

Datenbanksysteme II

Praktikum SS 16

Alexandra Vogel, Alexander Boxhorn

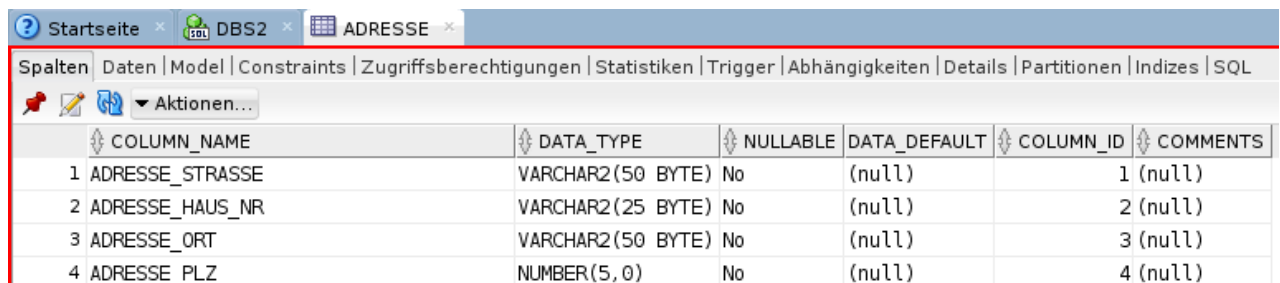
Praktikum 2:

Verwendeten Datenbank: DBST49

Teilaufgabe 1: Bankbeispiel

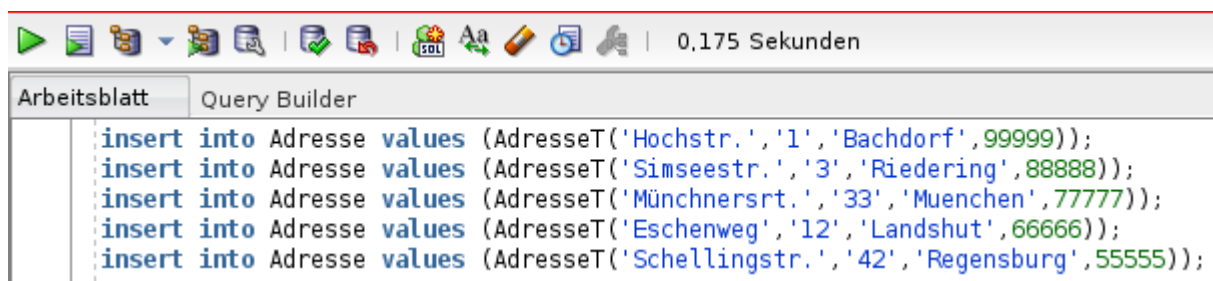
Die Tabellen für das Bankbeispiel sind in der Datenbank in zwei unterschiedlichen objektrationalen Varianten gespeichert. In der ersten Darstellung werden je Zweigstelle und je Kunde die zugehörigen Konten als Referenzen gespeichert. Die zweite Darstellung speichert die zu einer Zweigstelle gehörigen Konten nicht als Referenzen. Die Kunden werden hingegen in den einzelnen Konten als Referenzen gespeichert.

Bei beiden Darstellungsarten werden die Adressen als Referenzen bei den Kunden bzw. bei den Zweigstellen gespeichert. Für das speichern der Adressen in der Tabelle *Adresse* sind unten abgebildete SQL-Befehle notwendig.



The screenshot shows a database management tool interface with the 'ADRESSE' table selected. The table structure is displayed in a grid with columns: COLUMN_NAME, DATA_TYPE, NULLABLE, DATA_DEFAULT, COLUMN_ID, and COMMENTS. The table has four columns: ADRESSE_STRASSE, ADRESSE_HAUS_NR, ADRESSE_ORT, and ADRESSE_PLZ.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ADRESSE_STRASSE	VARCHAR2(50 BYTE)	No	(null)	1 (null)	
2	ADRESSE_HAUS_NR	VARCHAR2(25 BYTE)	No	(null)	2 (null)	
3	ADRESSE_ORT	VARCHAR2(50 BYTE)	No	(null)	3 (null)	
4	ADRESSE_PLZ	NUMBER(5,0)	No	(null)	4 (null)	

