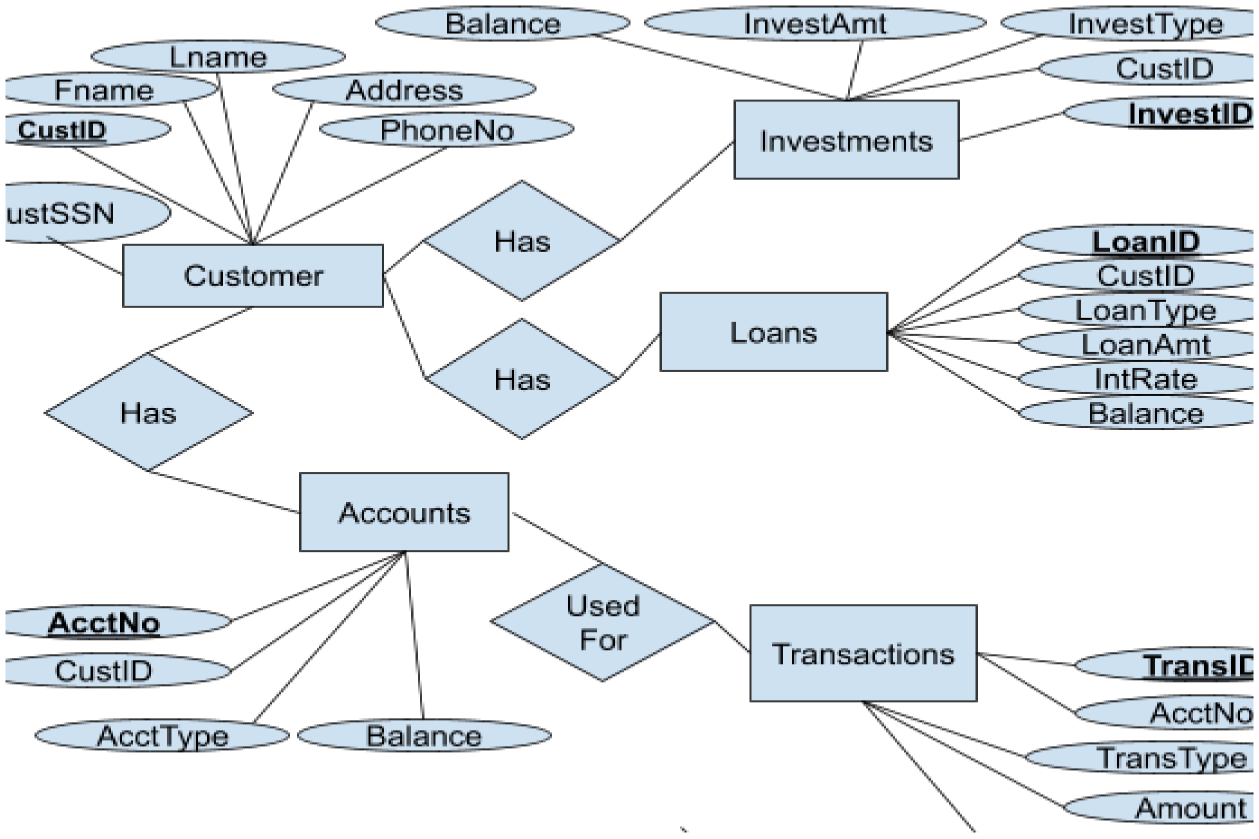
David Wright and Brian Ruehr
Current Trends in Database Tech
Spring 2025
Prof Wu

Final Project Assignment

1. The online banking database will require information to be kept on the customers in the database. Some attributes to be included will be a Customer ID, Customer SSN, name, address, and phone number. It will also contain information on the accounts in the system. The attributes included in this will be the account number, the customer ID, account type, and balance. The database will also include data on investment accounts which will include investment ID, customer ID, investment type (stocks, bonds, mutual funds), investment amount, current value. After that the database will contain information on loans which will contain a Loan ID, customer ID, loan type, loan amount, interest rate, current balance. Lastly the database will contain information on transactions which will contain Transaction ID, account number, transaction type (deposit, withdrawal, transfer), amount, transaction date.

2. Entity-Relationship Diagram



3. Database Schema (Verify that all items are in 3NF):
   Customer(CustID, CustSSN)
   CustomerData(CustSSN, Fname, Lname, Address, PhoneNum)
   Investment(InvestID, CustID, InvestType, InvestAmt, Balance)
   Loan(LoanD, CustID, LoanType, LoanAmt, IntRate, Balance)
   Account(AcctNo, CustID, AcctType, Balance)
   Transaction(TransID, AcctNo, TransType, Amount)

   Foreign Keys:
   Customer.CustSSN → CustomerData.CustSSN
   Invesment.CustID → Customer.CustID
   Loan.CustID → Customer.CustID
   Account.CustID → Customer.CustID
   Transaction.AcctNo → Account.AcctNo

