Fachbereich 07 Informatik/Mathematik

Praktikum Datenbanksysteme II
Wintersemester 2018/19

Prof. Dr. Martin Staudt

Übung 2

Wimmer, Anja
IF8
Gabl, Daniel
IF6

04.12.2018

Inhaltsverzeichnis

Aufgaben

Aufgabe 1

-

Aufgabe 2

Die zwei Möglichkeiten, die wir in Betracht ziehen:

1. Allmögliche Objekt-Typen definieren und gegenseitig referenzieren. Bspw. könnte die Adresse ein Typ sein, die sich aus der Straße und der Hausnummer (und ggf. PLZ und Ort) zusammensetzt. Auch könnte ein Kontotyp mit Konto-Nr., Kontostand, Art und ID der Zweigstelle ein eigenes Attribut sein. (Es ist keine Zuordnungstabelle erforderlich, da es sich bei Zweigstelle <-> Konto um eine 1:n-Beziehung handelt.)
2. Beispielsweiße könnten wir den Konto-Typ nicht als eigene Tabelle speichern sondern als innere Tabelle beim Zweigstellen-Typs speichern, dadurch entfällt die Referenz auf diese Tabelle.

Aufgabe 3

Wir würden folgendes Schema aufstellen: (Legende: 1. Möglichkeit, **2. Möglichkeit**, beide)

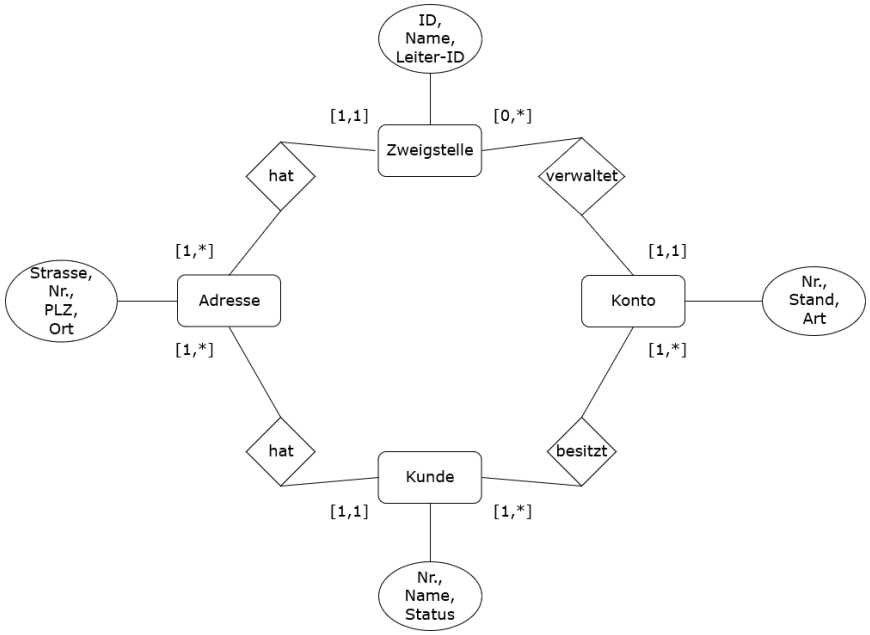
5 Typen wie folgt:

- Adress-Typ mit Straße und Hausnummer (und ggf. PLZ und Ort)
- **Kontolisten-Typ als Tabelle vom Typ Integer (Kontonummern)**
- Kunden-Typ mit Kunden-Nr., -Name, Adress-Typ, Status **und Kontolisten-Typ**
- Kontoinhaber-Typ als Tabelle vom Kunden-Typ
- **Zweigstellenkonten-Typ als Tabelle von Konto-Typen**
- Zweigstellen-Typ mit Zweigstellenname, Adress-Typ, Leiter-Id **und Zweigkonten**
- Konto-Typ mit Konto-Nr., Kontostand, Art, Kontoinhaber-Typ und Zweigstellen-Typ

Dazu noch folgende Tabellen:

- Kunden-Tabelle mit Kunden-Typ
- Zweigstellen-Adresse mit Zweigstellen-Typ
- Konto-Tabelle mit Konto-Typ **(entfällt bei der 2. Möglichkeit)**

Eine Skizzierung des Zusammenspiels der Typen und Tabellen:



SQL-Statements zum Erzeugen der Typen und Tabellen **(1. Möglichkeit)**:

```
CREATE TYPE AddressType AS Object(street VARCHAR(31), houseNr
VARCHAR(7), zip INT(5), place VARCHAR(31));
/
CREATE TYPE CustomerType AS Object(customerNr INT, customerName
VARCHAR(63), addr AddressType, status VARCHAR(15));
/
CREATE TYPE AccountOwnerType AS TABLE OF REF CustomerType;
/
CREATE TYPE BranchOfficeType AS Object(branchOfficeName VARCHAR(63),
addr AddressType, leaderId INT);
/
CREATE TYPE AccountType AS Object(accountNr INT,
balance DOUBLE PRECISION, kind VARCHAR(1),
owners AccountOwnerType, branchOffice REF BranchOfficeType);
/
CREATE TABLE Customer OF CustomerType;
CREATE TABLE BranchOffice OF BranchOfficeType;
CREATE TABLE AccountTable OF AccountType NESTED TABLE owners STORE
AS lorem_ipsum;
```

SQL-Statements zum Einfügen von Beispieldatensätzen in die Datenbank **(1. Möglichkeit)**:

```
INSERT INTO Customer VALUES(CustomerType(2345, 'H. Fach',
AddressType('Münchenerstr.', '33', 60329, ' Frankfurt am Main'),
'Geschäftskunde'));

INSERT INTO Customer VALUES(CustomerType(7654, 'B. Meier',
AddressType('Eschenweg', '12', 85354, 'Freising'), 'Privatkunde'));

INSERT INTO Customer VALUES(CustomerType(8764, 'J. Wiesner',
AddressType('Schellingstr.', '42', 80799, 'München'),
'Geschäftskunde'));


INSERT INTO BranchOffice VALUES(BranchOfficeType('Bachdorf',
AddressType('Hochstr.', '1', 81669, 'München'), 1768));

INSERT INTO BranchOffice VALUES(BranchOfficeType('Riedering',
AddressType('Simseestr.', '3', 81549, 'München'), 9823));


INSERT INTO AccountTable VALUES(AccountType(120768, 234.56, 'S',
AccountOwnerType((SELECT REF(c) FROM Customer c WHERE c.customerNr =
2345)), (SELECT REF(b) FROM BranchOffice b WHERE b.branchOfficeName =
'Bachdorf')));

INSERT INTO AccountTable VALUES(AccountType(678453, -456.78, 'G',
AccountOwnerType((SELECT REF(c) FROM Customer c WHERE c.customerNr =
8764)), (SELECT REF(b) FROM BranchOffice b WHERE b.branchOfficeName =
'Bachdorf')));

INSERT INTO AccountTable VALUES(AccountType(348973, 12567.56, 'G',
AccountOwnerType((SELECT REF(c) FROM Customer c WHERE c.customerNr =
2345), (SELECT REF(c) FROM Customer c WHERE c.customerNr = 8764)),
(SELECT REF(b) FROM BranchOffice b WHERE b.branchOfficeName =
'Bachdorf')));

INSERT INTO AccountTable VALUES(AccountType(987654, 789.65, 'G',
AccountOwnerType((SELECT REF(c) FROM Customer c WHERE c.customerNr =
7654)), (SELECT REF(b) FROM BranchOffice b WHERE b.branchOfficeName =
'Riedering')));

INSERT INTO AccountTable VALUES(AccountType(745363, -23.67, 'S',
AccountOwnerType((SELECT REF(c) FROM Customer c WHERE c.customerNr =
8764)), (SELECT REF(b) FROM BranchOffice b WHERE b.branchOfficeName =
'Riedering')));
```

