

# Blekinge Tekniska Högskola

## Rapport

Databasteknologier för webben, 7,5 hp

2018 VT

## Projekt: Internetbanken

Louise Jakobsson

920624-1243

[louise\\_jakobsson@hotmail.com](mailto:louise_jakobsson@hotmail.com)

# Innehållsförteckning

<b>1. Konceptuell modell: BankBluffOBåg AB</b>	<b>2</b>
1.1 Beskrivning av databasen	2
1.2 Entiteter	2
1.3 Relationer och matris	3
1.3.1 Relationer	3
1.3.2 Matris	3
1.4 ER-diagram	3
1.5 ER-diagram med kardinalitet	4
1.6 ER-diagram med attribut och kandidatnycklar	5
<b>2. Logisk modell: BankBluffOBåg AB</b>	<b>6</b>
2.1 ER-diagram enligt relationsmodellen	6
2.2 ER-diagram med primära och främmande nycklar	7
<b>3. Fysisk modell: BankBluffOBåg AB</b>	<b>8</b>
3.1 SQL DDL för tabellerna	8
3.2 Funktioner som databasen ska stödja (API)	9
<b>Bilaga 1: SQL DDL</b>	<b>9</b>

# 1. Konceptuell modell: BankBluffOBåg AB

## 1.1 Beskrivning av databasen

Vi ska utveckla en webb- och klientbaserad internetbank för företaget BankBluffOBåg. Internetbanken ska möjliggöra administration för kassören, det vill säga skapa nya kunder och konton samt hantera dessa. Ett konto kan vara åtkomligt för flera kunder och en kund kan ha flera konton. Kunderna registreras med id, namn, personnummer, adress, stad och pinkod. Konton skapas med id, kontonummer och eventuell balans. Vidare ska kassören kunna koppla ett existerande konto till en viss kund och vice versa, visa alla kunder med dess konton och kontots balans samt visa alla konton med räntekalkyl för varje konto. Räntekalkylen görs dagligen och utförs manuellt genom en lagrad procedur i databasen.

Internetbankens kunder ska kunna swisha pengar mellan sina egna och andras konton där de får verifiera sig med sina id:n och pinkoder. Varje gång ett kontos balans uppdateras ska det registreras i en logg. Kunder ska även kunna se alla konton de har tillgång till samt om de delas med någon annan kund och isåfall vem. De ska kunna flytta pengar mellan sina egna konton och se räntekalkyl för varje konto.

Internetbanken ska möjliggöra framtagning av rapporter som visar den totala summan av alla bankens konton och innehållet i loggen.

## 1.2 Entiteter

För att ta fram kandidater till tabeller har ett antal entiteter tagits fram utifrån beskrivningen av databasen.

- Kund (accountHolder)
- Konto (account)
- Kontoägare (accountOwner)
- Intressekalkyl (calculationOfInterest)
- Logg (log)

## 1.3 Relationer och matris

För att koppla ihop de olika entiteterna har ett antal relationer mellan dem, som anses relevanta, tagits fram.

