

---

# **Cryptokeeper Dokumentation**

***Release 0.1.2***

**s0556166 Steffen Exler und Alain Ngoufack Nguetack s0557214**

**21.03.2018**



---

## Contents:

---

<b>1</b>	<b>Beschreibung</b>	<b>1</b>
<b>2</b>	<b>ER-Diagramm</b>	<b>3</b>
<b>3</b>	<b>Implementationsentwurf</b>	<b>5</b>
3.1	Initial-Entwurf . . . . .	5
3.2	Verfeinerung durch Zusammenfassung . . . . .	6
3.3	Tabellen Form . . . . .	6
<b>4</b>	<b>SQL Script</b>	<b>9</b>
4.1	Schema . . . . .	9
4.2	Data . . . . .	11
<b>5</b>	<b>Operatorbaum</b>	<b>15</b>
5.1	Aufgabe: . . . . .	15
<b>6</b>	<b>SQL Befehle</b>	<b>17</b>
6.1	Befehl B (Join & Kathesisches Produkt) . . . . .	17
6.2	Befehl C (in or not exists) . . . . .	17
6.3	Befehl D (Group) . . . . .	17
6.4	Befehl E (Having) . . . . .	18



# KAPITEL 1

---

## Beschreibung

---

Cryptokeeper ist eine Treuhand investment Buchhaltungssoftware für Krypto Coins wie Bitcoin. Ziel der Software ist es eine einfache Gewinn und Verlust Rechnung zu erzeugen, so das es leicht zu erkennen ist, wer wie viel Gewinn und Verlust mit dem jeweiligen Coin getätigt hat, sowie eine Gesamt Gewinn und Verlust Rechnung für die jeweiligen Investoren.

Jede Transaktion wird gespeichert wo enthalten ist wann, wer, wie viel, welchen Coin und zu welchen Kurs gehandelt wurde, auch auf welcher Börse gehandelt wurde und zu welcher Wallet die Coins überwiesen wurden.

Von den Investoren wird der Firmenname, Anschrift, Geschäftsform und Kontakt Person gesichert.

Die Krypto Coins enthalten den Coin Namen, Kürzel, wechselkurs in Euro, Zeitstempel der letzten Aktualisierung, Algorithmus und Prüf Verfahren. In Algorithmus und Prüf Verfahren wird nur der jeweilige Name eingetragen.

In Wallet wird eingetragen um was für ein Krypto Coin Wallet es sich handelt und welche Adresse sie besitzt.

Die Börse enthält einen Namen und URL.