SQL-Statements zum Erzeugen der Typen und Tabellen **(2. Möglichkeit)**:

```
CREATE TYPE AddressType AS Object(street VARCHAR(31), houseNr
VARCHAR(7), zip INT(5), place VARCHAR(31));
/
CREATE TYPE AccountsT AS TABLE OF INT;
/
CREATE TYPE CustomerType AS Object(customerNr INT, customerName
VARCHAR(63), addr AddressType, status VARCHAR(15), accNr AccountsT);
/
CREATE TYPE AccountType AS Object(accountNr INT,
balance DOUBLE PRECISION, kind VARCHAR(1));
/
CREATE TYPE BranchAccountsType AS TABLE OF AccountType;
/
CREATE TYPE BranchOfficeType AS Object(branchOfficeName VARCHAR(63),
addr AddressType, leaderId INT, accounts BranchAccountsType);
/

CREATE TABLE Customer OF CustomerType
NESTED TABLE accNr STORE AS accNr_useless;

CREATE TABLE BranchOffice OF BranchOfficeType
NESTED TABLE accounts STORE AS accounts_useless;
```

SQL-Statements zum Einfügen von Beispieldatensätzen in die Datenbank **(2. Möglichkeit)**:

```
INSERT INTO Customer
VALUES(CustomerType(2345, 'H. Fach',
AddressType('Münchenerstr.', '33', 60329, 'Frankfurt am Main'),
'Geschäftskunde', AccountsT(120768, 348973)));

INSERT INTO Customer
VALUES(CustomerType(7654, 'B. Meier',
AddressType('Eschenweg', '12', 85354, 'Freising'), 'Privatkunde',
AccountsT(987654)));

INSERT INTO Customer
VALUES(CustomerType(8764, 'J. Wiesner',
AddressType('Schellingstr.', '42', 80799, 'München'), 'Geschäftskunde',
AccountsT(745363, 678453, 348973)));

INSERT INTO BranchOffice
VALUES(BranchOfficeType('Bachdorf', AddressType('Hochstr.', '1', 81669,
'München'), 1768, BranchAccountsType()));

INSERT INTO TABLE(SELECT accounts FROM BranchOffice WHERE
branchOfficeName='Bachdorf')
VALUES(AccountType(120768, 234.56, 'S'));

INSERT INTO TABLE(SELECT accounts FROM BranchOffice WHERE
branchOfficeName='Bachdorf')
VALUES(AccountType(678453, -456.78, 'G'));

INSERT INTO TABLE(SELECT accounts FROM BranchOffice WHERE
branchOfficeName='Bachdorf')
VALUES(AccountType(348973, 12567.56, 'G'));

INSERT INTO BranchOffice
VALUES(BranchOfficeType('Riedering', AddressType('Simseestr.', '3',
81549, 'München'), 9823, BranchAccountsType()));

INSERT INTO TABLE(SELECT accounts FROM BranchOffice WHERE
branchOfficeName='Riedering')
VALUES(AccountType(987654, 789.65, 'G'));

INSERT INTO TABLE(SELECT accounts FROM BranchOffice WHERE
branchOfficeName='Riedering')
VALUES(AccountType(745363, -23.67, 'S'));
```

Aufgabe 4

Bei der 1. Möglichkeit:

```
a) SELECT a.accountNr, a.balance, a.kind,
   CONCAT(CONCAT(DEREF(a.branchOffice).addr.street, ' '),
   DEREF(a.branchOffice).addr.houseNr) AS addr FROM AccountTable a;

b) SELECT a.accountNr, DEREF(o.COLUMN_VALUE).customerName AS
   customerName, CONCAT(CONCAT(DEREF(o.COLUMN_VALUE).addr.street, '
   '), DEREF(o.COLUMN_VALUE).addr.houseNr) as addr FROM AccountTable
   a, TABLE(a.owners) o;
```

Bei der 2. Möglichkeit:

```
a) SELECT a.accountNr, a.balance, a.kind,
   CONCAT(CONCAT(b.addr.street, ' '), b.addr.houseNr) AS addr
   FROM BranchOffice b, TABLE(b.accounts) a;

b) SELECT a.COLUMN_VALUE,
   CONCAT(CONCAT(c.addr.street, ' '),c.addr.houseNr) AS addr
   FROM Customer c, TABLE(c.accNr) a;
```

Screendumps

Hier Screendumps unserer Tabellen und den Ergebnissen aus Aufgabe 4.

1. Möglichkeit:

Screendump von Aufgabe 4a) (Kontonummer, -stand, -art und Adresse der Zweigstelle):

	ACCOUNTNR	BALANCE	KIND	ADDR
1	120768	234,56	S	Hochstr. 1
2	678453	-456,78	G	Hochstr. 1
3	348973	12567,56	G	Hochstr. 1
4	987654	789,65	G	Simseestr. 3
5	745363	-23,67	S	Simseestr. 3

Screendump von Aufgabe 4b) (Paare von Kontonummern, Namen und Adressen der Inhaber):

	ACCOUNTNR	CUSTOMERNAME	ADDR
1	120768	H. Fach	Münchenerstr. 33
2	678453	J. Wiesner	Schellingstr. 42
3	348973	H. Fach	Münchenerstr. 33
4	348973	J. Wiesner	Schellingstr. 42
5	987654	B. Meier	Eschenweg 12
6	745363	J. Wiesner	Schellingstr. 42

Zweigstellen-Tabelle:

	BRANCHOFFICENAME	ADDR	LEADERID
1	Bachdorf	[DBST42.ADDRESSTYPE]	1768
2	Riedering	[DBST42.ADDRESSTYPE]	9823

Kunden-Tabelle:

	CUSTOMERNR	CUSTOMERNAME	ADDR	STATUS
1	2345	H. Fach	[DBST42.ADDRESSTYPE]	Geschäftskunde
2	7654	B. Meier	[DBST42.ADDRESSTYPE]	Privatkunde
3	8764	J. Wiesner	[DBST42.ADDRESSTYPE]	Geschäftskunde

Konten-Tabelle:

	ACCOUNTNR	BALANCE	KIND	OWNERS	BRANCHOFFICE
1	120768	234,56	S	DBST42.ACCOUNTOWNERTYPE([DBST42.CUSTOMERTYPE])	[DBST42.BRANCHOFFICETYPE]
2	678453	-456,78	G	DBST42.ACCOUNTOWNERTYPE([DBST42.CUSTOMERTYPE])	[DBST42.BRANCHOFFICETYPE]
3	348973	12567,56	G	DBST42.ACCOUNTOWNERTYPE([DBST42.CUSTOMERTYPE],[DBST42.CUSTOMERTYPE])	[DBST42.BRANCHOFFICETYPE]
4	987654	789,65	G	DBST42.ACCOUNTOWNERTYPE([DBST42.CUSTOMERTYPE])	[DBST42.BRANCHOFFICETYPE]
5	745363	-23,67	S	DBST42.ACCOUNTOWNERTYPE([DBST42.CUSTOMERTYPE])	[DBST42.BRANCHOFFICETYPE]

