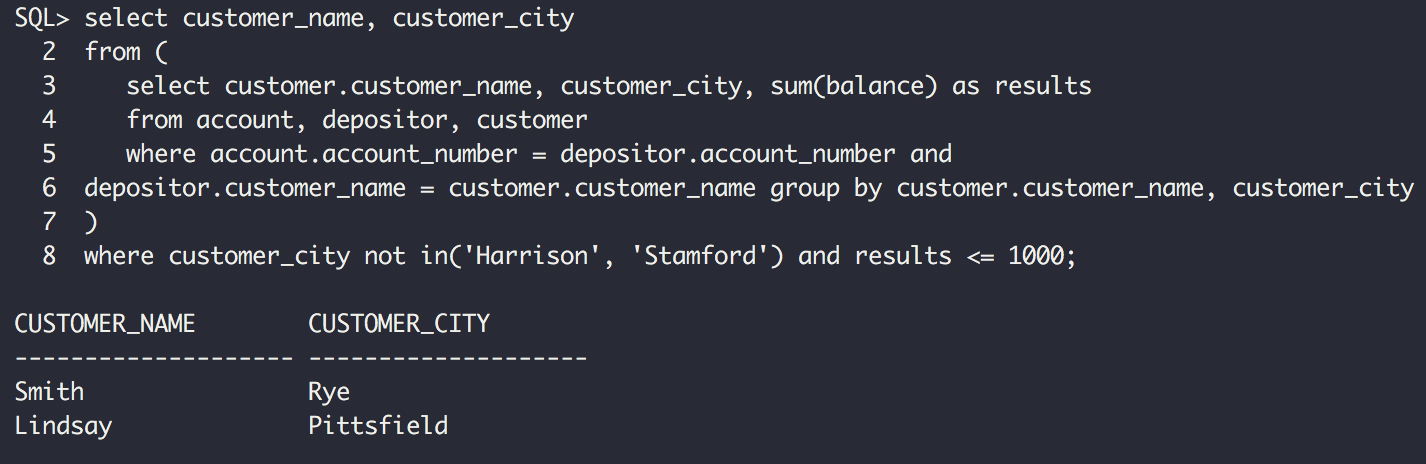
5) 은행에 대출과 계좌 모두를 가진 모든 고객을 나열하라.



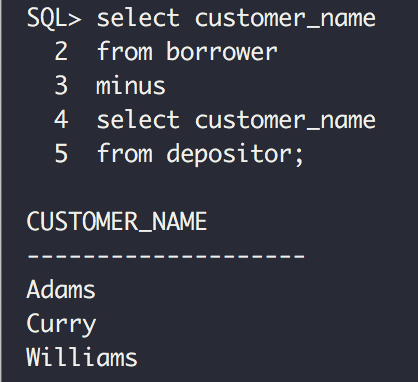
6) 대출 총액이 가장 큰 고객의 이름과 대출 총액을 구하여라.



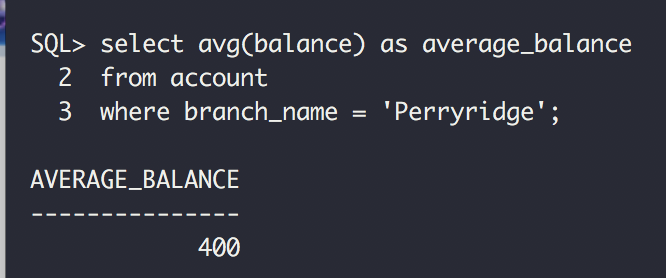
7) Harrison과 Stamford에 살지 않으면서 계좌에 잔고가 1000이하 있는 고객의 이름과 고객이 사는 도시를 구하라.