## ER-Diagramm

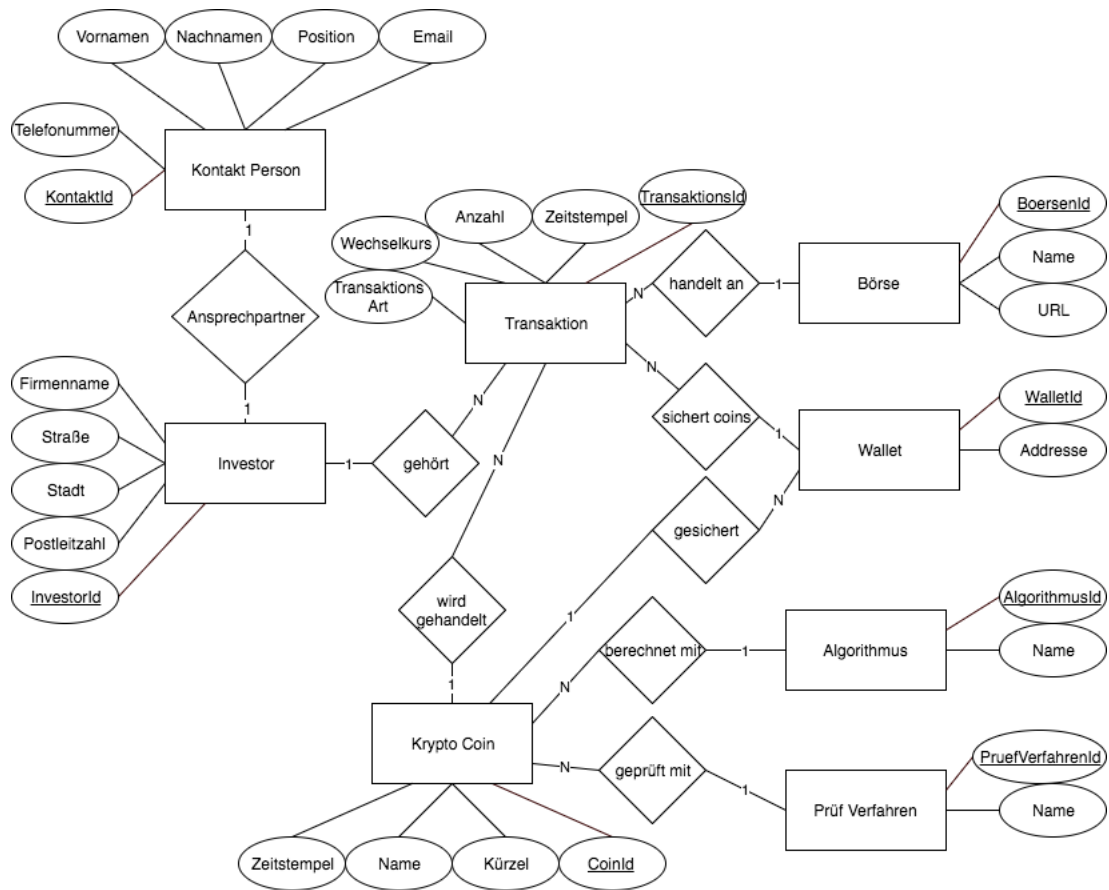


Abb. 2.1: ER-Diagramm





### 3.1 Initial-Entwurf

Kontakt Person: [KontaktId : integer, Vornamen : string, Nachnamen : string, Position : string, Email : string, Telefonnummer : string]

Investor: [InvestorId : integer, Firmenname : string, Straße : string, Stadt : string, Postleitzahl : string]

Transaktion: [TransaktionsId : integer, Anzahl : float, Zeitstempel : Zeitstempel, Wechselkurs : float, Transaktions Art : string]

Börse: [BoersenId : integer, Name : string, URL : string]

Wallet: [WalletId : integer, Adresse : string]

Krypto Coin: [CoinId : integer, Name : string, Kürzel : string, Zeitstempel : integer]

Algorithmus: [AlgorithmusId : integer, Name : string]

Prüf Verfahren: [PruefVerfahrenId : integer, Name : string]

Ansprechpartner: [InvestorId : integer, KontaktId : integer]

gehört: [InvestorId : integer, TransaktionsId : integer]

handelt an: [BoersenId : integer, TransaktionsId : integer]

sichert coins: [WalletId : integer, TransaktionsId : integer]

wird gehandelt: [CoinId : integer, TransaktionsId : integer]

berechnet mit: [CoinId : integer, AlgorithmusId : integer]

geprüft mit: [CoinId : integer, PruefVerfahrenId : integer]

gesichert: [CoinId : integer, WalletId : integer]

## 3.2 Verfeinerung durch Zusammenfassung

Kontakt Person: [KontaktId : integer, Vornamen : string, Nachnamen : string, Position : string, Email : string, Telefonnummer : string, ansprechpartnerVon : integer]

Investor: [InvestorId : integer, Firmenname : string, Straße : string, Stadt : string, Postleitzahl : string]

Transaktion: [TransaktionId : integer, Anazhl : float, Zeitstempel : Zeitstempel, Wechselkurs : float, Transaktions Art : string, CoinId : integer, InvestorId : integer, WalletId : integer, InvestorId : integer]