2. Möglichkeit:

Screendump von Aufgabe 4a)

	ACCOUNTNR	BALANCE	KIND	ADDR
1	120768	234,56	S	Hochstr. 1
2	678453	-456,78	G	Hochstr. 1
3	348973	12567,56	G	Hochstr. 1
4	987654	789,65	G	Simseestr. 3
5	745363	-23,67	S	Simseestr. 3

Screendump von Aufgabe 4b)

	COLUMN_VALUE	ADDR
1	120768	Münchenerstr. 33
2	348973	Münchenerstr. 33
3	987654	Eschenweg 12
4	745363	Schellingstr. 42
5	678453	Schellingstr. 42
6	348973	Schellingstr. 42

Zweigstellen-Tabelle:

	BRANCHOFFICENAME	ADDR	LEADERID	ACCOUNTS
1	Bachdorf	[DBST42.ADDRESSTYPE]	1768	DBST42.BRANCHACCOUNTSTYPE([DBST42.ACCOUNTTYPE],[DBST42.ACCOUNTTYPE],[DBST42.ACCOUNTTYPE])
2	Riedering	[DBST42.ADDRESSTYPE]	9823	DBST42.BRANCHACCOUNTSTYPE([DBST42.ACCOUNTTYPE],[DBST42.ACCOUNTTYPE])

Kunden-Tabelle:

	CUSTOMERNR	CUSTOMERNAME	ADDR	STATUS	ACCNR
1	2345	H. Fach	[DBST42.ADDRESSTYPE]	Geschäftskunde	DBST42.ACCOUNTST(120768,348973)
2	7654	B. Meier	[DBST42.ADDRESSTYPE]	Privatkunde	DBST42.ACCOUNTST(987654)
3	8764	J. Wiesner	[DBST42.ADDRESSTYPE]	Geschäftskunde	DBST42.ACCOUNTST(745363,678453,348973)

Aufgabe 5 (Fallstudie)

Zuerst haben wir für die Fallstudie die Aufgabenstellung analysiert und angefangen, Typen zu definieren um die Tabellen vollständig korrekt zu speichern.
Hierfür haben wir 30 Typen (Normale und List-Typen) angelegt, die wie folgt aussehen:

```
CREATE TYPE CampusT AS OBJECT (campus_location VARCHAR(15),
campus_addr VARCHAR(127), campus_phone VARCHAR(15), campus_fax
VARCHAR(15), campus_head VARCHAR(31));
/
CREATE TYPE ProfessorT AS OBJECT (prof_id INTEGER, prof_name
VARCHAR(31), prof_contact VARCHAR(31), prof_research
VARCHAR(63), prof_year INTEGER);
/
CREATE TYPE ProfessorListT AS TABLE OF REF ProfessorT;
/
CREATE TYPE DepartmentT AS OBJECT (dept_id VARCHAR(3),
dept_name VARCHAR(31), dept_head VARCHAR(31), dept_prof
ProfessorListT);
/
CREATE TYPE DepartmentListT AS TABLE OF DepartmentT;
/
CREATE TYPE SchoolT AS OBJECT (school_id VARCHAR(3),
school_name VARCHAR(31), school_head VARCHAR (31), school_prof
ProfessorListT);
/
CREATE TYPE SchoolListT AS TABLE OF SchoolT;
/
CREATE TYPE RCUnitT AS TABLE OF VARCHAR(127);
/
CREATE TYPE ResearchCentreT AS OBJECT (rc_id VARCHAR(3),
rc_name VARCHAR(127), rc_head VARCHAR(31), rc_unit RCUnitT);
/
CREATE TYPE ResearchCentreListT AS TABLE OF ResearchCentreT;
/
-- TODO aggregation clustering technique
CREATE TYPE FacultyT AS OBJECT (fac_id INTEGER, fac_name
VARCHAR(31), fac_dean VARCHAR(15), dept DepartmentListT,
school SchoolListT, rc ResearchCentreListT);
/
CREATE TYPE BuildingT AS OBJECT (bld_id VARCHAR(4), bld_name
VARCHAR(31), bld_location VARCHAR(2), bld_level INTEGER,
campus REF CampusT, fac REF FacultyT, MEMBER PROCEDURE
show_bld_details));
/
CREATE TYPE PersonT AS OBJECT (person_id VARCHAR(8),
person_surname VARCHAR(15), person_forename VARCHAR(15),
person_title VARCHAR(7), person_addr VARCHAR(127),
person_phone VARCHAR(15), person_postcode VARCHAR(5), campus
REF CampusT) NOT FINAL;
/
```

```
CREATE TYPE OfficeT AS OBJECT (office_No VARCHAR(7), bld REF
BuildingT, office_phone VARCHAR(15));
/
CREATE TYPE ClassroomT AS OBJECT (class_no VARCHAR(4), bld REF
BuildingT, class_capacity INTEGER);
/
CREATE TYPE LabEquipmentT AS TABLE OF VARCHAR(15);
/
CREATE TYPE LabT AS OBJECT (lab_no VARCHAR(5), bld REF
BuildingT, lab_capacity INTEGER, lab_equipment LabEquipmentT);
/
CREATE TYPE DegreeT AS OBJECT (deg_id VARCHAR(4), deg_name
VARCHAR(31), deg_length INTEGER, deg_prereq VARCHAR(31), fac
REF FacultyT);
/
CREATE TYPE ComputerskillsT AS TABLE OF VARCHAR(15);
/
CREATE TYPE OfficeskillsT AS TABLE OF VARCHAR(31);
/
CREATE TYPE TechnicianskillsT AS TABLE OF VARCHAR(15);
/
CREATE TYPE StaffT UNDER PersonT (office_No VARCHAR(7),
staff_type VARCHAR(15)) NOT FINAL;
/
CREATE TYPE StudentT UNDER PersonT (student_year INTEGER,
MEMBER PROCEDURE insert_student, MEMBER PROCEDURE
delete_student);
/
CREATE TYPE AdminT UNDER StaffT (admin_title VARCHAR(31),
admin_computerskills ComputerskillsT, admin_officeskills
OfficeskillsT);
/
CREATE TYPE TechnicianT UNDER StaffT (tech_title VARCHAR(31),
tech_skills TechnicianskillsT);
/
CREATE TYPE TutorT UNDER StaffT (tutor_hours INTEGER,
tutor_rate DOUBLE PRECISION);
/
CREATE TYPE LecturerT UNDER StaffT (lect_area VARCHAR(31),
lect_type VARCHAR(15)) NOT FINAL;
/
CREATE TYPE SeniorLecturerT UNDER LecturerT (senlect_phd
INTEGER, senlect_master INTEGER, senlect_honours INTEGER);
/
CREATE TYPE AssociateLecturerT UNDER LecturerT
(asslect_honours INTEGER, asslect_year INTEGER);
/
CREATE TYPE SubjectT AS OBJECT (subject_id VARCHAR(8),
subject_name VARCHAR(31), subject_credit INTEGER,
subject_prereq VARCHAR(8), person REF PersonT);
/
```