### 1.3.1 Relationer

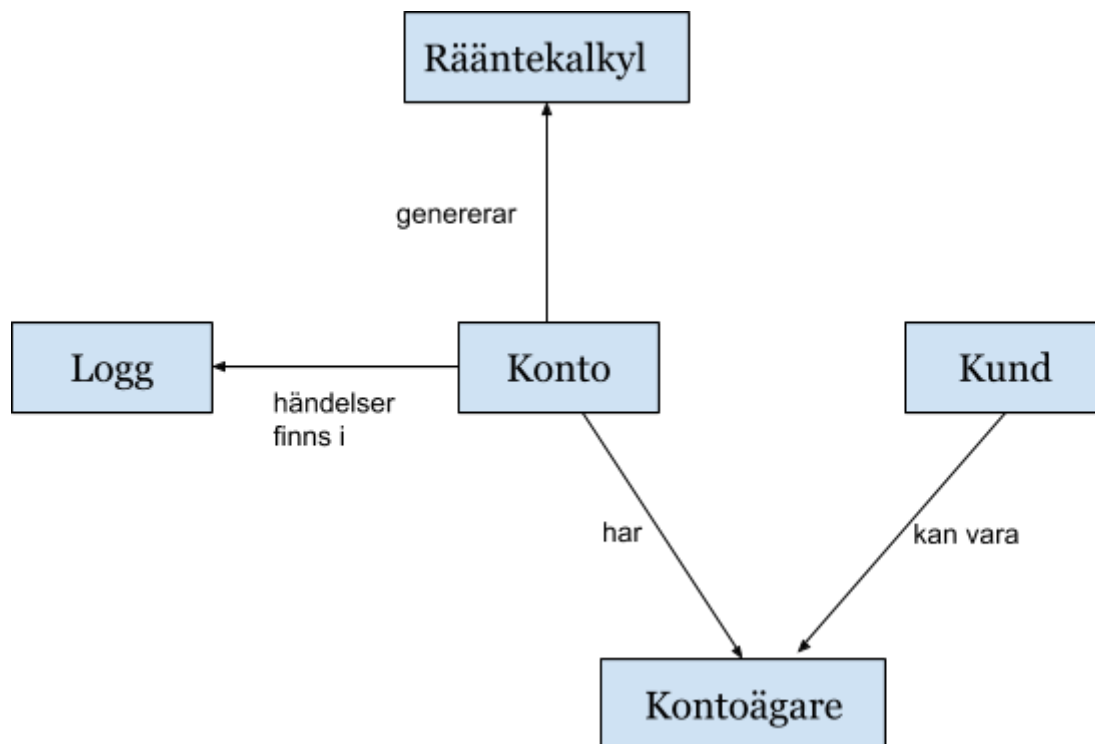
- Varje kund har noll eller flera konton
- Varje konto har en eller flera ägare
- Räntekalkylen som genereras dagligen registreras i en tabell
- Varje förändring i ett konto genererar en händelse i loggen

### 1.3.2 Matris

<i>Entiteter</i>	<b>Kund</b>	<b>Konto</b>	<b>Kontoägare</b>	<b>Räntekalkyl</b>	<b>Logg</b>
<b>Kund</b>		har			
<b>Konto</b>	har				händelse registreras i
<b>Kontoägare</b>		har tillgång till			
<b>Räntekalkyl</b>		hämtar data från			
<b>Logg</b>					

## 1.4 ER-diagram

Utifrån ovan nämnda entiteter och relationer har ett ER-diagram tagits fram.