Börse: [BoersenId : integer, Name : string, URL : string]

Wallet: [WalletId : integer, Adresse : string, CoinId : integer]

Krypto Coin: [CoinId : integer, Name : string, Kürzel : string, Zeitstempel : integer, AlgorithmusId : integer, PruefVerfahrenId : integer]

Algorithmus: [AlgorithmusId : integer, Name : string]

Prüf Verfahren: [PruefVerfahrenId : integer, Name : string]

## 3.3 Tabellen Form

investor				
investor_id	firmenname	straße	stadt	postleitzahl
1	CoinHolder GmbH	Mustersteg 27	Groß Kiez	13147
2	Fruits & Co Kg	Bananenlauf 15	Ackerland	13133
3	Trading AG	Kuhdamm	Berlin	10966
4	Getränke Handel Hans	Buschkrug Alle 12	Groß Kiez	13147

kontakt_person						
kontakt_id	vorna-men	nachna-men	position	email	telefonnum-mer	ansprechpart-ner_von
0	Lisa	Schmidt	Assistent	li-sa.schmidt@coin.de	123456789	1
1	Linda	Krans	Geschäfts-führer	krans@linda.de	234567890	2
2	Harald	Hard	Inhaber	hart@harald.com	345678901	3
3	Frido	Froh	Assistent	froh@hans-getranke.de	456789012	4

algorithmus	
algorithmus_id	name
1	SHA256
2	Scrypt
3	CryptoNight
4	X11

pruef_verfahren	
pruef_verfahren_id	name
1	PoW
2	PoS
3	PoW/PoS

krypto_coin				
coin_id	name	kuerzel	berechnet_mit	geprueft_mit
1	Bitcoin	BTC	1	1
2	Litecoin	LTC	2	1
3	Monero	XMR	3	1
4	CRAIG	CRAIG	4	2

boerse		
boerse_id	name	url
1	MtGox	<a href="https://mtgox.com">https://mtgox.com</a>
2	Bitstamp	<a href="https://bitstamp.net">https://bitstamp.net</a>
3	Kraken	<a href="https://kraken.com">https://kraken.com</a>
4	Localbitcoins	<a href="https://localbitcoins.com">https://localbitcoins.com</a>

wallet		
wallet_id	adresse	gesicherter
1	itdw27cN3mprpZ3PTAU54FKuEg7wAnh6URw8a14PCBk	1
2	N5u2a5u3szELc9SiEseVmjKiHMZKm5imcRQlQQZfLhe	2
3	uibRalcAD1Af5nPezQBfQKF24F48h68Vqlxn6WpmT1	3
4	tf89JYlxByyCklLdv96irL4QF7ibdyYgK62OFeyqly2	4
5	ddddJYlxByydsfggfdghCklLdv96irL4QF7isdIy2	1

transaktion							
transacti-on_id	an-zahl	zeitstempel	wechsel-kurs	transakti-ons_art	gehan-delt	gesi-chert_in	ge-hoert
1	12.5	2018-01-08 20:09:38.830624	1.99	buy	1	1	1
2	5	2018-01-08 20:09:38.833055	0.001	buy	2	2	2
3	19.99	2018-01-08 20:09:38.835596	10242.001	sell	3	3	3
4	0.0012	2018-01-08 20:09:38.837327	23.53	buy	4	4	3