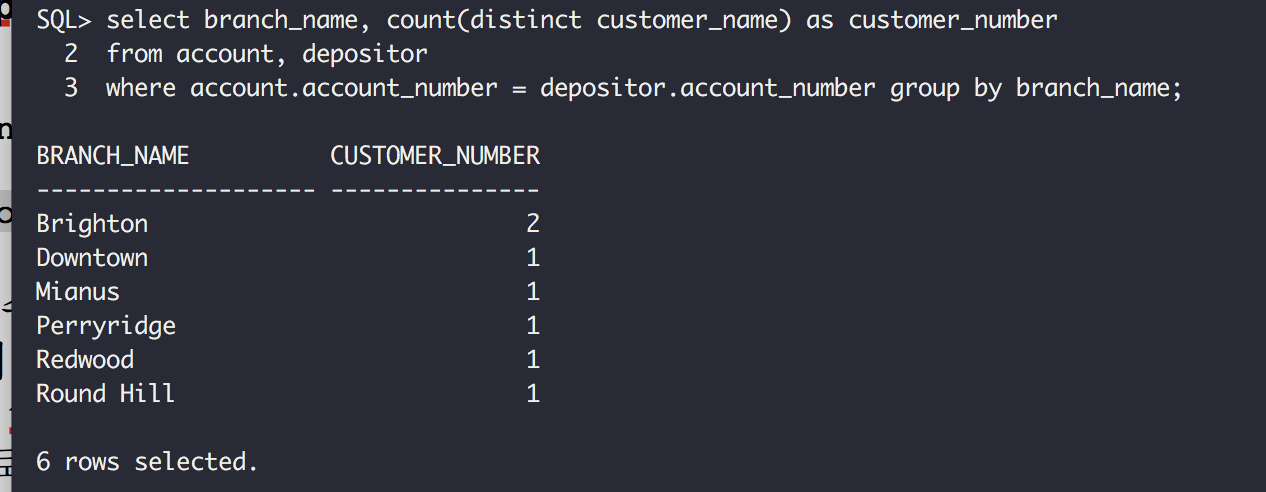
8) 은행에 계좌는 가지고 있지 않지만 대출은 가지고 있는 모든 고객들을 나열하라.



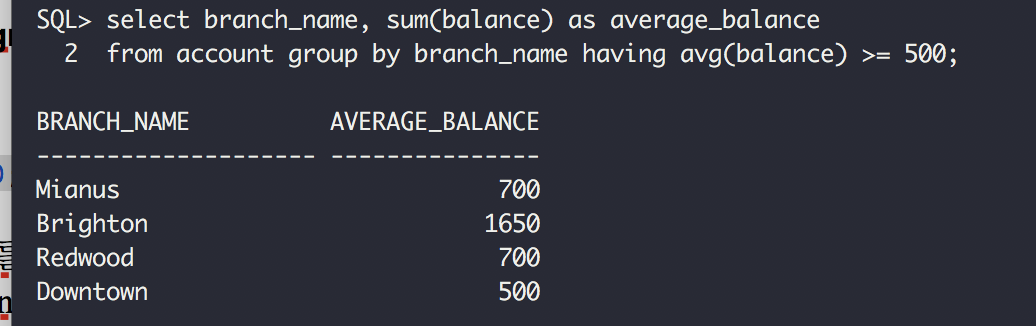
9) Perryridge 지점에서 계좌의 평균 잔고를 구하여라.