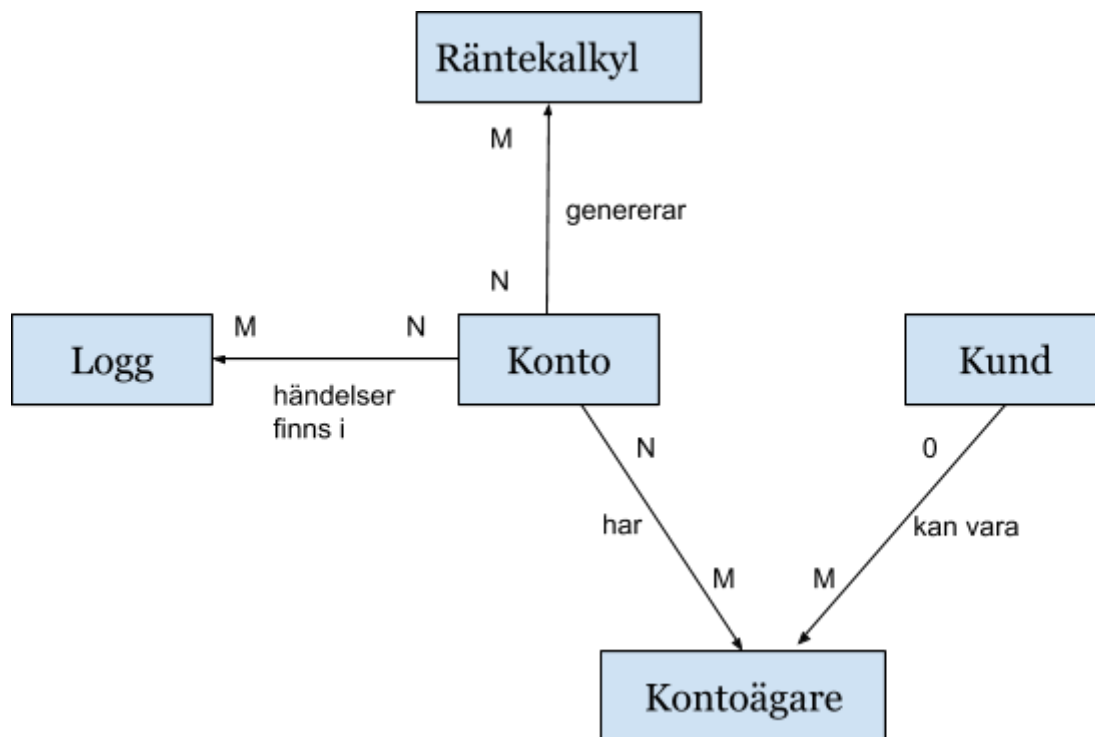


## 1.5 ER-diagram med kardinalitet

Utifrån ER-diagrammet ovan har kardinaliteter tagits fram. Varje konto har en eller flera kontoägare och varje kund kan vara kontoägare till noll eller flera konton. Alla händelser gällande konton