

Internetbanken

Projekt av David Hörnmark, Felix Kozma och Jonathan Ström

Konceptuell modellering

Vi skall skapa en internetbank-tjänst som har två gränssnitt. Ett via webbläsare för kunder och ett via kommandoradsinterface för administratörerna.

Tjänster

Tjänsterna inom internetbanken är förflyttning utav pengar mellan sina egna konton och en swish-tjänst där man kan förflytta pengar till ett konto som tillhör någon annan kund. En vanlig förflyttning utav pengar sker genom att man har loggat in på internetbanken med sitt användar-id och sin pinkod och gör överföringen via formuläret där, en avgift på 3% dras vid varje förflyttning.

En swish-överföring sker genom att man via webbläsaren tar sig till swish-applikationen och skriver in sitt användar-id, sin pinkod, konto från, konto till och summa.

Aktörer

Administratörerna har hand om kunderna och kunderna har in sin tur hand om sina konton. Ett konto kan vara delat mellan andra kunder också, vilket man ser på webbsidan när man är inloggad.

Tillhörande en kund så finns information som kund-id, en personlig kod på fyra siffror, förnamn, efternamn, personnummer, adress, postnummer, postort och telefonnummer.

Tillhörande ett konto så finns information som konto-id, kontonummer och saldo.

Ränta

Räntan räknas på daglig basis. Uträkningen sker genom att man kör en lagrad procedur som tar in följande argument; räntesats, datum. Räntan beräknas genom (räntesats * saldo / 365). Resultatet sparas i en tabell i databasen. Denna tabell sparar data så som räntan, datum och kontonummer.

Loggning

Loggning sker varje gång ett kontos saldo uppdateras. Informationen som sparas, kontonummer, summa och datum, läggs i en separat tabell.

Banken egna konto

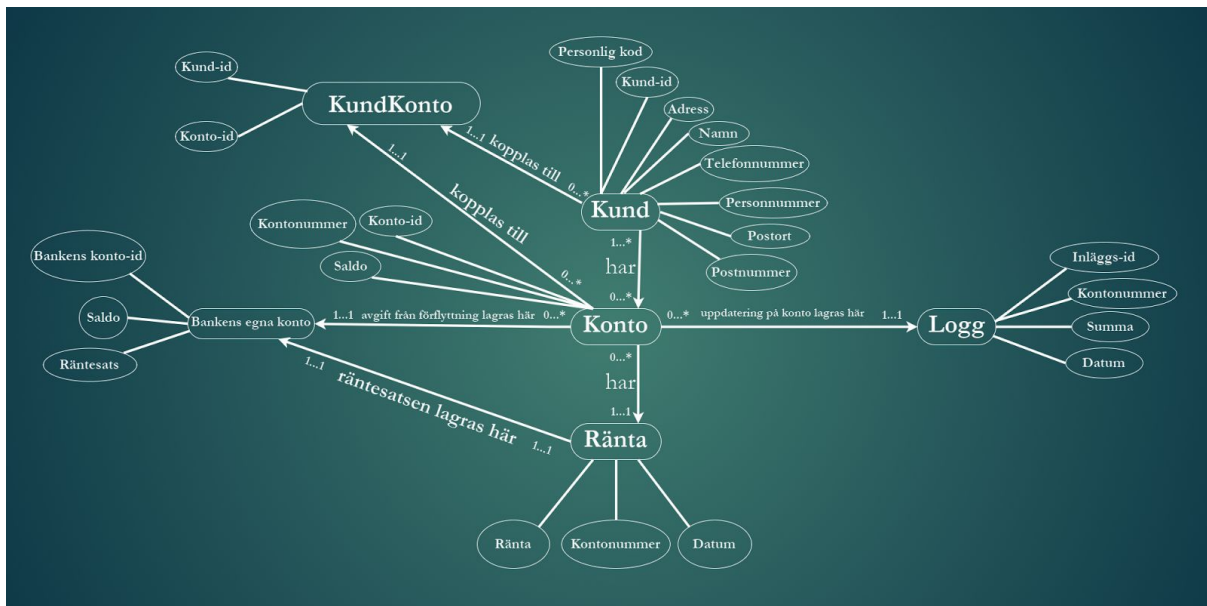
Banken har även en lista över sina egna konton där avgiftspengarna sparas. Informationen som lagras är bankens konto-id, saldo, räntesats.

Entiteter

- Kund
- Konto
- Ränta
- Logg
- Bankens egna konto
- KundKonto

Entiteter	Kund	Konto	Ränta	Logg	B. e. k	KundKonto
Kund		Har				Kopplas till
Konto			Har	Uppdatering på konto lagras här	Avgift från förflyttning lagras här	Kopplas till
KundKonto						
Ränta					Räntesatsen lagras här	
Logg						
B. e. k		Har				