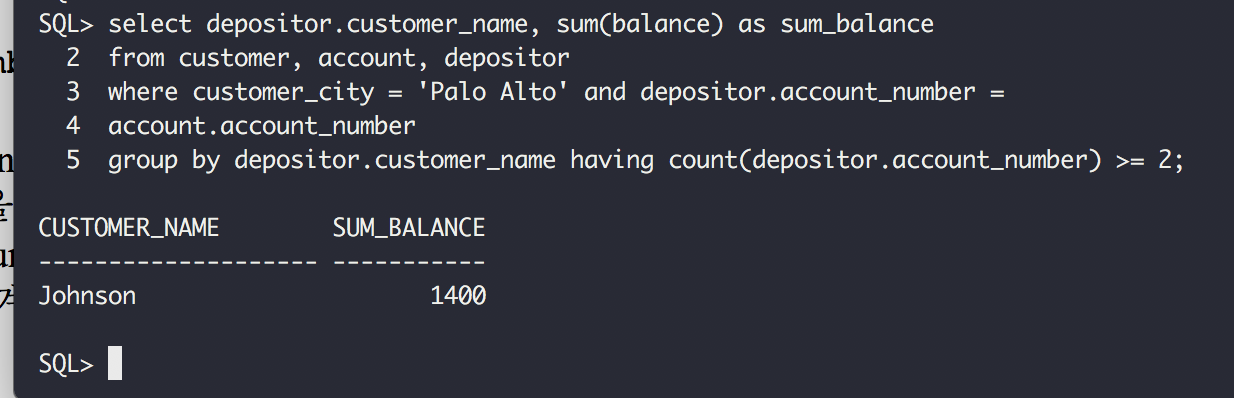
10) 각 지점의 예금자들의 수를 구하라.



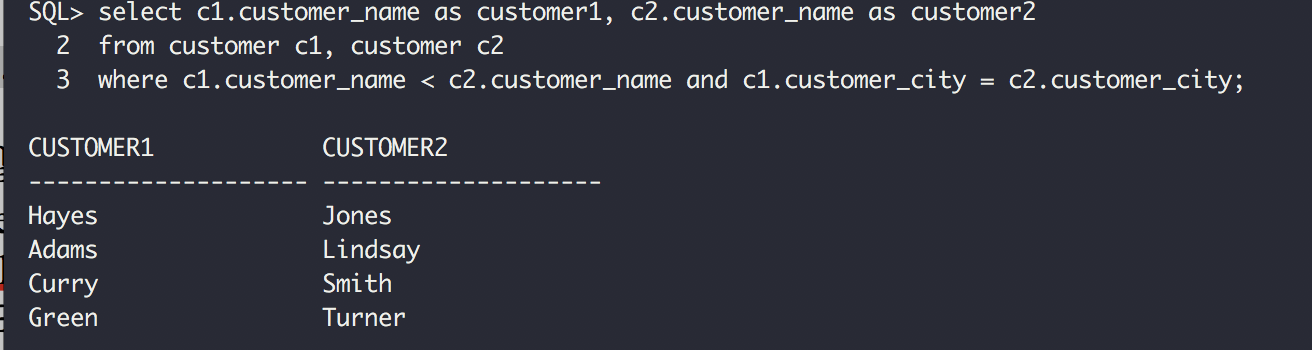
11) 평균 잔고가 $500 이상인 지점 이름과 총 잔고를 나열하라.