registreras i loggen som kan ha flera händelser om flera olika konton. Varje bankkontos summa används för att generera en räntekalkyl.

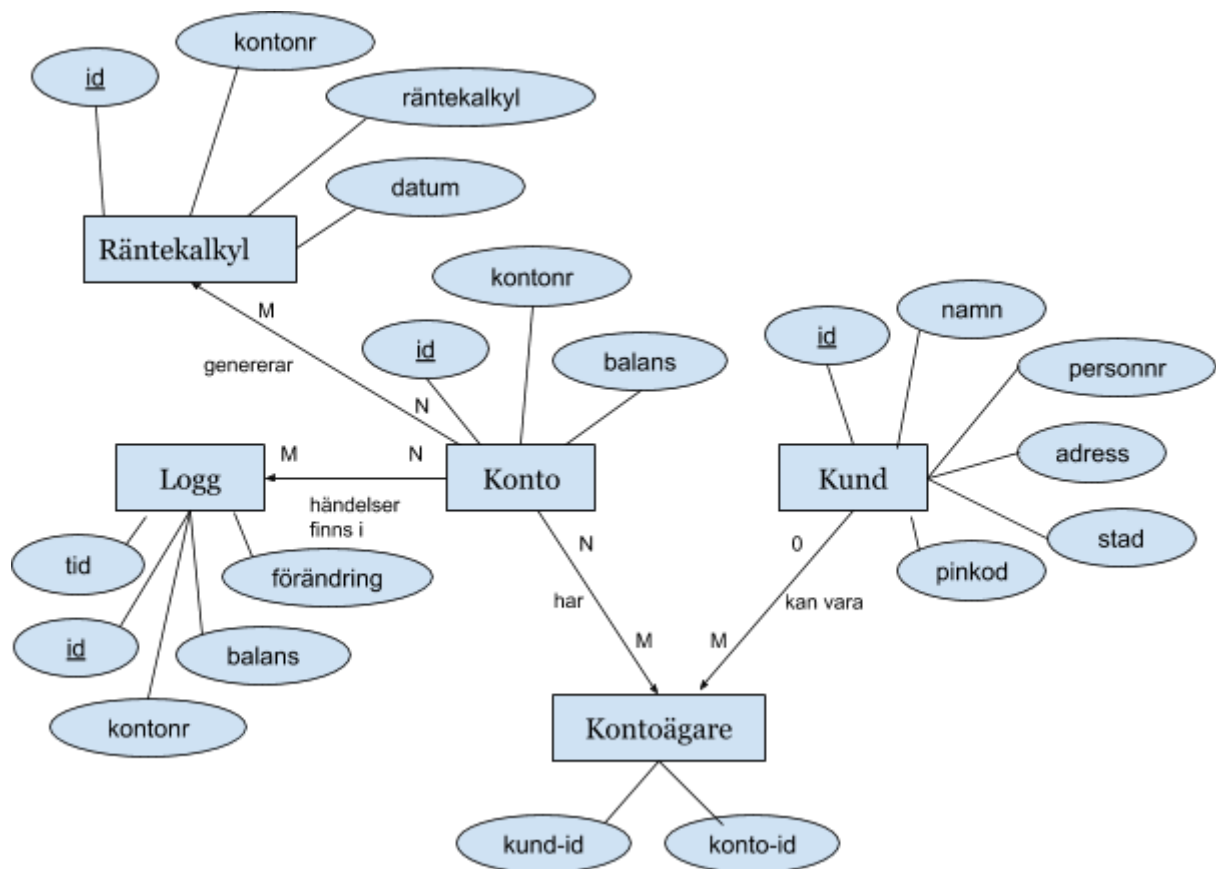
Relation	Typ
Konto - Kontoägare	N:M
Konto - Logg	N:M
Konto - Räntekalkyl	N:M
Kontoägare - Kund	0:M

