# Hands-on 6: Databases

Michael Plasmeier

1. theplaz=> create table accounts (username varchar(8), fullname varchar(128), balance int);

CREATE TABLE

theplaz=> insert into accounts values ('jones', 'Alice Jones', 82);

INSERT 0 1

theplaz=> insert into accounts values ('bitdiddl', 'Ben Bitdiddle', 65);

INSERT 0 1

theplaz=> insert into accounts values ('mike', 'Michael Dole', 73);

INSERT 0 1

theplaz=> insert into accounts values ('alyssa', 'Alyssa P. Hacker', 79);

INSERT 0 1

1. theplaz=> \d accounts

Table "public.accounts"

Column | Type | Modifiers

----------+------------------------+-----------

username | character varying(8) |

fullname | character varying(128) |

balance | integer |

1. theplaz=> select username, fullname from accounts;

username | fullname

----------+------------------

jones | Alice Jones

bitdiddl | Ben Bitdiddle

mike | Michael Dole

alyssa | Alyssa P. Hacker

(4 rows)

theplaz=> select sum(balance) from accounts;

sum

-----

299

(1 row)

theplaz=> select fullname from accounts where balance > 75;

fullname

------------------

Alice Jones

Alyssa P. Hacker

(2 rows)

theplaz=> select fullname from accounts where username = 'bitdiddl';

fullname

---------------

Ben Bitdiddle

(1 row)

theplaz=> select avg(balance) from accounts where balance > 70; avg

---------------------

78.0000000000000000

(1 row)

1. theplaz=> update accounts set balance = balance-10 where username = 'jones';

UPDATE 1

theplaz=> update accounts set balance = balance+10 where username = 'mike';

UPDATE 1

1. They are different

theplaz=> select \* from accounts;

username | fullname | balance

----------+------------------+---------

bitdiddl | Ben Bitdiddle | 65

alyssa | Alyssa P. Hacker | 79

jones | Alice Jones | 72

mike | Michael Dole | 83

(4 rows)

theplaz=> insert into accounts values ('chuck', 'Charlie Robinson', 55);

INSERT 0 1

theplaz=> select \* from accounts;

username | fullname | balance

----------+------------------+---------

bitdiddl | Ben Bitdiddle | 65

alyssa | Alyssa P. Hacker | 79

jones | Alice Jones | 72

mike | Michael Dole | 83

chuck | Charlie Robinson | 55

(5 rows)

1. No, it does not include chuck. It only includes the rows that were in place on BEGIN.
2. Yes, the output now includes chuck, since a new transaction has started.
3. The command does not return anything nor create a new command line. This is different than the example in the P-Set where it displays UPDATE 1.
4. The other transaction now completes and displays UPDATE 1.
5. Mike’s balance is now 73, which is 10 less than before.
6. No, the results do not change until a transaction is committed.
7. The transaction becomes visible when COMMIT is issued;
8. SERIALIZABLE has a performance implication. All of that locking makes it hard to get stuff done. In addition, having SERIALIZABLE correctness may not be important for the particular use. Thus, Postgres sets a default that is good for performance, and allows users to change it if they want.