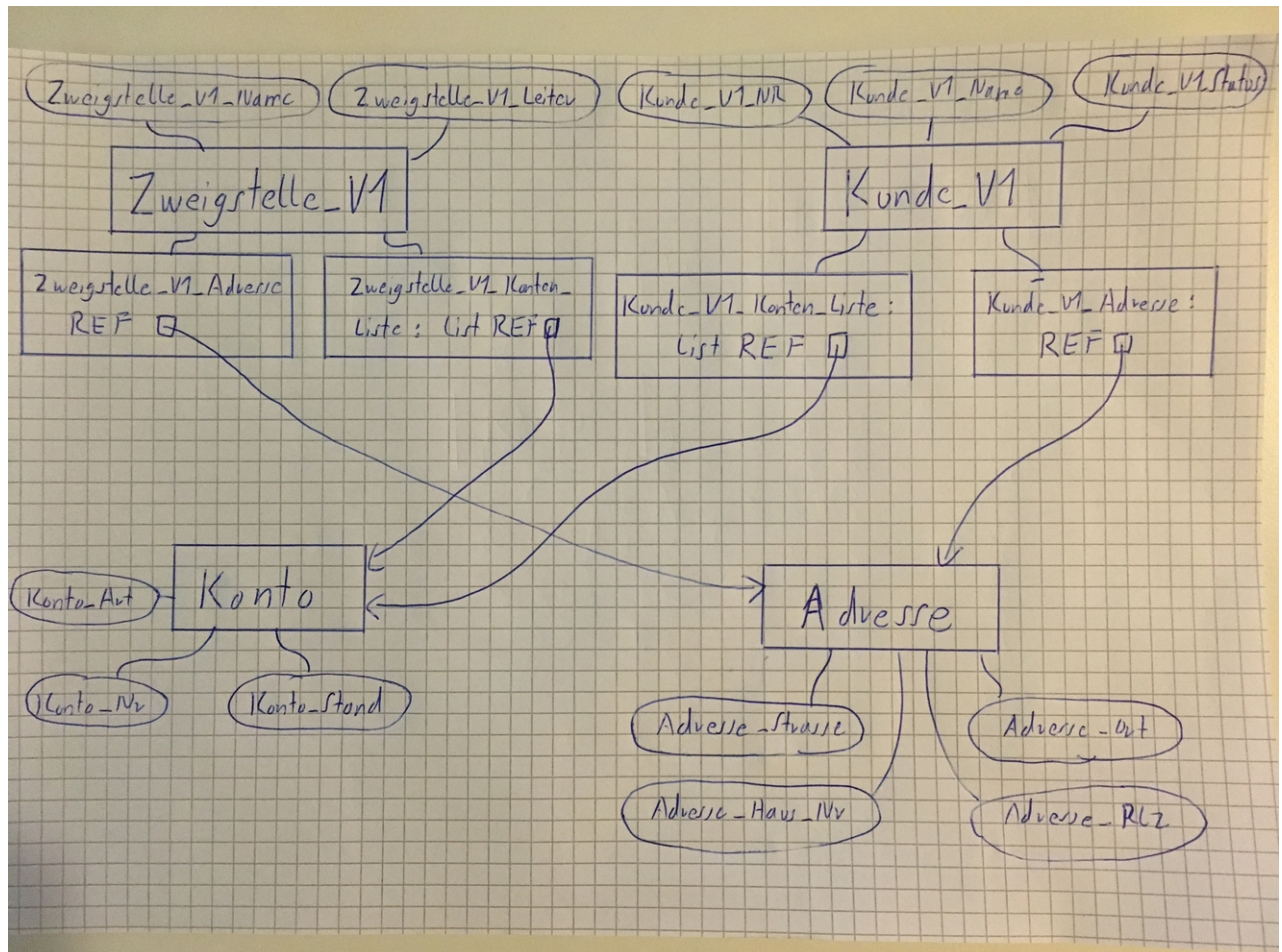


The screenshot shows a SQL query editor with five INSERT statements for the 'Adresse' table. The statements are as follows:

```
insert into Adresse values (AdresseT('Hochstr.', '1', 'Bachdorf', 99999));
insert into Adresse values (AdresseT('Simseestr.', '3', 'Riedering', 88888));
insert into Adresse values (AdresseT('Münchnersrt.', '33', 'Muenchen', 77777));
insert into Adresse values (AdresseT('Eschenweg', '12', 'Landshut', 66666));
insert into Adresse values (AdresseT('Schellingstr.', '42', 'Regensburg', 55555));
```

Darstellung 1:

Wie in der Übersichtszeichnung ersichtlich referenziert jeder Kunde und jede Zweigstelle auf eine Liste von Konten welche zum Kunden bzw. zur Zweigstelle gehören. Zusätzlich zeigt jeder Kunde und jede Zweigstelle zur dazugehörigen Adresse.



Das Datenbankschema für die Darstellung 1 umfasst die in den folgenden Abbildungen dargestellten Tabellen.

ZWEIGSTELLE_V1					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ZWEIGSTELLE_V1_NAME	VARCHAR2(50 BYTE)	No	(null)	1 (null)	
2 ZWEIGSTELLE_V1_LEITER	VARCHAR2(50 BYTE)	Yes	(null)	2 (null)	
3 ZWEIGSTELLE_V1_ADRESSE	ADRESSET	Yes	(null)	3 (null)	
4 ZWEIGSTELLE_V1_KONTEN_LISTE	KONTENLISTET	Yes	(null)	4 (null)	

Startseite x DBS2 x KUNDE_V1 x						
Spalten Daten Model Constraints Zugriffsberechtigungen Statistiken Trigger Abhängigkeiten Details Partitionen Indizes SQL						
Aktionen...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	KUNDE_V1_NR	NUMBER(4,0)	No	(null)	1	(null)