Nedan visas ER-diagrammet uppdaterat med kardinaliteter.



## 1.6 ER-diagram med attribut och kandidatnycklar

För varje entitet i ett ER-diagram finns ett eller flera attribut. Något av dessa attribut kan sedan komma att bli en nyckel. Utifrån databasbeskrivningen har ER-diagrammet uppdaterats med attribut.

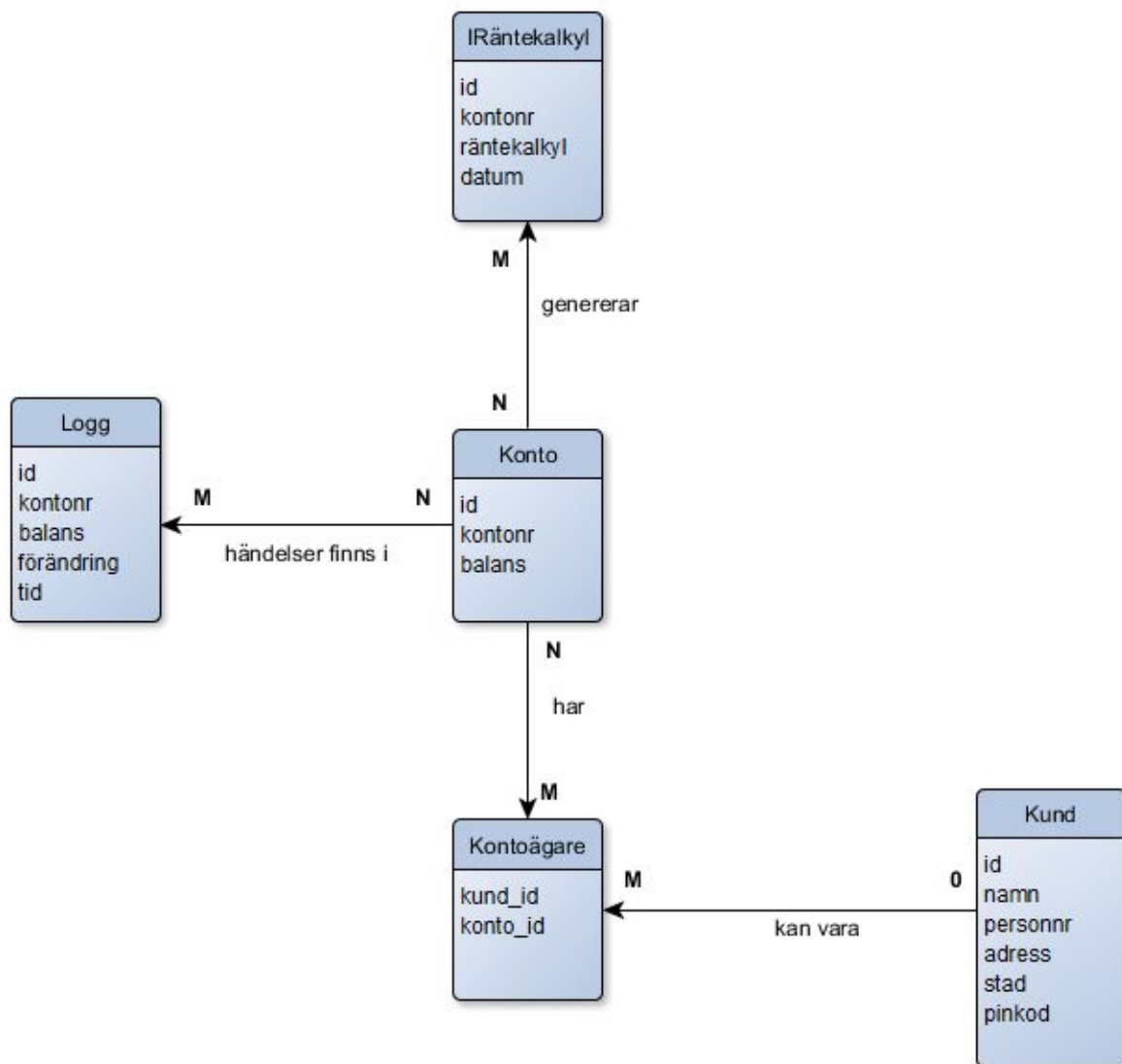


## 2. Logisk modell: BankBluffOBåg AB

## 2.1 ER-diagram enligt relationsmodellen

Ett ER-diagram enligt relationsmodellen har skapats i ritprogrammet Yed. Det uppdaterade diagrammet kan beskrivas enligt följande:

- Kund (id, namn, personnr, adress, stad, pinkod)
- Konto (id, kontonr, balans)
- Kontoägare (kund-id, konto-id)
- Logg (id, kontonr, balans, förändring, tid)
- Räntekalkyl (id, kontonr, räntekalkyl, datum)