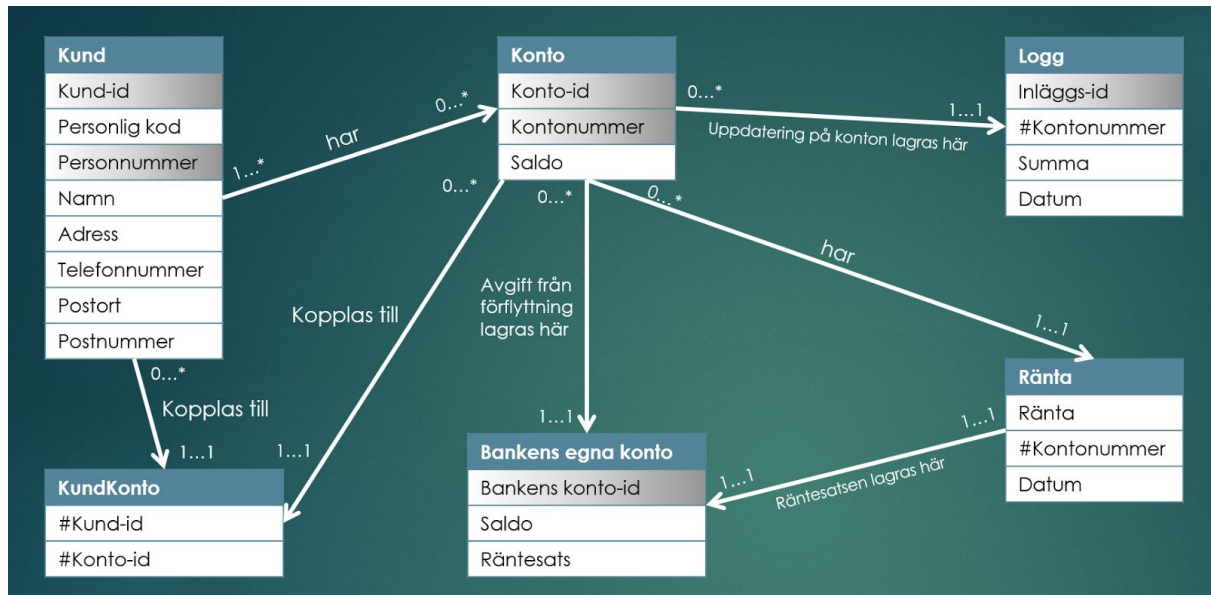
ER-modell – attribut och kandidatnycklar



Relation	Typ
Kund – Konto	1:N
Kund – KundKonto	N:1
Konto – Logg	N:1
Konto – Ränta	N:1
Konto – Bankens egna konto	N:1
Konto – KundKonto	N:1
Ränta – Bankens egna konto	1:1

Logisk modellering

ER-modell med primära och främmande nycklar



Fysisk modellering

```
CREATE TABLE Bank (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    balance INTEGER,  
    interestRate DECIMAL(3, 2)  
);  
  
CREATE TABLE User (  
    userId INT AUTO_INCREMENT PRIMARY KEY,  
    pinCode INT(4) UNSIGNED ZEROFILL NOT NULL,  
    civicNumber CHAR(10) UNIQUE NOT NULL, -- Format: YYMMDDXXXX  
    firstName CHAR(20) NOT NULL,  
    lastName CHAR(20) NOT NULL,  
    street CHAR(20) NOT NULL,  
    zip INT(5) UNSIGNED ZEROFILL NOT NULL,  
    city CHAR(20) NOT NULL,  
    phone CHAR(12) NOT NULL  
);  
  
CREATE TABLE Account (  
    accountId INT AUTO_INCREMENT,  
    accountNr NUMERIC(16, 0),  
    balance INT,  
  
    PRIMARY KEY (accountId, accountNr)  
);  
  
CREATE TABLE UserAccount (  
    userId INT NOT NULL,  
    accountId INT NOT NULL,  
  
    FOREIGN KEY (userId) REFERENCES User(userId),  
    FOREIGN KEY (accountId) REFERENCES Account(accountId)  
);  
  
CREATE TABLE CustomerLog (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    transferDate DATETIME NOT NULL,  
    userId INTEGER NOT NULL,  
    accountNrTo NUMERIC(16, 0) NOT NULL,  
    accountNrFrom NUMERIC(16, 0) NOT NULL,  
    amountSent NUMERIC(16, 3) NOT NULL,  
    transferType CHAR(10),  
  
    FOREIGN KEY (userId) REFERENCES User(userId)  
);  
  
CREATE TABLE InterestLog (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    dateOfCalculation DATETIME NOT NULL,  
    accountNr NUMERIC(16, 0) NOT NULL,  
    interestSum NUMERIC(16, 3) NOT NULL  
);
```


API

Systemets funktionalitet

1. Skapa kund.
2. Skapa konto till kund.
3. Koppla konto till kund.
4. Se vilka konton som en kund delar med en annan kund.
5. Förflytta pengar mellan två av kundens egna konton.
6. Swisha pengar mellan ett av kundens egna konton och ett konto hos en annan kund.
7. Beräkna räntan för alla kunders konton.
8. Beräkna räntan för ett specifikt konto.
9. Logga alla förflyttningar mellan konton.
10. Sätta in pengar på ett konto.
11. Visa alla kunder och deras konton samt saldo för dessa konton.
12. Visa saldot på bankens egna konto där avgifterna från förflyttningarna sparas.
13. Beräkna räntan för en specifik dag.