2	KUNDE_V1_NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	KUNDE_V1_STATUS	VARCHAR2(25 BYTE)	Yes	(null)	3	(null)
4	KUNDE_V1_ADRESSE	ADRESSET	Yes	(null)	4	(null)
5	KUNDE_V1_KONTEN_LISTE	KONTENLISTET	Yes	(null)	5	(null)

Startseite x DBS2 x KONTO x						
Spalten Daten Model Constraints Zugriffsberechtigungen Statistiken Trigger Abhängigkeiten Details Partitionen Indizes SQL						
Aktionen...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	KONTO_ART	VARCHAR2(50 BYTE)	Yes	(null)	1	(null)
2	KONTO_NR	NUMBER(9,0)	No	(null)	2	(null)
3	KONTO_STAND	NUMBER	Yes	(null)	3	(null)

Das Anlegen der Konten in Tabelle Konto wird mit folgendem SQL-Befehl durchgeführt.

```

insert into Konto values (KontoT('S', 120768, 234.56));
insert into Konto values (KontoT('G', 678453, -456.78));
insert into Konto values (KontoT('G', 348973, 12567.56));
insert into Konto values (KontoT('G', 987654, 789.65));
insert into Konto values (KontoT('S', 745363, -23.67));

```

Für die Tabelle Zweigstelle_V1 sind die folgenden SQL-Statemantes nötig.

```

insert into ZWEIGSTELLE_V1 values
(ZweigstelleT_v1( 'Bachdorf',
                  '1768',
                  (Select REF (w) from Adresse w
                   where w.Adresse_plz = 99999),
                  KontenlisteT()
                 ));

insert into ZWEIGSTELLE_V1 values
(ZweigstelleT_v1( 'Riedering',
                  '9823',
                  (Select REF (w) from Adresse w
                   where w.Adresse_plz = 88888),
                  KontenlisteT()
                 ));