Dazu haben wir noch folgende 20 Tabellen angelegt, um konkrete Exemplare zu speichern:

```
CREATE TABLE Campus OF CampusT;

CREATE TABLE Professor OF ProfessorT;

CREATE TABLE Faculty OF FacultyT
     NESTED TABLE dept STORE AS department_nm (NESTED TABLE
dept_prof STORE AS dept_prof_nm)
     NESTED TABLE school STORE AS school_nm(NESTED TABLE
school_prof STORE AS school_prof_nm)
     NESTED TABLE rc STORE AS rc_nm (NESTED TABLE rc_unit STORE
AS rc_unit_nm);

CREATE TABLE Building OF BuildingT;

CREATE TABLE Person OF PersonT;

CREATE TABLE Office OF OfficeT;

CREATE TABLE Classroom OF ClassroomT;

CREATE TABLE Lab OF LabT
     NESTED TABLE lab_equipment STORE AS lab_equip_nm;

CREATE TABLE DegreeTbl OF DegreeT;

CREATE TABLE Staff OF StaffT;

CREATE TABLE Student OF StudentT;

CREATE TABLE AdminTbl OF AdminT
     NESTED TABLE admin_computerskills STORE AS adm_comskill_nm
     NESTED TABLE admin_officeskills STORE AS adm_offskill_nm;

CREATE TABLE Technician OF TechnicianT
     NESTED TABLE tech_skills STORE AS tech_skill_nm;

CREATE TABLE Tutor OF TutorT;

CREATE TABLE Lecturer OF LecturerT;

CREATE TABLE SeniorLecturer OF SeniorLecturerT;

CREATE TABLE AssociateLecturer OF AssociateLecturerT;

CREATE TABLE Subject OF SubjectT;

CREATE TABLE Enrolls_in (student REF StudentT, deg REF
DegreeT);

CREATE TABLE Takes (student REF StudentT, subject REF
SubjectT, mark INTEGER);
```

Dazu haben wir noch die im Diagramm eingezeichneten Funktionen wir folgt implementiert:

```
CREATE OR REPLACE FUNCTION show_bld_details(in_bld_id IN
VARCHAR2)
   RETURN VARCHAR2
   IS building_details VARCHAR2(255);

BEGIN

SELECT 'ID: '|| building.bld_id ||', Name: '|| building.bld_name
||', Location: '|| building.bld_location ||', Level: '||
building.bld_level ||', Campus: '||
DEREF(building.campus).campus_location ||', Faculty:
'||DEREF(building.fac).fac_name
  INTO building_details
  FROM Building building
  WHERE building.bld_id = in_bld_id;

  RETURN(building_details);

END show_bld_details;
-- SELECT show_bld_details('BB1') AS "Building Info" FROM DUAL;

-- not working correctly, declaration would be needed
CREATE OR REPLACE FUNCTION insert_student(in_person_id IN
VARCHAR2, in_year IN NUMBER)
   RETURN VARCHAR2
   IS success_state VARCHAR2(5);  -- boolean type not supported

BEGIN

INSERT INTO Student (
  SELECT pers.*, in_year FROM Person pers
  WHERE pers.person_id = in_person_id
);
success_state := 'TRUE';
RETURN(success_state);

END insert_student;


CREATE OR REPLACE FUNCTION delete_student(in_person_id IN
VARCHAR2)
   RETURN VARCHAR2
   IS success_state VARCHAR2(5);  -- boolean type not supported

BEGIN

DELETE FROM Student student WHERE student.person_id =
in_person_id;
success_state := 'TRUE';
RETURN(success_state);

END delete_student;
```