### 4.1 Schema

```
--drop tables if exists

DROP TABLE kontakt_person;
DROP TABLE transaktion;
DROP TABLE boerse;
DROP TABLE wallet;
DROP TABLE krypto_coin;
DROP TABLE algorithmus;
DROP TABLE pruef_verfahren;
DROP TABLE investor;

--create new tables

CREATE TABLE investor
(
  investor_id  INTEGER PRIMARY KEY,
  firmenname   VARCHAR(30),
  straÙe       VARCHAR(30),
  stadt        VARCHAR(30),
  postleitzahl VARCHAR(30)
);

CREATE TABLE kontakt_person
(
  kontakt_id   INTEGER PRIMARY KEY,
  vornamen     VARCHAR(30) NOT NULL,
  nachnamen    VARCHAR(30) NOT NULL,
  position      CHAR(15) CHECK (position IN ('Inhaber', 'Geschäftsführer',
→ 'Assistent')),
  email        VARCHAR(50) UNIQUE,
  telefonnummer VARCHAR(20) UNIQUE NOT NULL,
```

```
    ansprechpartner_von INTEGER REFERENCES investor ON DELETE CASCADE
);

CREATE TABLE algorithmus
(
    algorithmus_id INTEGER PRIMARY KEY,
    name           VARCHAR(30) NOT NULL
);

CREATE TABLE pruef_verfahren
(
    pruef_verfahren_id INTEGER PRIMARY KEY,
    name               VARCHAR(30) NOT NULL
);

CREATE TABLE krypto_coin
(
    coin_id          INTEGER PRIMARY KEY,
    name             VARCHAR(30) NOT NULL,
    kuerzel          VARCHAR(8)  NOT NULL,
    berechnet_mit    INTEGER REFERENCES algorithmus,
    geprueft_mit     INTEGER REFERENCES pruef_verfahren
);

CREATE TABLE boerse
(
    boerse_id INTEGER PRIMARY KEY,
    name      VARCHAR(30) NOT NULL,
    url       VARCHAR(30) NOT NULL
);

CREATE TABLE wallet
(
    wallet_id   INTEGER PRIMARY KEY,
    adresse     VARCHAR(100) NOT NULL,
    gesicherter INTEGER REFERENCES krypto_coin
);

CREATE TABLE transaktion
(
    transaction_id  SERIAL PRIMARY KEY,
    anzahl         FLOAT NOT NULL,
    zeitstempel     TIMESTAMP DEFAULT current_timestamp, --auto fill with current unix_
    ↳time
    wechselkurs     FLOAT NOT NULL,
    transaktions_art CHAR(4) CHECK (transaktions_art IN ('buy', 'sell')),
    gehandelt       INTEGER REFERENCES krypto_coin,
    gesichert_in    INTEGER REFERENCES wallet,
    gehoert         INTEGER REFERENCES investor
);
```