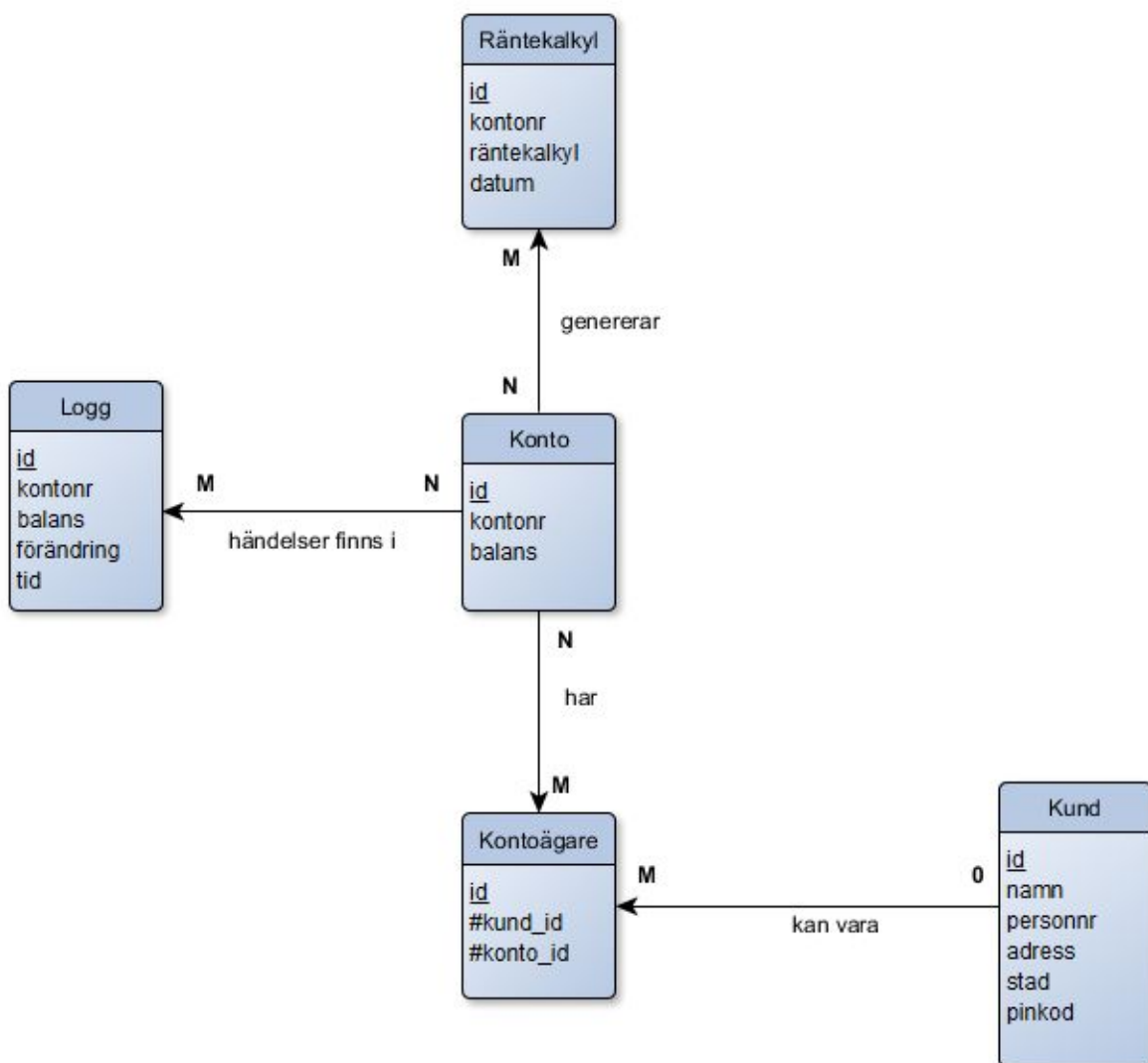


## 2.2 ER-diagram med primära och främmande nycklar

ER-diagrammet har nu kompletterats med nycklar. Det uppdaterade diagrammet kan beskrivas enligt följande (primärnycklar är understrukna och främmande nycklar inleds med #):

- Kund (id, namn, personnr, adress, stad, pinkod)
- Konto (id, kontonr, balans)
- Kontoägare (id, #kund-id, #konto-id)
- Logg (id, kontonr, balans, förändring, tid)
- Räntekalkyl (id, kontonr, räntekalkyl, datum)

Nedan visualiseras ER-diagrammet, understrukna rader är primärnycklar och rader som inleds med # är främmande nycklar.



## 3. Fysisk modell: BankBluffOBåg AB

### 3.1 SQL DDL för tabellerna

SQL DDL för tabellerna i databasen eshop skapades för hand utifrån ER-diagrammet i punkt 2.2, se bilaga 1.

### 3.2 Funktioner som databasen ska stödja (API)

För nyttjande av databasen bör ett antal transaktioner stödjas. Nedan följer en lista med de funktioner som ska stödjas av databasen internetbanken.

1. Skapa ny kund
2. Skapa nytt konto och koppla det till existerande kund
3. Addera existerande kund som ägare till existerande konto
4. Swisha pengar mellan konton
5. Skapa räntekalkyl
6. Lista alla kunder och deras konton med tillhörande balans
7. Lista alla konton som tillhör en viss kund
8. Visa vilka som har tillgång till ett visst konto
9. Flytta pengar mellan egna konton
10. Lista alla konton med tillhörande räntekalkyl
11. Visa räntekalkyl för konton som tillhör viss kund
12. Visa den totala summan av alla konton
13. Lista alla rader i logg-tabellen

### 3.3 Applikationer

#### 3.3.1 Räntekalkyl

För att utföra räntekalkyl, visa summerad ränta för ett visst datum samt visa summerad ränta för att visst år används lagrade procedurer som anropas enligt följande:

- CALL makeCalculationOfInterest('1.1', '2018-04-19');

Där 1.1 är räntan och 2018-04-19 är datumet då kalkylen utförs.

- CALL showCalculationOfInterestDate('2018-04-19');
- CALL showCalculationOfInterestYear('2018');

Där 2018-04-19 är datumet då vilka räntorna ska summeras och 2018 är året då räntorna ska summeras.