Schlussendlich haben wir mittels folgenden Insert-Statements Daten in die Tabellen angelegt:

```
INSERT INTO Campus
VALUES (CampusT('Albury/Wodonga', 'Parkers Road Wodonga VIC
3690',
          '61260583700', '620260583777', 'John Hill'));
INSERT INTO Campus
VALUES (CampusT('City', '215 Franklin St. Melb VIC 3000',
          '61392855100', '610392855111', 'Michael A.
O''Leary'));
INSERT INTO Campus
VALUES (CampusT('Mildura', 'Benetook Ave. Mildura VIC 3502',
          '61350223757', '61350223646', 'Ron Broadhead'));
INSERT INTO Campus
VALUES (CampusT('Bundoora', '221b Baker St. London NW1',
          '6195135755', '6137196482', 'Sherlock Holmes'));

INSERT INTO Faculty
VALUES (FacultyT(1, 'Health Science', 'S. Duckett',
          DepartmentListT(), SchoolListT(),
ResearchCentreListT()));
INSERT INTO Faculty
VALUES (FacultyT(2, 'Humanity '||'&'||' Social Sc.', 'J. A.
Salmond',
          DepartmentListT(), SchoolListT(),
ResearchCentreListT()));
INSERT INTO Faculty
VALUES (FacultyT(3, 'Law '||'&'||' Management', 'G. C.
O''Brien',
          DepartmentListT(), SchoolListT(),
ResearchCentreListT()));
INSERT INTO Faculty
VALUES (FacultyT(4, 'Science, Tech. '||'&'||' Eng.', 'D.
Finlay',
          DepartmentListT(), SchoolListT(),
ResearchCentreListT()));
INSERT INTO Faculty
VALUES (FacultyT(5, 'Regional Department', 'L. Kilmartin',
          DepartmentListT(), SchoolListT(),
ResearchCentreListT()));

INSERT INTO Professor VALUES (ProfessorT(42, 'Nick
Hoogenraad', 'hoogenraad@cu.edu', 'Advanced Software
Engineering', 2007));
INSERT INTO Professor VALUES (ProfessorT(88, 'Robin Anders',
'robin.anders@cu.edu', 'Computer Architecture', 2017));
INSERT INTO Professor VALUES (ProfessorT(101, 'Claude
Bernard', 'cbernard@cu.edu', 'Networking', 2016));
INSERT INTO Professor VALUES (ProfessorT(420, 'Bruce Stone',
'b.stone@cu.edu', 'Software Development', 2015));
INSERT INTO Professor VALUES (ProfessorT(555, 'Chris Handley',
'handley@cu.edu', 'Compiler', 2009));
```

```
INSERT INTO Professor VALUES (ProfessorT(782, 'Sheena Reilly',
'sheenareilly@cu.edu', 'Discrete Mathematics', 2005));
INSERT INTO Professor VALUES (ProfessorT(1001, 'Alison Perry',
'Alison.Perry@cu.edu', 'Physics', 2012));
INSERT INTO Professor VALUES (ProfessorT(1337, 'Jan Branson',
'branson@cu.edu', 'Software Architecture', 2018));

INSERT INTO TABLE (SELECT dept FROM Faculty WHERE fac_id=4)
VALUES(DepartmentT('4-1', 'Agricultural Sciences', 'Mark
Sandeman',
          ProfessorListT()));
INSERT INTO TABLE (SELECT dept FROM Faculty WHERE fac_id=4)
VALUES(DepartmentT('4-2', 'Biochemistry', 'Nick Hoogenraad',
          ProfessorListT()));

INSERT INTO TABLE (SELECT school FROM Faculty WHERE fac_id=1)
VALUES(SchoolT('1-1', 'Human Biosciences', 'Chris Handley',
          ProfessorListT((SELECT REF(p) FROM Professor p WHERE
p.prof_id=555))));
INSERT INTO TABLE (SELECT school FROM Faculty WHERE fac_id=1)
VALUES(SchoolT('1-2', 'Human Comm. Sciences', 'Elizabeth
Lavender',
          ProfessorListT()));

INSERT INTO TABLE (SELECT rc FROM Faculty WHERE fac_id=1)
VALUES(ResearchCentreT('1-1', 'Australian Research Centre in
Sex, Health '||'&'||' Society', 'Martin Pitts', RCUnitT()));
INSERT INTO TABLE (SELECT rc FROM Faculty WHERE fac_id=1)
VALUES(ResearchCentreT('1-2', 'Australian Institute for
Primary Care', 'Hal Swerissen', RCUnitT()));

INSERT INTO TABLE(SELECT dept_prof FROM TABLE(SELECT dept FROM
Faculty WHERE fac_id=4) dep WHERE dep.dept_id = '4-2') (SELECT
REF(p) FROM Professor p WHERE p.prof_id=42 OR p.prof_id=88 OR
p.prof_id=101 OR p.prof_id=420);

INSERT INTO TABLE(SELECT school_prof FROM TABLE(SELECT school
FROM Faculty WHERE fac_id=1) scho WHERE scho.school_id = '1-
2') (SELECT REF(p) FROM Professor p WHERE p.prof_id=782 OR
p.prof_id=1001 OR p.prof_id=1337);

INSERT INTO TABLE (SELECT rc_unit FROM TABLE (SELECT rc FROM
Faculty WHERE fac_id=1) r WHERE r.rc_id='1-1')
VALUES('SSAY Projects');
INSERT INTO TABLE (SELECT rc_unit FROM TABLE (SELECT rc FROM
Faculty WHERE fac_id=1) r WHERE r.rc_id='1-1')
VALUES('HIV Futures');
INSERT INTO TABLE (SELECT rc_unit FROM TABLE (SELECT rc FROM
Faculty WHERE fac_id=1) r WHERE r.rc_id='1-1')
VALUES('Australian Study of Health and Relationships');
```

```
INSERT INTO TABLE (SELECT rc_unit FROM TABLE (SELECT rc FROM
Faculty WHERE fac_id=1) r WHERE r.rc_id='1-2')
VALUES('Centre for Dev. and Innovation in Health');
INSERT INTO TABLE (SELECT rc_unit FROM TABLE (SELECT rc FROM
Faculty WHERE fac_id=1) r WHERE r.rc_id='1-2')
VALUES('Centre for Quality in Health '||'&'||' Community
Svc.');
INSERT INTO TABLE (SELECT rc_unit FROM TABLE (SELECT rc FROM
Faculty WHERE fac_id=1) r WHERE r.rc_id='1-2')
VALUES('Lincoln Gerontology Centre');

INSERT INTO Building
VALUES (BuildingT('BB1', 'Beth Gleeson', 'D5', 4,
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora'),
          (SELECT REF(f) FROM Faculty f WHERE f.fac_id=4)));
INSERT INTO Building
VALUES (BuildingT('BB2', 'Martin Building', 'F5', 4,
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora'),
          (SELECT REF(f) FROM Faculty f WHERE f.fac_id=3)));
INSERT INTO Building
VALUES (BuildingT('BB3', 'Thomas Cherry', 'D4', 4,
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora'),
          (SELECT REF(f) FROM Faculty f WHERE f.fac_id=1)));
INSERT INTO Building
VALUES (BuildingT('BB4', 'Physical Science 1', 'D5', 3,
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora'),
          (SELECT REF(f) FROM Faculty f WHERE f.fac_id=4)));

INSERT INTO Office
VALUES (OfficeT('BG207',
          (SELECT REF(b) FROM Building b WHERE
b.bld_id='BB4'),
          '94791118'));
INSERT INTO Office
VALUES (OfficeT('BG208',
          (SELECT REF(b) FROM Building b WHERE
b.bld_id='BB4'),
          '94792393'));

INSERT INTO Classroom
VALUES (ClassroomT('TCLT',
          (SELECT REF(b) FROM Building b WHERE
b.bld_id='BB3'),
          50));
INSERT INTO Classroom
VALUES (ClassroomT('TC01',
          (SELECT REF(b) FROM Building b WHERE
b.bld_id='BB3'),
          30));
```

```
INSERT INTO Lab
VALUES (LabT('BG113',
          (SELECT REF(b) FROM Building b WHERE
b.bld_id='BB1'),
          25, LabEquipmentT('25 PC', '1 Printer')));
INSERT INTO Lab
VALUES (LabT('BG114',
          (SELECT REF(b) FROM Building b WHERE
b.bld_id='BB1'),
          20, LabEquipmentT('21 PC')));

INSERT INTO DegreeTbl
VALUES (DegreeT('D100', 'Bachelor of Comp. Sci', 3, 'Year 12
or equivalent',
          (SELECT REF(f) FROM Faculty f WHERE f.fac_id=4)));
INSERT INTO DegreeTbl
VALUES (DegreeT('D101', 'Master of Comp. Sci', 2, 'Bachelor of
Comp. Sci',
          (SELECT REF(f) FROM Faculty f WHERE f.fac_id=4)));

INSERT INTO Person
VALUES (PersonT('01234234', 'Grant', 'Felix', 'Mr', '2 Boadle
Rd Bundoora VIC', '0398548753', '3083',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora')));
INSERT INTO Person