## 4.2 Data

```
--investor

INSERT INTO investor (investor_id, firmenname, straÙe, stadt, postleitzahl)
VALUES (1, 'CoinHolder GmbH', 'Mustersteg 27', 'GroÙ Kiez', '13147');

INSERT INTO investor (investor_id, firmenname, straÙe, stadt, postleitzahl)
VALUES (2, 'Fruits & Co Kg', 'Bananenlauf 15', 'Ackerland', '13133');

INSERT INTO investor (investor_id, firmenname, straÙe, stadt, postleitzahl)
VALUES (3, 'Trading AG', 'Kuhdamm', 'Berlin', '10966');

INSERT INTO investor (investor_id, firmenname, straÙe, stadt, postleitzahl)
VALUES (4, 'Getrânke Handel Hans', 'Buschkrug Alle 12', 'GroÙ Kiez', '13147');

--kontakt_person

INSERT INTO kontakt_person (kontakt_id, vornamen, nachnamen, position, email, ↵
↵telefonnummer, ansprechpartner_von)
VALUES (0, 'Lisa', 'Schmidt', 'Assistent', 'lisa.schmidt@coin.de', '123456789', 1);

INSERT INTO kontakt_person (kontakt_id, vornamen, nachnamen, position, email, ↵
↵telefonnummer, ansprechpartner_von)
VALUES (1, 'Linda', 'Krans', 'Geschäftsführer', 'krans@linda.de', '234567890', 2);

INSERT INTO kontakt_person (kontakt_id, vornamen, nachnamen, position, email, ↵
↵telefonnummer, ansprechpartner_von)
VALUES (2, 'Harald', 'Hard', 'Inhaber', 'hart@harald.com', '345678901', 3);

INSERT INTO kontakt_person (kontakt_id, vornamen, nachnamen, position, email, ↵
↵telefonnummer, ansprechpartner_von)
VALUES (3, 'Frido', 'Froh', 'Assistent', 'froh@hans-getranke.de', '456789012', 4);

--boerse

INSERT INTO boerse (boerse_id, name, url)
VALUES (1, 'MtGox', 'https://mtgox.com');

INSERT INTO boerse (boerse_id, name, url)
VALUES (2, 'Bitstamp', 'https://bitstamp.net');

INSERT INTO boerse (boerse_id, name, url)
VALUES (3, 'Kraken', 'https://kraken.com');

INSERT INTO boerse (boerse_id, name, url)
VALUES (4, 'Localbitcoins', 'https://localbitcoins.com');

--algorithmus

INSERT INTO algorithmus (algorithmus_id, name)
VALUES (1, 'SHA256');

INSERT INTO algorithmus (algorithmus_id, name)
VALUES (2, 'Scrypt');

INSERT INTO algorithmus (algorithmus_id, name)
VALUES (3, 'CryptoNight');
```

```
INSERT INTO algorithmus (algorithmus_id, name)
VALUES (4, 'X11');

--pruef_verfahren

INSERT INTO pruef_verfahren (pruef_verfahren_id, name)
VALUES (1, 'PoW');

INSERT INTO pruef_verfahren (pruef_verfahren_id, name)
VALUES (2, 'PoS');

INSERT INTO pruef_verfahren (pruef_verfahren_id, name)
VALUES (3, 'PoW/PoS');

--krypto_coin

INSERT INTO krypto_coin (coin_id, name, kuerzel, berechnet_mit, geprueft_mit)
VALUES (1, 'Bitcoin', 'BTC', 1, 1);

INSERT INTO krypto_coin (coin_id, name, kuerzel, berechnet_mit, geprueft_mit)
VALUES (2, 'Litecoin', 'LTC', 2, 1);

INSERT INTO krypto_coin (coin_id, name, kuerzel, berechnet_mit, geprueft_mit)
VALUES (3, 'Monero', 'XMR', 3, 1);

INSERT INTO krypto_coin (coin_id, name, kuerzel, berechnet_mit, geprueft_mit)
VALUES (4, 'CRAIG', 'CRAIG', 4, 2);

--wallet

INSERT INTO wallet (wallet_id, adresse, gesicherter)
VALUES (1, 'itdw27cN3mprpZ3PTAU54FKuEg7wAnh6URw8a14PCBk', 1);

INSERT INTO wallet (wallet_id, adresse, gesicherter)
VALUES (2, 'N5u2a5u3szELc9SiEseVmjKiHmZKm5imcRQlQQZfLhe', 2);

INSERT INTO wallet (wallet_id, adresse, gesicherter)
VALUES (3, 'uibRalcAD1Af5nPezQBfQKF24F48h68Vqlxn6WpmT1', 3);

INSERT INTO wallet (wallet_id, adresse, gesicherter)
VALUES (4, 'tf89JY1xByyCk1Ldv96irL4QF7ibdyYgK620FeYqIy2', 4);

INSERT INTO wallet (wallet_id, adresse, gesicherter)
VALUES (5, 'ddddJY1xByydsfggfdghCk1Ldv96irL4QF7isdIy2', 1);

--transaktion

INSERT INTO transaktion (transaction_id, anzahl, wechselkurs, transaktions_art, ↵
↵gehandelt, gesichert_in, gehoert)
VALUES (1, 12.5, 1.99, 'buy', 1, 1, 1);

INSERT INTO transaktion (transaction_id, anzahl, wechselkurs, transaktions_art, ↵
↵gehandelt, gesichert_in, gehoert)
VALUES (2, 5, 0.001, 'buy', 2, 2, 2);

INSERT INTO transaktion (transaction_id, anzahl, wechselkurs, transaktions_art, ↵
↵gehandelt, gesichert_in, gehoert)
```



```
VALUES (3, 19.99, 10242.001, 'sell', 3, 3, 3);

INSERT INTO transaktion (transaction_id, anzahl, wechselkurs, transaktions_art, ↵
↳gehandelt, gesichert_in, gehoert)
VALUES (4, 0.0012, 23.53, 'buy', 4, 4, 3);
```