```

Für die Kunden wird in der Tabelle Kunde_V1 der folgende Befehl verwendet.

```

Arbeitsblatt | Query Builder
--
insert into Kunde_v1 values
(2345,
'H. Fach',
'Geschäftskunde',
(Select Ref (w) from Adresse w
where w.adresse_Plz = 77777),
KontenListeT());

insert into Kunde_v1 values
(7654,
'B. Meier',
'Privatkunde',
(Select Ref (w) from Adresse w
where w.adresse_Plz = 66666),
KontenListeT());

insert into Kunde_v1 values
(8764,
'J. Wiesner',
'Geschäftskunde',
(Select Ref (w) from Adresse w
where w.adresse_Plz = 55555),
KontenListeT());

```

Das abspeichern der Referenzen zu den Konten in den Tabellen Kunde_V1 und Zweigstelle_V1 wird in folgenden zwei Abbildung dargestellt.

```

Arbeitsblatt | Query Builder | 0,178 Sekunden
--
insert into Table (Select Kunde_V1_Konten_Liste
From Kunde_v1
Where Kunde_v1_Nr = 2345)
( Select REF (w) from Konto w
Where w.Konto_Nr = 120768
Or w.Konto_Nr = 348973
);

insert into Table (Select Kunde_V1_Konten_Liste
From Kunde_v1
Where Kunde_v1_Nr = 7654)
( Select REF (w) from Konto w
Where w.Konto_Nr = 987654
);

insert into Table (Select Kunde_V1_Konten_Liste
From Kunde_v1
Where Kunde_v1_Nr = 8764)
( Select REF (w) from Konto w
Where w.Konto_Nr = 745363
Or w.Konto_Nr = 678453
Or w.Konto_Nr = 348973
);

```

```

Arbeitsblatt Query Builder
insert into Table (Select ZWEIGSTELLE_V1_Konten_Liste
                    From Zweigstelle_v1
                    Where ZWEIGSTELLE_V1_Name = 'Bachdorf')
( Select REF (w) from Konto w
  Where w.Konto_Nr = 120768
    Or w.Konto_Nr = 348973
    Or w.Konto_Nr = 678453
);

insert into Table (Select ZWEIGSTELLE_V1_Konten_Liste
                    From Zweigstelle_v1
                    Where ZWEIGSTELLE_V1_Name = 'Riedering')
( Select REF (w) from Konto w
  Where w.Konto_Nr = 987654
    Or w.Konto_Nr = 745363
);

```

Die SQL-Abfrage für alle Kontennummern mit Kontostand, Art und Adresse der Zweigstelle sieht wie folgt aus.

Arbeitsblatt Query Builder

```

Select Bezeichnung,
       Strasse,
       HausNr,
       PLZ,
       Ort,
       deref(COLUMN_VALUE).Konto_Nr as KontoNr,
       deref(COLUMN_VALUE).Konto_Art as KontoArt,
       deref(COLUMN_VALUE).Konto_Stand as KontoStand
from (select w.Zweigstelle_v1_Name as Bezeichnung,
            deref(w.Zweigstelle_v1_Adresse).Adresse_Strasse as Strasse,
            deref(w.Zweigstelle_v1_Adresse).Adresse_Haus_Nr as HausNr,
            deref(w.Zweigstelle_v1_Adresse).Adresse_PLZ as PLZ,
            deref(w.Zweigstelle_v1_Adresse).Adresse_Ort as Ort,
            v.*
      from Zweigstelle_v1 w, table (w.Zweigstelle_v1_Konten_Liste) v );

```

Skriptausgabe x Abfrageergebnis x

SQL | Alle Zeilen abgerufen:5 in 0,039 Sekunden

	BEZEICHNUNG	STRASSE	HAUSNR	PLZ	ORT	KONTONR	KONTOART	KONTOSTAND
1	Bachdorf	Hochstr.	1	99999	Bachdorf	120768	S	234,56
2	Bachdorf	Hochstr.	1	99999	Bachdorf	348973	G	12567,56
3	Bachdorf	Hochstr.	1	99999	Bachdorf	678453	G	-456,78
4	Riedering	Simseestr.	3	88888	Riedering	745363	S	-23,67
5	Riedering	Simseestr.	3	88888	Riedering	987654	G	789,65

Bei der SQL-Abfrage für die Ermittlung der Paare von Kontonummern und Adressen von Kontoinhabern ist folgende Abfrage nötig.