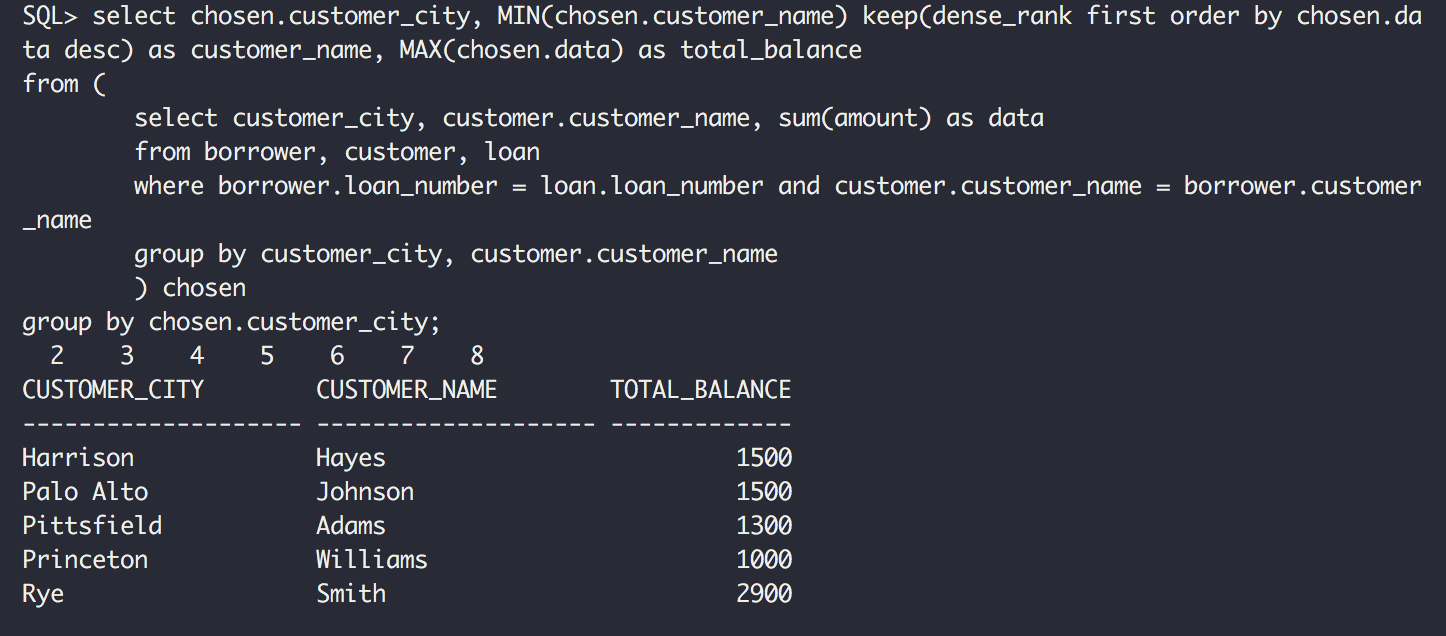
12) Palo Alto에 살고 최소한 두 개의 계좌를 가진 각각의 고객들의 이름과 잔고의 합을 구하라.



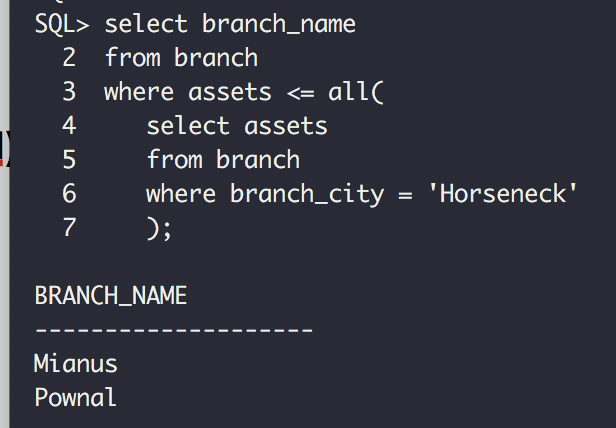
13) 같은 도시에 사는 고객의 이름의 쌍을 구하여라.



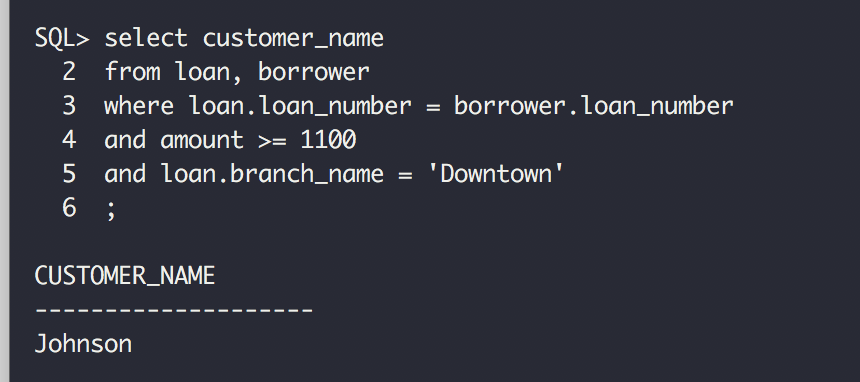
14) 각 도시 별로 가장 낮은 대출 총액을 가지고 있는 고객의 이름과 대출 총액을 구하여라. 단, 대출을 가 진 고객이 살지 않는 도시는 표시하지 않는다.