VALUES (PersonT('10008895', 'Xin', 'Harry', 'Mr', '6 Kelley St
Kew VIC',
          '0398875542', '3088',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora')));
INSERT INTO Person
VALUES (PersonT('10002935', 'Jones', 'Felicity', 'Ms', '14
Rennie St Thornbury VIC', '0398722001', '3071',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Bundoora')));
-- additional People needed for Staff
INSERT INTO Person
VALUES (PersonT('01958652', 'Doe', 'John', 'Mr', '64 Austin St
Holow VIC', '0321343123', '1337',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='City')));
INSERT INTO Person
VALUES (PersonT('10008957', 'Jane', 'Patrick', 'Mr', '23
Rainbow Rd Allumy VIC', '0236263636', '3033',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='City')));
INSERT INTO Person
VALUES (PersonT('10005825', 'Gibbs', 'Lewroy', 'Mr', '127
Moltres Way Jotho VIC', '0285624733', '3042',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Mildura')));
```

```
INSERT INTO Person
VALUES (PersonT('10015826', 'Beckett', 'Kate', 'Ms', '42
Donestry St Jibisy VIC', '0263957394', '3087',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Mildura')));
INSERT INTO Person
VALUES (PersonT('10000255', 'Morgan', 'Henry', 'Mr', '2 London
Ave Karrigan VIC', '0395182649', '3062',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='City')));
INSERT INTO Person
VALUES (PersonT('10000258', 'Flow', 'Max', 'Mr', '26 Hollow
Tips St Precidense VIC', '0492849184', '3012',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Mildura')));
INSERT INTO Person
VALUES (PersonT('10006935', 'Gunn', 'Montgomery', 'Mr', '65
Arrow Ave Catery VIC', '0492847294', '3085',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='Mildura')));
INSERT INTO Person
VALUES (PersonT('10012568', 'Shields', 'Duncan', 'Mr', '13
Flame St Seaside VIC', '0195837592', '3037',
          (SELECT REF(c) FROM Campus c WHERE
c.campus_location='City')));
-- additional Staff over

INSERT INTO Staff (SELECT pers.*, 'BG212', 'Lecturer' FROM
Person pers WHERE pers.person_id = 10008895);
INSERT INTO Staff (SELECT pers.*, 'BG210', 'Admin' FROM Person
pers WHERE pers.person_id = 10002935);
-- additional Staff needed for Subtypes
INSERT INTO Staff (SELECT pers.*, 'BG221', 'Admin' FROM Person
pers WHERE pers.person_id = 10008957);
INSERT INTO Staff (SELECT pers.*, 'BG231', 'Technician' FROM
Person pers WHERE pers.person_id = 10005825);
INSERT INTO Staff (SELECT pers.*, 'BG232', 'Technician' FROM
Person pers WHERE pers.person_id = 10015826);
INSERT INTO Staff (SELECT pers.*, 'BG225', 'Lecturer' FROM
Person pers WHERE pers.person_id = 10000255);
INSERT INTO Staff (SELECT pers.*, 'BG226', 'Lecturer' FROM
Person pers WHERE pers.person_id = 10000258);
INSERT INTO Staff (SELECT pers.*, 'BG225', 'Lecturer' FROM
Person pers WHERE pers.person_id = 10006935);
INSERT INTO Staff (SELECT pers.*, 'BG265', 'Tutor' FROM Person
pers WHERE pers.person_id = 01234234);
INSERT INTO Staff (SELECT pers.*, 'BG265', 'Tutor' FROM Person
pers WHERE pers.person_id = 01958652);
-- additional Subtypes over
```

```
INSERT INTO Student (SELECT pers.*, 2000 FROM Person pers
WHERE pers.person_id = 01234234);
INSERT INTO Student (SELECT pers.*, 2000 FROM Person pers
WHERE pers.person_id = 01958652);
-- additional Student needed for Enrolls_in / Takes
INSERT INTO Student (SELECT pers.*, 1995 FROM Person pers
WHERE pers.person_id = 10012568);
-- additional Students over

INSERT INTO AdminTbl (SELECT staff.*, 'Office Manager',
ComputerskillsT(), OfficeskillsT() FROM Staff staff WHERE
staff.person_id = 10002935);
INSERT INTO AdminTbl (SELECT staff.*, 'Receptionist',
ComputerskillsT(), OfficeskillsT() FROM Staff staff WHERE
staff.person_id = 10008957);

INSERT INTO TABLE (SELECT admin_officeskills FROM AdminTbl
WHERE person_id = '10002935')
VALUES('Managerial');
INSERT INTO TABLE (SELECT admin_computerskills FROM AdminTbl
WHERE person_id = '10008957')
VALUES('MS Office');
INSERT INTO TABLE (SELECT admin_officeskills FROM AdminTbl
WHERE person_id = '10008957')
VALUES('Customer Service');
INSERT INTO TABLE (SELECT admin_officeskills FROM AdminTbl
WHERE person_id = '10008957')
VALUES('Phone');

INSERT INTO Technician (SELECT staff.*, 'Network Officer',
TechnicianskillsT() FROM Staff staff WHERE staff.person_id =
10005825);
INSERT INTO Technician (SELECT staff.*, 'Photocopy
Technician', TechnicianskillsT() FROM Staff staff WHERE
staff.person_id = 10015826);

INSERT INTO TABLE (SELECT tech_skills FROM Technician WHERE
person_id = '10005825')
VALUES('UNIX');
INSERT INTO TABLE (SELECT tech_skills FROM Technician WHERE
person_id = '10005825')
VALUES('NT');
INSERT INTO TABLE (SELECT tech_skills FROM Technician WHERE
person_id = '10015826')
VALUES('Electrician');

INSERT INTO Lecturer (SELECT staff.*, 'Software Engineering',
'Associate' FROM Staff staff WHERE staff.person_id =
10008895);
INSERT INTO Lecturer (SELECT staff.*, 'Business Information',
'Senior' FROM Staff staff WHERE staff.person_id = 10000255);
-- additional Lecturers needed for Sub-Lecturers
```

```
INSERT INTO Lecturer (SELECT staff.*, 'Business
Administration', 'Senior' FROM Staff staff WHERE
staff.person_id = 10000258);
INSERT INTO Lecturer (SELECT staff.*, 'Software Development',
'Associate' FROM Staff staff WHERE staff.person_id =
10006935);
-- additional Sub-Lecturers over

INSERT INTO SeniorLecturer (SELECT lec.*, 2, 5, 7 FROM
Lecturer lec WHERE lec.person_id = 10000255);
INSERT INTO SeniorLecturer (SELECT lec.*, NULL, 1, 5 FROM
Lecturer lec WHERE lec.person_id = 10000258);

INSERT INTO AssociateLecturer (SELECT lec.*, 2, 1999 FROM
Lecturer lec WHERE lec.person_id = 10008895);
INSERT INTO AssociateLecturer (SELECT lec.*, NULL, 2001 FROM
Lecturer lec WHERE lec.person_id = 10006935);

INSERT INTO Tutor (SELECT staff.*, 10, 20.00 FROM Staff staff
WHERE staff.person_id = 01234234);
INSERT INTO Tutor (SELECT staff.*, 30, 35.00 FROM Staff staff
WHERE staff.person_id = 01958652);

INSERT INTO Subject
VALUES (SubjectT('CSE21NET', 'Networking', 10, 'CSE11IS',
          (SELECT REF(p) FROM Person p WHERE
p.person_id='10008895')));
INSERT INTO Subject
VALUES (SubjectT('CSE42ADB', 'Advanced Database', 15,
'CSE21DB',
          (SELECT REF(p) FROM Person p WHERE
p.person_id='10006935')));

INSERT INTO Enrolls_in VALUES ((SELECT REF(s) FROM Student s
WHERE s.person_id='01234234'), (SELECT REF(d) FROM DegreeTbl d
WHERE d.deg_id='D101'));
INSERT INTO Enrolls_in VALUES ((SELECT REF(s) FROM Student s
WHERE s.person_id='10012568'), (SELECT REF(d) FROM DegreeTbl d
WHERE d.deg_id='D101'));

INSERT INTO Takes VALUES ((SELECT REF(s) FROM Student s WHERE
s.person_id='01234234'), (SELECT REF(s) FROM Subject s WHERE
s.subject_id='CSE42ADB'), 70);
INSERT INTO Takes VALUES ((SELECT REF(st) FROM Student st
WHERE st.person_id='10012568'), (SELECT REF(su) FROM Subject
su WHERE su.subject_id='CSE42ADB'), 80);
```