Arbeitsblatt Query Builder

```

Select KundenNr,
        KundenName,
        KundenStatus,
        Strasse,
        HausNr,
        PLZ,
        Ort,
        deref(COLUMN_VALUE).Konto_Nr as KontoNr,
        deref(COLUMN_VALUE).Konto_Art as KontoArt,
        deref(COLUMN_VALUE).Konto_Stand as KontoStand
from (select w.Kunde_v1_Nr as KundenNr,
        w.Kunde_v1_Name as KundenName,
        w.Kunde_v1_Status as KundenStatus,
        deref(w.Kunde_v1_Adresse).Adresse_Strasse as Strasse,
        deref(w.Kunde_v1_Adresse).Adresse_Haus_Nr as HausNr,
        deref(w.Kunde_v1_Adresse).Adresse_PLZ as PLZ,
        deref(w.Kunde_v1_Adresse).Adresse_Ort as Ort,
        v.*
from Kunde_v1 w, table (w.Kunde_v1_Konten_Liste) v );

```

Skriptausgabe x Abfrageergebnis x

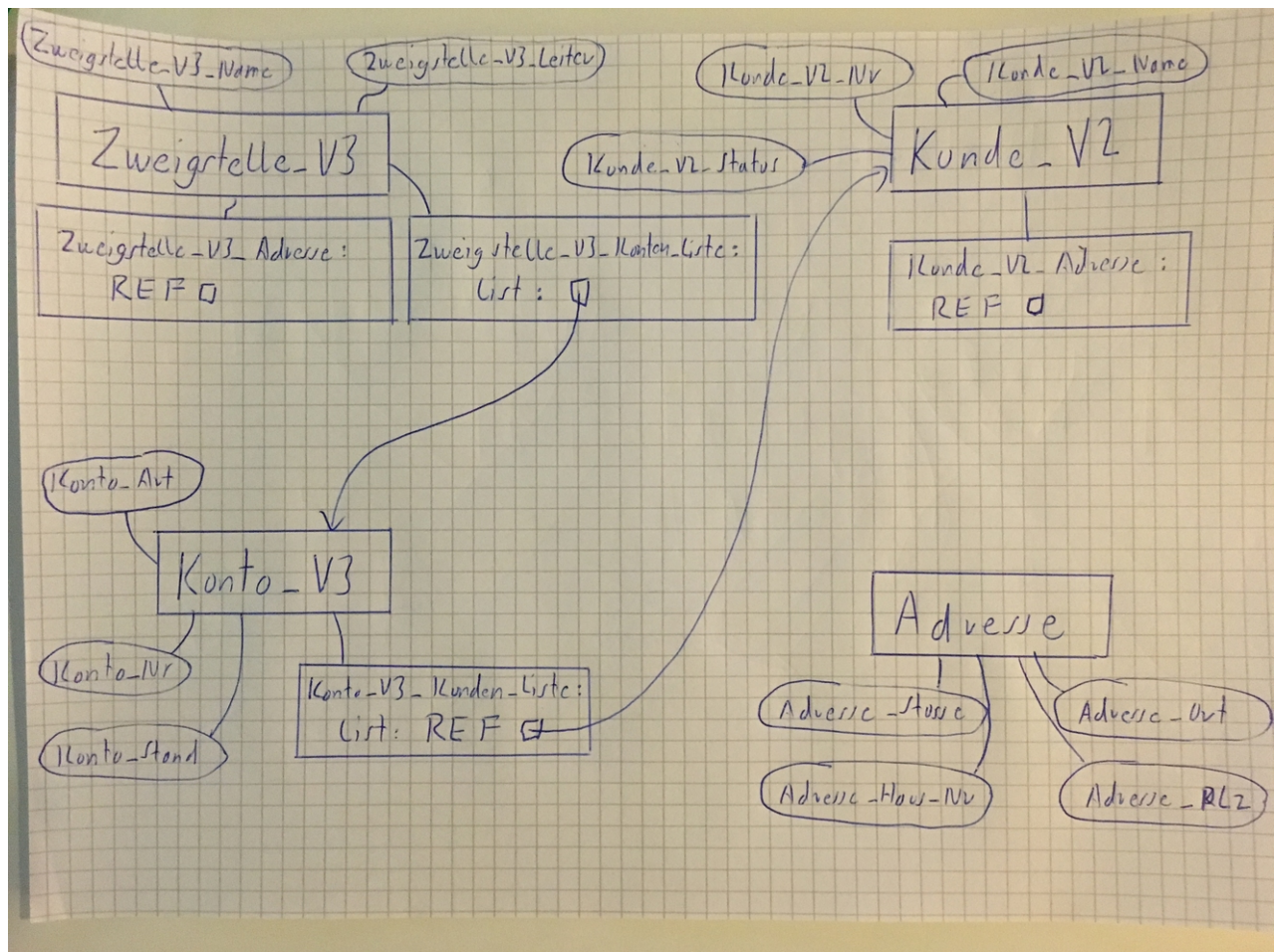
SQL | Alle Zeilen abgerufen: 6 in 0,042 Sekunden