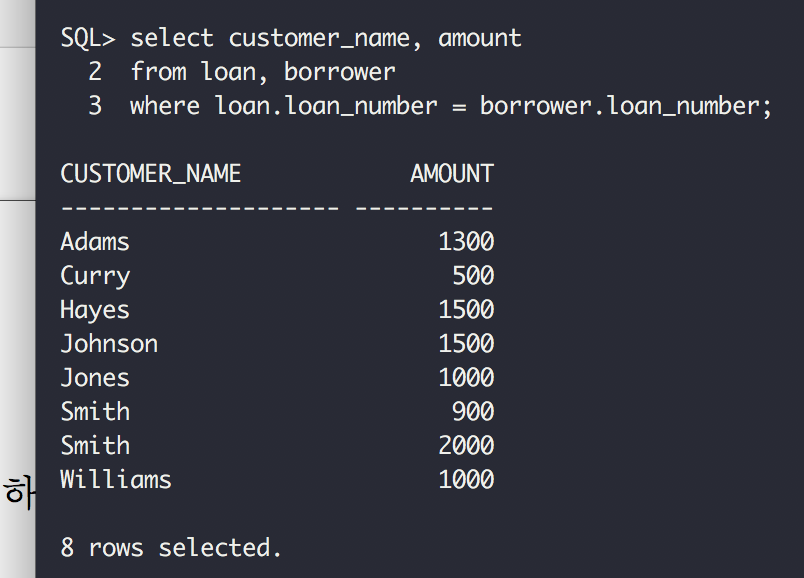
15) Horseneck에 있는 각 지점보다 작거나 같은 자산 값을 갖는 모든 지점들의 이름을 나열하라.



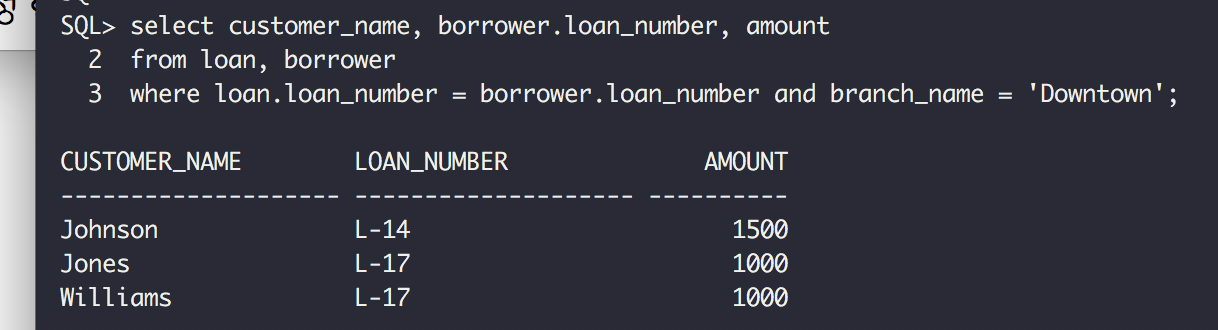
16) Downtown 지점에서 $1100 이상의 대출 총액을 지닌 고객들을 전부 구하라.



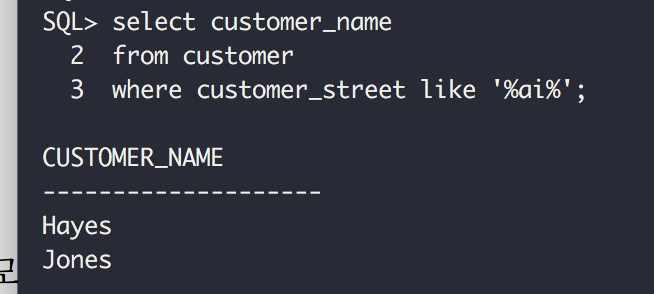
17) 은행에 대출을 가지고 있는 모든 고객들에 대해 그들의 이름과 대출 총액을 구하라.