Bitte verzeihen Sie das fehlende Syntax-Highlighting, wir machen dies immer von Hand.

Screendumps der Tabellen

Campus-Table:

	CAMPUS_LOCATION	CAMPUS_ADDR	CAMPUS_PHONE	CAMPUS_FAX	CAMPUS_HEAD
1	Albury/Wodonga	Parkers Road Wodonga VIC 3690	61260583700	620260583777	John Hill
2	City	215 Franklin St. Melb VIC 3000	61392855100	610392855111	Michael A. O'Leary
3	Mildura	Benetook Ave. Mildura VIC 3502	61350223757	61350223646	Ron Broadhead
4	Bundoora	221b Baker St. London NW1	6195135755	6137196482	Sherlock Holmes

Professor-Table:

	PROF_ID	PROF_NAME	PROF_CONTACT	PROF_RESEARCH	PROF_YEAR
1	42	Nick Hoogenraad	hoogenraad@cu.edu	Advanced Software Engineering	2007
2	88	Robin Anders	robin.anders@cu.edu	Computer Architecture	2017
3	101	Claude Bernard	cbernard@cu.edu	Networking	2016
4	420	Bruce Stone	b.stone@cu.edu	Software Development	2015
5	555	Chris Handley	handley@cu.edu	Compiler	2009
6	782	Sheena Reilly	sheenareilly@cu.edu	Discrete Mathematics	2005
7	1001	Alison Perry	Alison.Perry@cu.edu	Physics	2012
8	1337	Jan Branson	branson@cu.edu	Software Architecture	2018

Faculty-Table:

	FAC_ID	FAC_NAME	FAC_DEAN	DEPT
1	1	Health Science	S. Duckett	DBST42.DEPARTMENTLISTT()
2	2	Humanity & Social Sc.	J. A. Salmond	DBST42.DEPARTMENTLISTT()
3	3	Law & Management	G. C. O'Brien	DBST42.DEPARTMENTLISTT()
4	4	Science, Tech. & Eng.	D. Finlay	DBST42.DEPARTMENTLISTT([DBST42.DEPARTMENTT],[DBST42.DEPARTMENTT])
5	5	Regional Department	L. Kilmartin	DBST42.DEPARTMENTLISTT()

SCHOOL	RC
DBST42.SCHOOLLISTT([DBST42.SCHOOLT],[DBST42.SCHOOLT])	DBST42.RESEARCHCENTRELISTT([DBST42.RESEARCHCENTRET],[DBST42.RESEARCHCENTRET])
DBST42.SCHOOLLISTT()	DBST42.RESEARCHCENTRELISTT()
DBST42.SCHOOLLISTT()	DBST42.RESEARCHCENTRELISTT()
DBST42.SCHOOLLISTT()	DBST42.RESEARCHCENTRELISTT()
DBST42.SCHOOLLISTT()	DBST42.RESEARCHCENTRELISTT()

Building-Table:

	BLD_ID	BLD_NAME	BLD_LOCATION	BLD_LEVEL	CAMPUS	FAC
1	BB1	Beth Gleeson	D5	4	[DBST42.CAMPUST]	[DBST42.FACULTYT]
2	BB2	Martin Building	F5	4	[DBST42.CAMPUST]	[DBST42.FACULTYT]
3	BB3	Thomas Cherry	D4	4	[DBST42.CAMPUST]	[DBST42.FACULTYT]
4	BB4	Physical Science 1	D5	3	[DBST42.CAMPUST]	[DBST42.FACULTYT]

Person-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE	CAMPUS
1	01234234	Grant	Felix	Mr	2 Boadle Rd Bundoora VIC	0398548753	3083	[DBST42.CAMPUST]
2	10008895	Xin	Harry	Mr	6 Kelley St Kew VIC	0398875542	3088	[DBST42.CAMPUST]
3	10002935	Jones	Felicity	Ms	14 Rennie St Thornbury VIC	0398722001	3071	[DBST42.CAMPUST]
4	01958652	Doe	John	Mr	64 Austin St Holow VIC	0321343123	1337	[DBST42.CAMPUST]
5	10008957	Jane	Patrick	Mr	23 Rainbow Rd Allumy VIC	0236263636	3033	[DBST42.CAMPUST]
6	10005825	Gibbs	Lewroy	Mr	127 Moltres Way Jotho VIC	0285624733	3042	[DBST42.CAMPUST]
7	10015826	Beckett	Kate	Ms	42 Donestry St Jibisy VIC	0263957394	3087	[DBST42.CAMPUST]
8	10000255	Morgan	Henry	Mr	2 London Ave Karrigan VIC	0395182649	3062	[DBST42.CAMPUST]
9	10000258	Flow	Max	Mr	26 Hollow Tips St Precidense VIC	0492849184	3012	[DBST42.CAMPUST]
10	10006935	Gunn	Montgomery	Mr	65 Arrow Ave Catery VIC	0492847294	3085	[DBST42.CAMPUST]

Office-Table:

	OFFICE_NO	BLD	OFFICE_PHONE
1	BG207	[DBST42.BUILDINGT]	94791118
2	BG208	[DBST42.BUILDINGT]	94792393

Classroom-Table:

	CLASS_NO	BLD	CLASS_CAPACITY
1	TCLT	[DBST42.BUILDINGT]	50
2	TC01	[DBST42.BUILDINGT]	30

Lab-Table:

	LAB_NO	BLD	LAB_CAPACITY	LAB_EQUIPMENT
1	BG113	[DBST42.BUILDINGT]	25	DBST42.LABEQUIPMENTT('25 PC','1 Printer')
2	BG114	[DBST42.BUILDINGT]	20	DBST42.LABEQUIPMENTT('21 PC')

Degree-Table:

	DEG_ID	DEG_NAME	DEG_LENGTH	DEG_PREREQ	FAC
1	D100	Bachelor of Comp. Sci	3	Year 12 or equivalent	[DBST42.FACULTYT]
2	D101	Master of Comp. Sci	2	Bachelor of Comp. Sci	[DBST42.FACULTYT]

Student-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE	CAMPUS	STUDENT_YEAR
1	01234234	Grant	Felix	Mr	2 Boadle Rd Bundoora VIC	0398548753	3083	[DBST42.CAMPUST]	2000
2	01958652	Doe	John	Mr	64 Austin St Holow VIC	0321343123	1337	[DBST42.CAMPUST]	2000
3	10012568	Shields	Duncan	Mr	13 Flame St Seaside VIC	0195837592	3037	[DBST42.CAMPUST]	1995

Staff-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE	CAMPUS	OFFICE_NO	STAFF_TYPE
1	10008895	Xin	Harry	Mr	6 Kelley St Kew VIC	0398875542	3088	[DBST42.CAMPUST]	BG212	Lecturer
2	10002935	Jones	Felicity	Ms	14 Rennie St Thornbury VIC	0398722001	3071	[DBST42.CAMPUST]	BG210	Admin
3	10008957	Jane	Patrick	Mr	23 Rainbow Rd Allumy VIC	0236263636	3033	[DBST42.CAMPUST]	BG221	Admin
4	10005825	Gibbs	Lewroy	Mr	127 Moltres Way Jotho VIC	0285624733	3042	[DBST42.CAMPUST]	BG231	Technician
5	10015826	Beckett	Kate	Ms	42 Donestry St Jibisy VIC	0263957394	3087	[DBST42.CAMPUST]	BG232	Technician
6	10000255	Morgan	Henry	Mr	2 London Ave Karrigan VIC	0395182649	3062	[DBST42.CAMPUST]	BG225	Lecturer
7	10000258	Flow	Max	Mr	26 Hollow Tips St Precidense VIC	0492849184	3012	[DBST42.CAMPUST]	BG226	Lecturer
8	10006935	Gunn	Montgomery	Mr	65 Arrow Ave Catery VIC	0492847294	3085	[DBST42.CAMPUST]	BG225	Lecturer
9	01234234	Grant	Felix	Mr	2 Boadle Rd Bundoora VIC	0398548753	3083	[DBST42.CAMPUST]	BG265	Tutor
10	01958652	Doe	John	Mr	64 Austin St Holow VIC	0321343123	1337	[DBST42.CAMPUST]	BG265	Tutor