	KUNDENNR	KUNDENNAME	KUNDENSTATUS	STRASSE	HAUSNR	PLZ	ORT	KONTONR	KONTOART	KONTOSTAND
1	2345	H. Fach	Geschäftskunde	Münchnersrt.	33	77777	Muenchen	120768	S	234,56
2	2345	H. Fach	Geschäftskunde	Münchnersrt.	33	77777	Muenchen	348973	G	12567,56
3	7654	B. Meier	Privatkunde	Eschenweg	12	66666	Landshut	987654	G	789,65
4	8764	J. Wiesner	Geschäftskunde	Schellingstr.	42	55555	Regensburg	348973	G	12567,56
5	8764	J. Wiesner	Geschäftskunde	Schellingstr.	42	55555	Regensburg	678453	G	-456,78
6	8764	J. Wiesner	Geschäftskunde	Schellingstr.	42	55555	Regensburg	745363	S	-23,67

Darstellung 2:

Die Übersichtszeichnung bei der zweiten Darstellungsweise zeigt, dass jede Zweigstelle eine Liste von dazugehörigen Konten besitzt welche direkt in dieser Liste gespeichert und nicht als Referenz eingefügt werden. Jedes Konto besitzt hingegen eine Liste von Referenzen zu den dazugehörigen Kunden.

Somit ist für die Konten keine eigene Tabelle nötig, da diese wie oben beschrieben nicht als Referenz gespeichert werden. Für den Kunden sind hingegen schon Tabellen erforderlich damit die Kunden darin abgespeichert werden können. Als Folge davon können die Kunden als Referenz in die Kundenliste der jeweiligen Konten hinterlegt werden.



In den folgenden Abbildungen werden die benötigten Tabellen aufgezeigt.

Startseite

DBS2

ZWEIGSTELLE_V3

Spalten

Daten

Model

Constraints

Zugriffsberechtigungen

Statistiken

Trigger

Abhängigkeiten

Details

Partitionen

Indizes

SQL

Aktionen...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ZWEIGSTELLE_V3_NAME	VARCHAR2(50 BYTE)	Yes	(null)	1	(null)
2	ZWEIGSTELLE_V3_LEITER	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	ZWEIGSTELLE_V3_ADRESSE	ADRESSET	Yes	(null)	3	(null)
4	ZWEIGSTELLE_V3_KONTEN_LISTE	KONTENLISTET_V3	Yes	(null)	4	(null)

Startseite

DBS2

KUNDE_V2

KONTENLISTET_V3

KONTOT_V3

KUNDENLISTET_V2

Spalten

Daten

Model

Constraints

Zugriffsberechtigungen

Statistiken

Trigger

Abhängigkeiten

Details

Partitionen

Indizes

SQL

Aktionen...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	KUNDE_V2_NR	NUMBER(4,0)	Yes	(null)	1 (null)	
2	KUNDE_V2_NAME	VARCHAR2(50 BYTE)	Yes	(null)	2 (null)	
3	KUNDE_V2_STATUS	VARCHAR2(25 BYTE)	Yes	(null)	3 (null)	
4	KUNDE_V2_ADRESSE	ADRESSET	Yes	(null)	4 (null)	

