**Entities (tables)**

bank

employee

customer

account

transaction

**Relationships**

1 bank

1 employee

1 bank has 1 employee

1 bank has 0 or many customers

1 customer has 1 or many accounts

1 account has 1 or many transactions (first one is Open)

1 employee has 0 or many accounts

**bank**

bank\_id PK identity

bank\_name varchar

bank\_address varchar

**employee**

emp\_id PK identity

emp\_bank\_id FK int4

emp\_fname varchar

emp\_lname varchar

emp\_log\_id varchar

emp\_log\_pw varchar

**customer**

cust\_id PK identity

cust\_bank\_id FK int4

cust\_fname varchar

cust\_lname varchar

cust\_log\_id varchar

cust\_log\_pw varchar

**account**

acct\_id PK identity

acct\_owner\_type PK varchar (C=customer or E=employee)

acct\_bank\_id FK int4

acct\_cust\_own\_id FK int4

acct\_emp\_own\_id FK int4

acct\_type varchar not null (C=checking, S=savings, L=loan, CC=credit card)

acct\_approved Boolean (default false)

acct\_approving\_emp\_id FK int4

acct\_approval\_date timestamptz

acct\_initial\_deposit\_amt decimal not null

acct\_current\_bal not null

**transaction**

tran\_id PK identity

tran\_acct\_id FK int4

tran\_bank\_id FK int4

tran\_date timestamptz not null

tran\_type varchar not null (O=open, D=deposit, W=withdraw, TT=transfer\_to, RF=receive\_transfer\_from)

tran\_transfer\_to\_acct\_id FK int4

tran\_transfer\_from\_acct\_id FK int4

tran\_transfer\_from\_accepted Boolean

tran\_amt decimal

**DDL**

--DROP SCHEMA bank;

--CREATE SCHEMA bank\_schema AUTHORIZATION postgres;

DROP TABLE IF EXISTS bank.transaction;

DROP TABLE IF EXISTS bank.account;

DROP TABLE IF EXISTS bank.customer CASCADE;

DROP TABLE IF EXISTS bank.employee CASCADE;

DROP TABLE IF EXISTS bank.bank CASCADE;

--DROP TABLE IF EXISTS bank.bank CASCADE;

CREATE TABLE bank.bank (

bank\_id serial NOT NULL,

bank\_name varchar(50) NOT NULL,

bank\_address varchar(50) NOT NULL,

CONSTRAINT bank\_pkey PRIMARY KEY (bank\_id)

);

--DROP TABLE IF EXISTS bank.employee CASCADE;

CREATE TABLE bank.employee (

emp\_id serial NOT NULL,

emp\_bank\_id int4 NOT NULL,

emp\_fname varchar(50) NOT NULL,

emp\_lname varchar(50) NOT NULL,

emp\_log\_id varchar(50) NOT NULL,

emp\_log\_pw varchar(50) NOT NULL,

CONSTRAINT employee\_pkey PRIMARY KEY (emp\_id),

CONSTRAINT fk\_bank FOREIGN KEY(emp\_bank\_id) references bank.bank(bank\_id)

);

--DROP TABLE IF EXISTS bank.customer CASCADE;

CREATE TABLE bank.customer (

cust\_id serial NOT NULL,

cust\_bank\_id int4 NOT NULL,

cust\_fname varchar(50) NOT NULL,

cust\_lname varchar(50) NOT NULL,

cust\_log\_id varchar(50) NOT NULL,

cust\_log\_pw varchar(50) NOT NULL,

CONSTRAINT cust\_pkey PRIMARY KEY (cust\_id),

CONSTRAINT fk\_bank FOREIGN KEY(cust\_bank\_id) references bank.bank(bank\_id)

);

--DROP TABLE IF EXISTS bank.account CASCADE;

--DROP TABLE IF EXISTS bank.account;

CREATE TABLE bank.account (

acct\_id serial NOT NULL,

acct\_owner\_type varchar(1) NOT NULL **CONSTRAINT** owner\_type **CHECK**(acct\_owner\_type **IN** ('C', 'E')),

acct\_bank\_id int4 NOT NULL,

acct\_cust\_own\_id int4,

acct\_emp\_own\_id int4,

acct\_type varchar(1) NOT NULL **CONSTRAINT** type **CHECK**(acct\_type **IN** ('C', 'S', 'L', 'CC')),

acct\_approved bool,

acct\_approving\_emp\_id int4,

acct\_approval\_date timestamp,

acct\_initial\_deposit\_amt decimal NOT NULL,

acct\_current\_bal decimal,

CONSTRAINT acct\_pkey PRIMARY KEY (acct\_id),

CONSTRAINT fk\_bank FOREIGN KEY(acct\_bank\_id) references bank.bank(bank\_id),

CONSTRAINT fk\_cust FOREIGN KEY(acct\_cust\_own\_id) references bank.customer(cust\_id),

CONSTRAINT fk\_emp FOREIGN KEY(acct\_emp\_own\_id) references bank.employee(emp\_id)

);

--DROP TABLE IF EXISTS bank.transaction;

CREATE TABLE bank.transaction (

tran\_id serial NOT NULL,

tran\_acct\_id int4 NOT NULL,

tran\_bank\_id int4 NOT NULL,

tran\_date timestamp DEFAULT CURRENT\_TIMESTAMP NOT NULL,

tran\_type varchar(1) NOT NULL **CONSTRAINT** type **CHECK**(tran\_type **IN** ('O', 'D', 'W', 'TT', 'RF')),

tran\_transfer\_to\_acct\_id int4,

tran\_transfer\_from\_acct\_id int4,

tran\_transfer\_from\_accepted bool DEFAULT false,

tran\_amt decimal,

CONSTRAINT tran\_pkey PRIMARY KEY (tran\_id),

CONSTRAINT fk\_acct FOREIGN KEY(tran\_acct\_id) references bank.account(acct\_id),

CONSTRAINT fk\_bank FOREIGN KEY(tran\_bank\_id) references bank.bank(bank\_id),

CONSTRAINT fk\_xfer\_to FOREIGN KEY(tran\_transfer\_to\_acct\_id) references bank.account(acct\_id),

CONSTRAINT fk\_xfer\_from FOREIGN KEY(tran\_transfer\_from\_acct\_id) references bank.account(acct\_id)

);

**--INITIAL TEST DATA (FK ids increment so they need to be changed before each new test INSERT)**

**INSERT** **INTO** bank.bank

(bank\_name, bank\_address)

**VALUES**

('Bank of America', '123 Main St, Houston, TX 77000');

**INSERT** **INTO** bank.employee

(emp\_bank\_id, emp\_fname, emp\_lname, emp\_log\_id, emp\_log\_pw)

**VALUES**

(1, 'John', 'Banker', 'id', 'pw');

**INSERT** **INTO** bank.customer

(cust\_bank\_id, cust\_fname, cust\_lname, cust\_log\_id, cust\_log\_pw)

**VALUES**

(1, 'Joe', 'Customer', 'id', 'pw');

**INSERT** **INTO** bank.account

(acct\_owner\_type, acct\_bank\_id, acct\_cust\_own\_id, acct\_type, acct\_initial\_deposit\_amt)

**VALUES**

('C', 1, 1, 'C', 123.99);

**INSERT** **INTO** bank.**transaction**

(tran\_acct\_id, tran\_bank\_id, tran\_type, tran\_amt)

**VALUES**

(1, 1, 'D', 321.88);