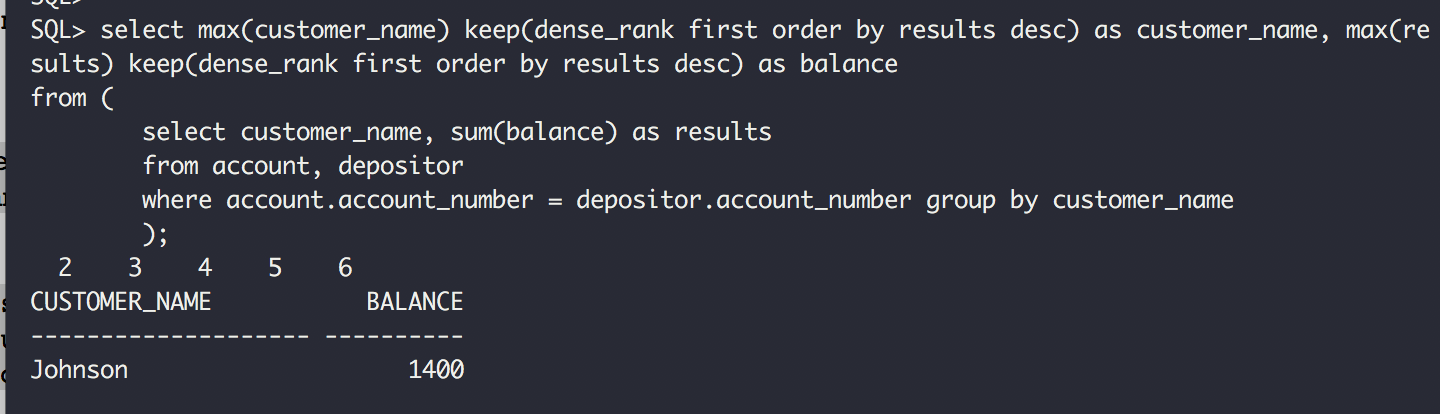
18) Downtown 지점의 모든 대출에 대해여 고객의 이름과 대출 번호, 대출액을 구하라.



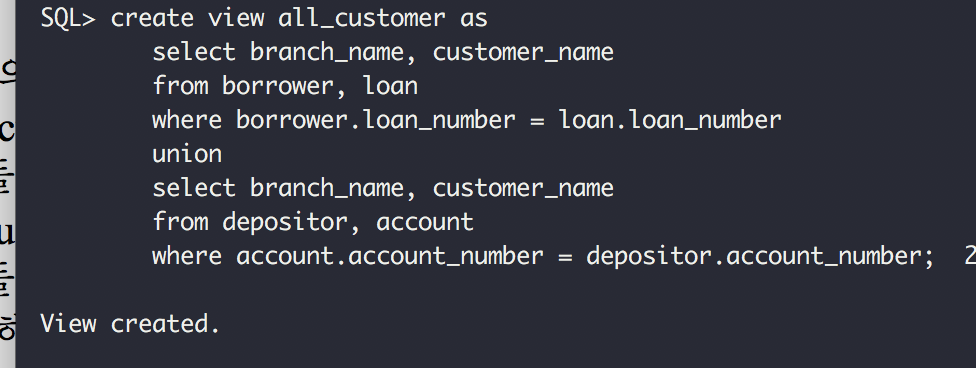
19) 이름에 'ai'이라는 부분 문자열이 포함된 거리에 살고 있는 모든 고객들의 이름을 구하여라.