### 3.3.2 CLI program

Cli-programmet startas med node cli.js. Om användaren matar in "menu" listas vad användaren kan göra och hur hen går till väga.

```
$ node cli.js
Välkommen till internetbanken! Här kan du se dina konton och använda swish.
Internetbanken: menu
Du kan välja att göra följande:
$ help, menu - visa denna menyn.
swish <användar-id> <pin> <från kontonummer>
<summa> <till kontonummer> - swisha pengar.
konton <användar-id> <pin> - lista dina bankkonton.
exit, quit, ctrl-d - avsluta programmet.
Internetbanken:
```

Programmet består av en swishapplikation

```
Internetbanken: swish 1 1243 900-5566 100 901-5567
Du har nu swishat 100 kronor till konto 901-5567!
```

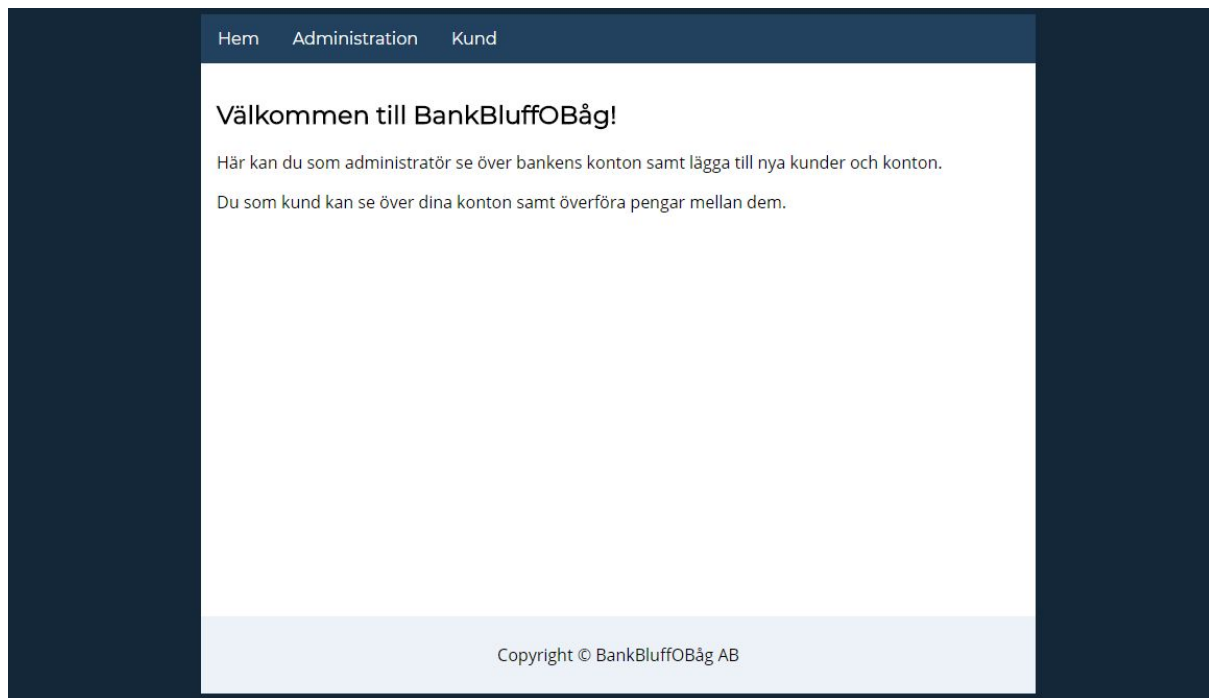
och en funktion för att lista sina konton.

```
Internetbanken: konton 1 1243
Mina konton:
```

Id	Kontonummer	Balans	Delas med
1	900-5566	19990	Louise Jakobsson, Frank Franksson, Kråkan Kråk, Siv Ert
2	901-5567	110100	Louise Jakobsson, Frank Franksson, Kråkan Kråk
7	906-6679	40000	Louise Jakobsson

### 3.3.3 WEB interface

Webbapplikationen består av funktioner för administratörer och kunder.



Administratörer kan bland annat, på sidan “konton”, lista konton, lägga till kontoägare och lägga till nya konton.

Hem Administration Kund

Start Kunder Konton Räntekalkyl Logg

## Konton

Nedan visas bankens konton och räntekalkyl.