Admin-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE	CAMPUS	OFFICE_NO
1	10002935	Jones	Felicity	Ms	14 Rennie St Thornbury VIC	0398722001	3071	[DBST42.CAMPUST]	BG210
2	10008957	Jane	Patrick	Mr	23 Rainbow Rd Allumy VIC	0236263636	3033	[DBST42.CAMPUST]	BG221

OFFICE_NO	STAFF_TYPE	ADMIN_TITLE	ADMIN_COMPUTERSKILLS	ADMIN_OFFICESKILLS
BG210	Admin	Office Manager	DBST42.COMPUTERSKILLST()	DBST42.OFFICESKILLST('Managerial')
BG221	Admin	Receptionist	DBST42.COMPUTERSKILLST('MS Office')	DBST42.OFFICESKILLST('Customer Service','Phone')

Technician-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE
1	10005825	Gibbs	Lewroy	Mr	127 Moltres Way Jotho VIC	0285624733	3042
2	10015826	Beckett	Kate	Ms	42 Donestry St Jibisy VIC	0263957394	3087

PERSON_POSTCODE	CAMPUS	OFFICE_NO	STAFF_TYPE	TECH_TITLE	TECH_SKILLS
3042	[DBST42.CAMPUST]	BG231	Technician	Network Officer	DBST42.TECHNICIANSKILLST('UNIX','NT')
3087	[DBST42.CAMPUST]	BG232	Technician	Photocopy Technician	DBST42.TECHNICIANSKILLST('Electrician')

Tutor-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE
1	01234234	Grant	Felix	Mr	2 Boadle Rd Bundoora VIC	0398548753
2	01958652	Doe	John	Mr	64 Austin St Holow VIC	0321343123

PERSON_PHONE	PERSON_POSTCODE	CAMPUS	OFFICE_NO	STAFF_TYPE	TUTOR_HOURS	TUTOR_RATE
0398548753	3083	[DBST42.CAMPUST]	BG265	Tutor	10	20
0321343123	1337	[DBST42.CAMPUST]	BG265	Tutor	30	35

Lecturer-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE
1	10008895	Xin	Harry	Mr	6 Kelley St Kew VIC	0398875542
2	10000255	Morgan	Henry	Mr	2 London Ave Karrigan VIC	0395182649
3	10000258	Flow	Max	Mr	26 Hollow Tips St Precidense VIC	0492849184
4	10006935	Gunn	Montgomery	Mr	65 Arrow Ave Catery VIC	0492847294

PERSON_PHONE	PERSON_POSTCODE	CAMPUS	OFFICE_NO	STAFF_TYPE	LECT_AREA	LECT_TYPE
0398875542	3088	[DBST42.CAMPUST]	BG212	Lecturer	Software Engineering	Associate
0395182649	3062	[DBST42.CAMPUST]	BG225	Lecturer	Business Information	Senior
0492849184	3012	[DBST42.CAMPUST]	BG226	Lecturer	Business Administration	Senior
0492847294	3085	[DBST42.CAMPUST]	BG225	Lecturer	Software Development	Associate

Senior-Lecturer-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE
1	10000255	Morgan	Henry	Mr	2 London Ave Karrigan VIC	0395182649	3062
2	10000258	Flow	Max	Mr	26 Hollow Tips St Precidense VIC	0492849184	3012

PERSON_POSTCODE	CAMPUS	OFFICE_NO	STAFF_TYPE	LECT_AREA	LECT_TYPE	SENLECT_PHD	SENLECT_MASTER	SENLECT_HONOURS
3062	[DBST42.CAMPUST]	BG225	Lecturer	Business Information	Senior	2	5	7
3012	[DBST42.CAMPUST]	BG226	Lecturer	Business Administration	Senior	(null)	1	5

Associate-Lecturer-Table:

	PERSON_ID	PERSON_SURNAME	PERSON_FORENAME	PERSON_TITLE	PERSON_ADDR	PERSON_PHONE	PERSON_POSTCODE
1	10008895	Xin	Harry	Mr	6 Kelley St Kew VIC	0398875542	3088
2	10006935	Gunn	Montgomery	Mr	65 Arrow Ave Catery VIC	0492847294	3085

PERSON_POSTCODE	CAMPUS	OFFICE_NO	STAFF_TYPE	LECT_AREA	LECT_TYPE	ASSLECT_HONOURS	ASSLECT_YEAR
3088	[DBST42.CAMPUST]	BG212	Lecturer	Software Engineering	Associate	2	1999
3085	[DBST42.CAMPUST]	BG225	Lecturer	Software Development	Associate	(null)	2001

Subject-Table:

	SUBJECT_ID	SUBJECT_NAME	SUBJECT_CREDIT	SUBJECT_PREREQ	PERSON
1	CSE21NET	Networking	10	CSE11IS	[DBST42.PERSONT]
2	CSE42ADB	Advanced Database	15	CSE21DB	[DBST42.PERSONT]

Enrolls-In-Table:

	STUDENT	DEG
1	[DBST42.STUDENTT]	[DBST42.DEGREET]
2	[DBST42.STUDENTT]	[DBST42.DEGREET]

Takes-Table:

	STUDENT	SUBJECT	MARK
1	[DBST42.STUDENTT]	[DBST42.SUBJECTT]	70
2	[DBST42.STUDENTT]	[DBST42.SUBJECTT]	80

Anmerkungen

Uns ist bei der Planung von Typen und Tabellen aufgefallen, dass einige Tabellenattribute unnötig waren, da man über Joins an dieses Attribut gelangen könnte. So hatte bspw. die Staff-Tabelle in der Fallstudien-Beschreibung noch ein Attribut für das Gebäude (Building), dieses haben wir nicht übernommen, da man mit dem Office_Nr-Attribut über einen Join zur Office-Tabelle zu dem Gebäude gelangen kann. So haben wir unnötige Redundanzen / Speicherplatz-Verschwendung vermieden.

Auf der anderen Seite haben wir in vielen Tabellen wie bspw. Building, Person, Office, Classroom, Lab uvm. mit Referenzen statt einfachen IDs gehabt, dies hat zwei Vorteile:

- Wir können ohne Join sofort auf die Daten zugreifen (Dereferenzierung)
- Wir können sicherstellen das in bspw. der Takes-Tabelle nur gültige Studenten sind

Ich hoffe, dass diese Änderungen in Ihrem Sinn sind und unsere Gedankengänge für Sie sinnvoll bzw. nachvollziehbar erscheinen.

Falls Sie sich die SQL-Statements genauer ansehen oder zum selbst Testen kopieren möchten, finden Sie diese und alle anderen Dateien / Informationen auf GitHub.com/AnyaW/DBS2.