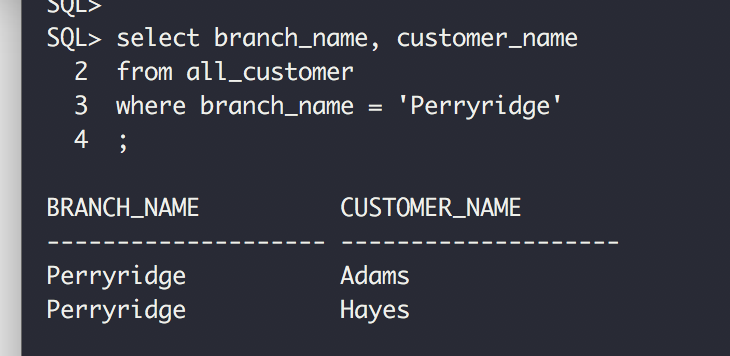
20) 가장 높은 평균 잔고를 가진 고객의 이름과 총잔고를 구하라.



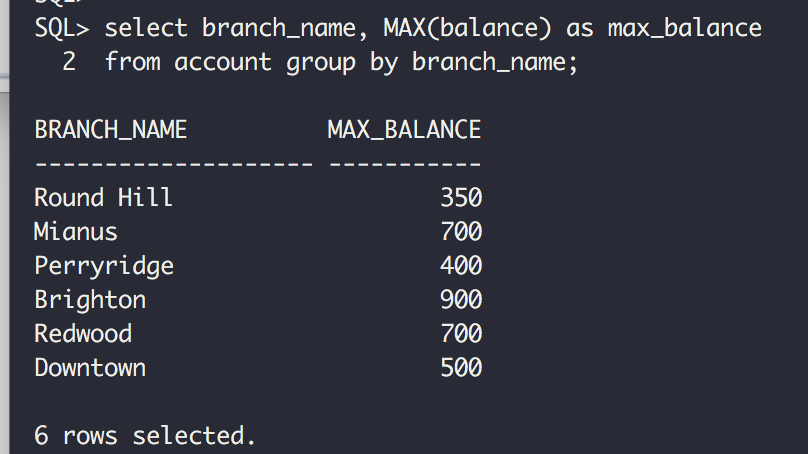
21) 지점 이름과 그 지점에 계좌나 대출 둘 중 하나를 가진 고객 이름으로 구성된 View를 작성하라. (단 View의 이름은 all\_customer이다.)



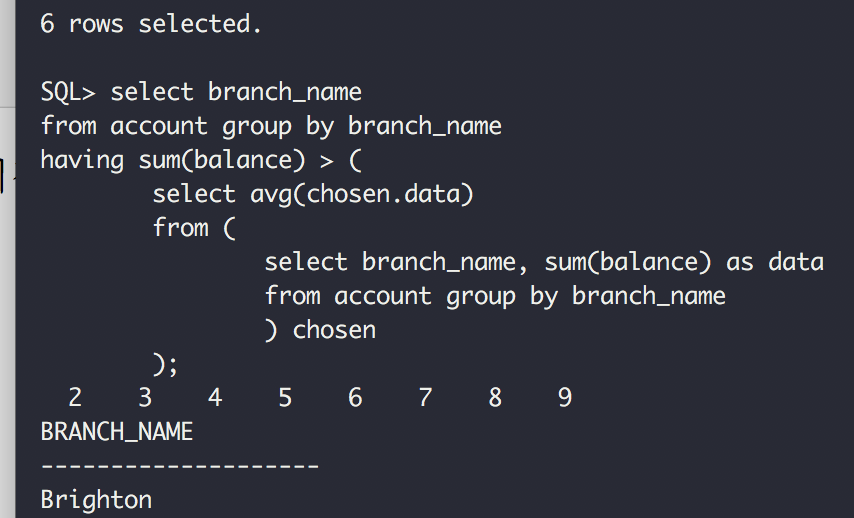
22) 21에서 생성된 View를 이용하여 Perryridge 지점의 모든 고객 이름을 나열하라.



23) 각 지점에서 총 잔고의 최대값을 나열하라.



24) 모든 지점의 총 계좌 예금의 평균보다 많은 총 계좌 예금을 갖는 모든 지점을 나열하라.



25) 평균 대출 총액보다 적은 대출 총액을 가지고 있는 고객의 이름과 대출 총액을 구하라.