## 5.1 Aufgabe:

Welcher Investor hat noch keine Transaktion getätigt.

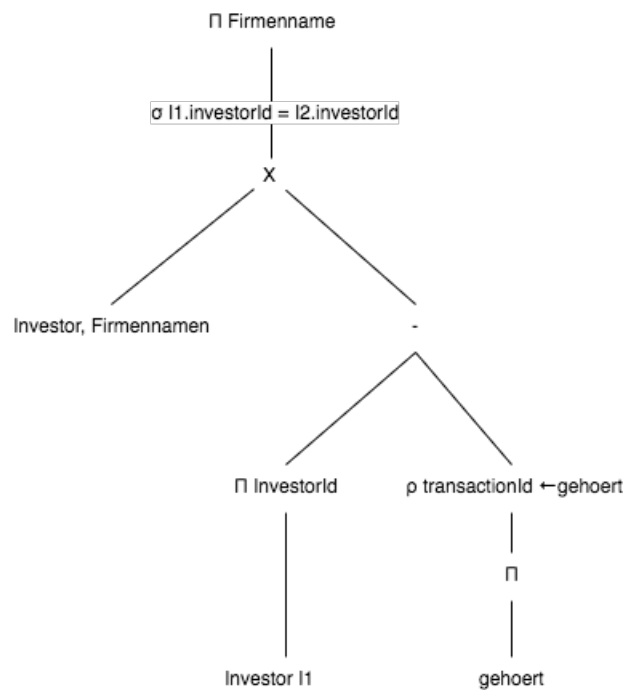


Abb. 5.1: Operatorbaum

```
SELECT firmenname FROM investor WHERE investor_id=(SELECT investor_id from investor_
EXCEPT (SELECT gehoert from transaktion));
```



### 6.1 Befehl B (Join & Kathesisches Produkt)

Welche Investoren haben welche Transaktionen getätigt?

```
SELECT transaction_id, firmenname FROM transaktion JOIN investor i ON transaktion.  
↳ gehoert = i.investor_id;
```

Welche Investoren haben mindestens eine Transaktion getätigt?

```
SELECT investor_id from investor INTERSECT (SELECT gehoert from transaktion);
```

### 6.2 Befehl C (in or not exists)

Welche Wallet wurde nicht verwendet?

```
SELECT * from wallet WHERE NOT EXISTS (SELECT * FROM transaktion WHERE gesichert_  
↳ in=wallet.wallet_id);
```

### 6.3 Befehl D (Group)

Wie oft kommen die jeweiligen Postleitzahlen in Investor vor?

```
SELECT count(investor_id), postleitzahl FROM investor GROUP BY postleitzahl;
```

Wie oft wird welche Wallet Adresse in Transaktion verwendet, die mindestens einmal vorkommt?

```
SELECT count(wallet_id), adresse FROM wallet WHERE EXISTS (SELECT * FROM transaktion_  
↳ WHERE gesichert_in=wallet.wallet_id) GROUP BY adresse;
```

## 6.4 Befehl E (Having)

Welche Krypto Coins wurde im durchschnitt Wechselkurs mit weniger als 2 Euro gehandelt?

```
SELECT gehandelt FROM transaktion GROUP BY gehandelt HAVING avg(wechselkurs) < 2;
```