Da wie oben geschrieben keine Tabelle vom Type Konto für diese Darstellungsweise existiert, können die einzelnen Konten während des Anlegens der Zweigstellen in der Tabelle `Zweigstelle_V3` eingefügt werden. Das nachfolgende SQL-Statement veranschaulicht dies.


```

insert into Zweigstelle_v3 values (
    ZweigstelleT_v3( 'Bachdorf',
                    '1768',
                    (Select REF (w) from Adresse w
                     where w.Adresse_plz = 99999),
                    KontenListeT_v3( KontoT_v3( 'S',
                                                120768,
                                                234.56,
                                                KUNDENLISTET_V2()),
                                    KontoT_v3( 'G',
                                                348973,
                                                12567.56,
                                                KUNDENLISTET_V2()),
                                    KontoT_v3( 'G',
                                                678453,
                                                -456.78,
                                                KUNDENLISTET_V2())
                                    )
                    )
);

insert into Zweigstelle_v3 values (
    ZweigstelleT_v3( 'Riedering',
                    '9823',
                    (Select REF (w) from Adresse w
                     where w.Adresse_plz = 88888),
                    KontenListeT_v3( KontoT_v3( 'G',
                                                987654,
                                                789.65,
                                                KUNDENLISTET_V2()),
                                    KontoT_v3( 'S',
                                                745363,
                                                -23.67,
                                                KUNDENLISTET_V2())
                                    )
                    )
);|

```

Die Kunden in Tabelle Kunde_V2 werden wie in der ersten Darstellungsweise angelegt. Der Unterschied ist nur, dass beim Anlegen der Kunden für die zweite Darstellungsweise keine Kontenliste benötigt wird.

```

Arbeitsblatt | Query Builder
--
insert into Kunde_v2 values
(2345,
'H. Fach',
'Geschäftskunde',
(Select REF (w) from Adresse w
where w.ADRESSE_PLZ = 77777));

insert into Kunde_v2 values
(7654,
'B. Meier',
'Privatkunde',
(Select REF (w) from Adresse w
where w.ADRESSE_PLZ = 66666));

insert into Kunde_v2 values
(8764,
'J. Wiesner',
'Geschäftskunde',
(Select REF (w) from Adresse w
where w.ADRESSE_PLZ = 55555));

```

Als nächstes müssen in den Konten die dazugehörigen Kunden referenziert werden. Dies ist in der folgenden Darstellung aufgezeigt.

```

Arbeitsblatt | Query Builder
--
insert into table (Select v.Konto_v3_Kunden_Liste from Zweigstelle_v3 w, table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v
where v.Konto_Nr = 120768)
(Select REF (x) from Kunde_v2 x
where x.Kunde_v2_Nr = 2345);

insert into table (Select v.Konto_v3_Kunden_Liste from Zweigstelle_v3 w, table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v
where v.Konto_Nr = 348973)
(Select REF (x) from Kunde_v2 x
where x.Kunde_v2_Nr = 2345
or x.Kunde_v2_Nr = 8764);

insert into table (Select v.Konto_v3_Kunden_Liste from Zweigstelle_v3 w, table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v
where v.Konto_Nr = 987654)
(Select REF (x) from Kunde_v2 x
where x.Kunde_v2_Nr = 7654);

insert into table (Select v.Konto_v3_Kunden_Liste from Zweigstelle_v3 w, table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v
where v.Konto_Nr = 745363)
(Select REF (x) from Kunde_v2 x
where x.Kunde_v2_Nr = 8764);

insert into table (Select v.Konto_v3_Kunden_Liste from Zweigstelle_v3 w, table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v
where v.Konto_Nr = 678453)
(Select REF (x) from Kunde_v2 x
where x.Kunde_v2_Nr = 8764);

```

Die SQL-Abfrage für alle Kontennummern mit Kontostand, Art und Adresse der Zweigstelle ist nachfolgend aufgezeigt.