4. This database system will be created using OracleDB.

5.
```
CREATE TABLE CustomerData(

        CustSSN varchar2(9) PRIMARY KEY,

        Fname VARCHAR2(100),

        Name VARCHAR2(100),

        PhoneNum VARCHAR2(10),

        StreetNum VARCHAR2(10),

        StreetName VARCHAR2(100),

        CityName VARCHAR2(100),

        StateAbb VARCHAR2(2),

        ZipCode VARCHAR2(5)

)

CREATE TABLE Customer(

        CustID NUMBER PRIMARY KEY,

        CustSSN VARCHAR2(9)

        Constraint fk_custssn (custSSN) references CustomerData(custSSN)
```

```
)

CREATE TABLE Investment(

        InvestId NUMBER PRIMARY KEY,

        CustID NUMBER,

        InvestType VARCHAR2(100),

        InvestAmt NUMBER(20, 2),

        Balance NUMBER(20, 2)

        CONSTRAINT fk_cust FOREIGN KEY (custID) REFERENCES
CUSTOMER(custSSN)

)


CREATE TABLE Loan (

        LoanID NUMBER PRIMARY KEY,

        CustID NUMBER,

        LoanType VARCHAR2(50),

        LoanAmt NUMBER(15, 2),

        IntRate NUMBER(6, 3),

        Balance NUMBER(15, 2),

        CONSTRAINT fk_cust FOREIGN KEY (CustID) REFERENCES
Customer(CustID)


);

CREATE TABLE Account (

        AcctNo NUMBER PRIMARY KEY,

        CustID NUMBER,
        AcctType VARCHAR2(30),

        Balance NUMBER(15, 2),

        CONSTRAINT fk_account_customer FOREIGN KEY (CustID) REFERENCES
Customer(CustID)

);


CREATE TABLE Transaction (

        TransID NUMBER PRIMARY KEY,

        AcctNo NUMBER,

        TransType VARCHAR2(20),

        Amount NUMBER(15, 2),

        CONSTRAINT fk_transaction_account FOREIGN KEY (AcctNo) REFERENCES
Account(AcctNo)

);
```

6. **1. Get user information from userID (all users):**

```
SELECT custID, fname, lname, phonenum

FROM customerdata, customer

WHERE customerdata.custssn = customer.custssn;
```

2. Get user info and account information and balance (all account):

```
SELECT c.custID, d.fname, d.lname, a.acctno, a.balance

FROM customer c

inner join customerdata d on c.custssn = d.custssn

inner join account a on a.custid = c.custid
```

```
order by custid;
```

3. Add interest to a loan account:

```
create or replace procedure add_monthly_interest_loan(
        target_loan_id in loan.loanid%type)
as
        new_balance loan.balance%type;
begin
        update loan
        set balance = balance + ((balance* intrate)/12)
        where loanid = target_loan_id
        returning balance into new_balance;

        dbms_output.put_line('Interest charged to account ' || target_loan_id || ' balance
is: ' || new_balance);

        commit;
end add_monthly_interest_loan;
```

To run:

```
Begin
        add_monthly_interest_loan(<loanid>);
End;
```

4. **Procedure to get total assets for a given user:**

```
create or replace procedure total_user_assets(

        target_cust_id in customer.custid%type

)

is

        account_total number := 0;

        loan_total number := 0;

        invest_total number := 0;

        total_assets number := 0;

begin


        select NVL(sum(balance), 0)

        into account_total

        from account

        where custid = target_cust_id;


        select NVL(sum(balance), 0)

        into invest_total

        from investment

        where custid = target_cust_id;


        select NVL(sum(balance), 0)

        into loan_total

        from loan

        where custid = target_cust_id;
```

```
            total_assets := account_total + invest_total - loan_total;

            dbms_output.put_line('Total assets for user ' || target_cust_id || ' : ' ||
total_assets);

end total_user_assets;
```

To run:

```
begin

            total_user_assets(<custid>);

End;
```

5. Get customer Id and customer SSN for investors with a balance of more than $20,000.

```
SELECT Investment.CustID,  Customer.CustSSN, investment.balance

FROM Investment

JOIN Customer ON Investment.CustID = Customer.CustID

WHERE Investment.Balance > 20000;
```

6. Retrieve the last name, account number, transaction type, and amount for withdrawals made.

```
SELECT

   cd.Lname,

   a.AcctNo,

   t.TransType,

   t.Amount

FROM CustomerData cd

JOIN Customer c ON cd.CustSSN = c.CustSSN

JOIN Account a ON c.CustID = a.CustID

JOIN Transaction t ON a.AcctNo = t.AcctNo
```

```sql
WHERE t.TransType = 'Withdrawal';
```

7.List all customers who have both a loan and an investment.

```sql
SELECT DISTINCT cd.Fname, cd.Lname, c.custssn

FROM CustomerData cd

JOIN Customer c ON cd.CustSSN = c.CustSSN

JOIN Loan l ON c.CustID = l.CustID

JOIN Investment i ON c.CustID = i.CustID;
```

8.Get customer Id and customer SSN customers with a loan balance of less than $20,000.

```sql
SELECT loan.CustID,  Customer.CustSSN, loan.balance

FROM loan

JOIN Customer ON loan.CustID = Customer.CustID

WHERE loan.Balance< 20000;
```