Id	Kontonummer	Balans	Ägare	
1	900-5566	19990	Louise Jakobsson, Frank Franksson, Kråkan Kråk, Siv Ert	Lägg till ägare
2	901-5567	110100	Louise Jakobsson, Frank Franksson, Kråkan Kråk	Lägg till ägare
3	902-5568	10000	Milo Karlsson	Lägg till ägare
4	903-5569	50000	Stig Stigsson	Lägg till ägare
5	904-6677	60000	Sten Stensson, Jacob Unger	Lägg till ägare
6	905-6678	1000	Ada Ada	Lägg till ägare
7	906-6679	40000	Louise Jakobsson	Lägg till ägare
8	907-7788	98910	Frank Franksson	Lägg till ägare
9	908-7789	56000	Kråkan Kråk	Lägg till ägare
10	909-7790	23000	Milo Karlsson	Lägg till ägare
11	910-8866	2000	Stig Stigsson	Lägg till ägare

Kunder verifierar sig genom att klicka på sitt eget id.

[Hem](#) [Administration](#) [Kund](#)

### Min sida

Klicka på ditt id nedan för att komma åt dina bankkonton.

	<b>Id</b>	<b>Namn</b>	<b>Född</b>	<b>Adress</b>	<b>Kontonummer</b>
1	Louise Jakobsson	Wed Jun 24 1992	Hinderbanan 26, Göteborg	900-5566,901-5567,906-6679	
2	Frank Franksson	Fri Sep 22 1995	Frankvägen 5, Lerum	900-5566,901-5567,907-7788	
3	Kråkan Kråk	Mon Jan 16 1956	Kråkvägen 20, Alingsås	900-5566,901-5567,908-7789	
4	Milo Karlsson	Thu Feb 12 1998	Milogränden 28, Stockholm	902-5568,909-7790,914-9967	
5	Stig Stigsson	Wed Jan 27 1988	Stigen 6, Kiruna	903-5569,910-8866,915-9977	
6	Sten Stensson	Thu May 06 1982	Stenen 7, Floda	904-6677,911-8877	
7	Ada Ada	Mon Jun 03 1940	Adaslingen 55, Gränna	905-6678,912-8888	
8	Siv Ert	Wed Aug 24 1955	Siverts gata 10, Partille	900-5566,913-8899	
9	Jacob Unger	Thu Jan 16 1992	Inärogatan 16, Göteborg	904-6677	

De kan bland annat lista sina konton och föra över pengar.

[Hem](#) [Administration](#) [Kund](#)

[Min sida](#) [Mina konton](#) [Räntekalkyl](#)

### Mina konton

<b>Id</b>	<b>Kontonummer</b>	<b>Balans</b>	<b>Delas med</b>
1	900-5566	19990	Louise Jakobsson, Frank Franksson, Kråkan Kråk, Siv Ert
2	901-5567	110100	Louise Jakobsson, Frank Franksson, Kråkan Kråk
7	906-6679	40000	Louise Jakobsson

Överföring mellan konton

Från konto (nnn-nnnn):

Till konto (nnn-nnnn):

Summa:

# Bilaga 1: SQL DDL

## **accountHolder:**

```
CREATE TABLE accountHolder
(
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(50),
  born DATE,
  address VARCHAR(50),
  city VARCHAR(30),
  pin INT,

  PRIMARY KEY (id)
);
```

## **account:**

```
CREATE TABLE account
(
  id INT NOT NULL AUTO_INCREMENT,
  accountNumber CHAR(8),
  balance DECIMAL,

  PRIMARY KEY (id)
);
```

## **accountOwner:**

```
CREATE TABLE accountOwner
(
  id INT NOT NULL AUTO_INCREMENT,
  accountHolder_id INT,
  FOREIGN KEY (accountHolder_id) REFERENCES accountHolder(id),
  account_id INT,
  FOREIGN KEY (account_id) REFERENCES account(id),

  PRIMARY KEY (id)
);
```

**log:**

```
CREATE TABLE log
(
    id INT NOT NULL AUTO_INCREMENT,
    accountNumber CHAR(8),
    balance DECIMAL,
    amountChange DECIMAL,
    logTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    PRIMARY KEY (id)
);
```

**calculationInterest:**

```
CREATE TABLE calculationOfInterest
(
    id INT NOT NULL AUTO_INCREMENT,
    accountNumber CHAR(8),
    calcInterest DECIMAL,
    calcDate DATE,

    PRIMARY KEY (id)
);
```