Arbeitsblatt

Query Builder

Select

w.ZWEIGSTELLE_V3_NAME as ZweigName,
w.ZWEIGSTELLE_V3_LEITER as Leiter,
deref(w.ZWEIGSTELLE_V3_ADRESSE).Adresse_Strasse as Strasse,
deref(w.ZWEIGSTELLE_V3_ADRESSE).Adresse_Haus_Nr as HausNr,
deref(w.ZWEIGSTELLE_V3_ADRESSE).Adresse_Plz as Plz,
deref(w.ZWEIGSTELLE_V3_ADRESSE).Adresse_Ort as Ort,
v.Konto_Art as KontoArt,
v.Konto_Nr as KontoNr,
v.Konto_Stand as KontoStand
from Zweigstelle_v3 w, table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v;

Skriptausgabe x

Abfrageergebnis x

SQL

Alle Zeilen abgerufen:5 in 0,009 Sekunden

	ZWEIGNAME	LEITER	STRASSE	HAUSNR	PLZ	ORT	KONTOART	KONTONR	KONTOSTAND
1	Bachdorf	1768	Hochstr.	1	99999	Bachdorf	S	120768	234,56
2	Bachdorf	1768	Hochstr.	1	99999	Bachdorf	G	348973	12567,56
3	Bachdorf	1768	Hochstr.	1	99999	Bachdorf	G	678453	-456,78
4	Riedering	9823	Simseestr.	3	88888	Riedering	G	987654	789,65
5	Riedering	9823	Simseestr.	3	88888	Riedering	S	745363	-23,67

Bei der SQL-Abfrage für die Ermittlung der Paare von Kontonummern und Adressen von Kontoinhabern ist folgende Abfrage nötig.

Arbeitsblatt

Query Builder

Select

deref(COLUMN_VALUE).KUNDE_V2_NR as KundenNr,

deref(COLUMN_VALUE).KUNDE_V2_NAME as KundenName,

deref(COLUMN_VALUE).KUNDE_V2_STATUS as Status,

deref(COLUMN_VALUE).KUNDE_V2_ADRESSE.ADRESSE_STRASSE as Strasse,

deref(COLUMN_VALUE).KUNDE_V2_ADRESSE.ADRESSE_HAUS_NR as HausNr,

deref(COLUMN_VALUE).KUNDE_V2_ADRESSE.ADRESSE_ORT as Ort,

deref(COLUMN_VALUE).KUNDE_V2_ADRESSE.ADRESSE_PLZ as Plz,

KontoArt,

KontoNr

KontoStand

from

(Select

v.Konto_Art as KontoArt,

v.Konto_Nr as KontoNr,

v.Konto_Stand as KontoStand,

x.*

from Zweigstelle_v3 w,

table (w.ZWEIGSTELLE_V3_KONTEN_LISTE) v,

table (v.Konto_v3_Kunden_Liste) x);

Skriptausgabe x

Abfrageergebnis x

SQL

Alle Zeilen abgerufen:6 in 0,013 Sekunden

	KUNDENNR	KUNDENNAME	STATUS	STRASSE	HAUSNR	ORT	PLZ	KONTOART	KONTOSTAND
1	2345	H. Fach	Geschäftskunde	Münchnersrt.	33	Muenchen	77777	S	120768
2	2345	H. Fach	Geschäftskunde	Münchnersrt.	33	Muenchen	77777	G	348973
3	8764	J. Wiesner	Geschäftskunde	Schellingstr.	42	Regensburg	55555	G	348973
4	7654	B. Meier	Privatkunde	Eschenweg	12	Landshut	66666	G	987654
5	8764	J. Wiesner	Geschäftskunde	Schellingstr.	42	Regensburg	55555	S	745363
6	8764	J. Wiesner	Geschäftskunde	Schellingstr.	42	Regensburg	55555	G	678453

Teilaufgabe 2: Retailer Case Study

Das objektrationale Datenbankschema ist mit den entsprechenden Datentypen (beginnend mit CS_...) und Tabelle (beginnend mit CS_...) in der Datenbank DBST49 angelegt.

