

Exercise 1

1. Use double negation - all customers who have a bank account minus those that also have a loan at the same bank (here join is natural join)

```
r1 = project cstrmr_name, branch_name ( depositor join account )
```

```
r2 = project cstrmr_name, branch_name ( borrower join loan )
```

```
r3 = project cstrmr_name (r1) - project cstrmr_name (r1 join r2)
```

2. Exactly one account is expressed as at least one minus at least two.

We take those with exactly one account, exactly one loan and take the intersection.

```
d1[c1,a1] = depositor
```

```
at_least_one_account = project cstrmr_name (depositor)
```

```
at_least_two_accounts = project cstrmr_name (join cstrmr_name = c1, acct_num != a1 (depositor, d1))
```

```
exactly_one_account = at_least_one_account - at_least_two_accounts
```

```
b1[c1,l1] = borrower
```

```
at_least_one_loan = project cstrmr_name (borrower)
```

```
at_least_two_loans = project cstrmr_name (join cstrmr_name = c1, loan_num != l1 (borrower, b1))
```

```
exactly_one_loan = at_least_one_loan - at_least_two_loans
```

```
r = exactly_one_account intersection exactly_one_loan
```

3. Note that the amount loaned is always positive, while the balance might be negative.

```
r1 = project cstrmr_name, balance (account join depositor)
```

```
r2 = project cstrmr_name, amount (loan join borrower)
```

```
r3 = project cstrmr_name (customer) - project cstrmr_name (r2)
```

```
r4[cstrmr_name, amount] = project cstrmr_name, 0 (r3)
```

```
r5 = r3 union r4
```

```
r = project cstrmr_name (select balance >= amount (r5 join r1))
```

4. The branch(es) with the lowest assets is the one for which no other branch has a lower asset

Using a self-join, we obtain the branches for which at least one branch has a lower asset.

The answer is the set difference between all branches and these branches.

```
b1[bn1,bc1,a1] = branch
```

```
r1 = project bn1 (join bn1 = branch_name and assets < a1 (b1, branch))
```

```
r2 = project branch_name (branch)
```

```
r = r2 - r1
```

Exercise 2

1. Find the name of a city where at least 5 hotels are located

```
project h_zip1 (
```

```
hotel[h_hno1, h_name1, h_zip1, h_address1] join
```

```
hotel[h_hno2, h_name2, h_zip2, h_address2] join
```

```
hotel[h_hno3, h_name3, h_zip3, h_address3] join
```

```
hotel[h_hno4, h_name4, h_zip4, h_address4] join
```

```
hotel[h_hno5, h_name5, h_zip5, h_address5] join
```

```
on h_zip1=h_zip2 and h_zip2=h_zip3 and h_zip3=h_zip4 and h_zip4=h_zip5
```

*and not (h_hno1=h_hno2 and h_hno2=h_hno3 and h_hno3=h_hno4 and
h_hno4=h_hno5)
)*

2. Find the name of all the hotels located in Los Angeles

project name (hotel join city on hotel.zip=city.zip and city.name = "Los Angeles")

3. Find the name of the hotels that have double rooms

project name (select room.type="double" (hotel join room on hotel.hno = room.hno))

4. Find the name of the customers that have had reservations in all the hotels from Los Angeles

project name (project cno,hno (reservation) / project hno (hotel join city on hotel.zip=city.zip and city.name = "Los Angeles"))[d_cno] join customer on d_cno = cno)

5. Find the hotel with the most expensive room

project hno (hotel)

minus

project hno2 (room[hno1, type1, free1, price1] join room[hno2, type2, free2, price2] on price1>price2)