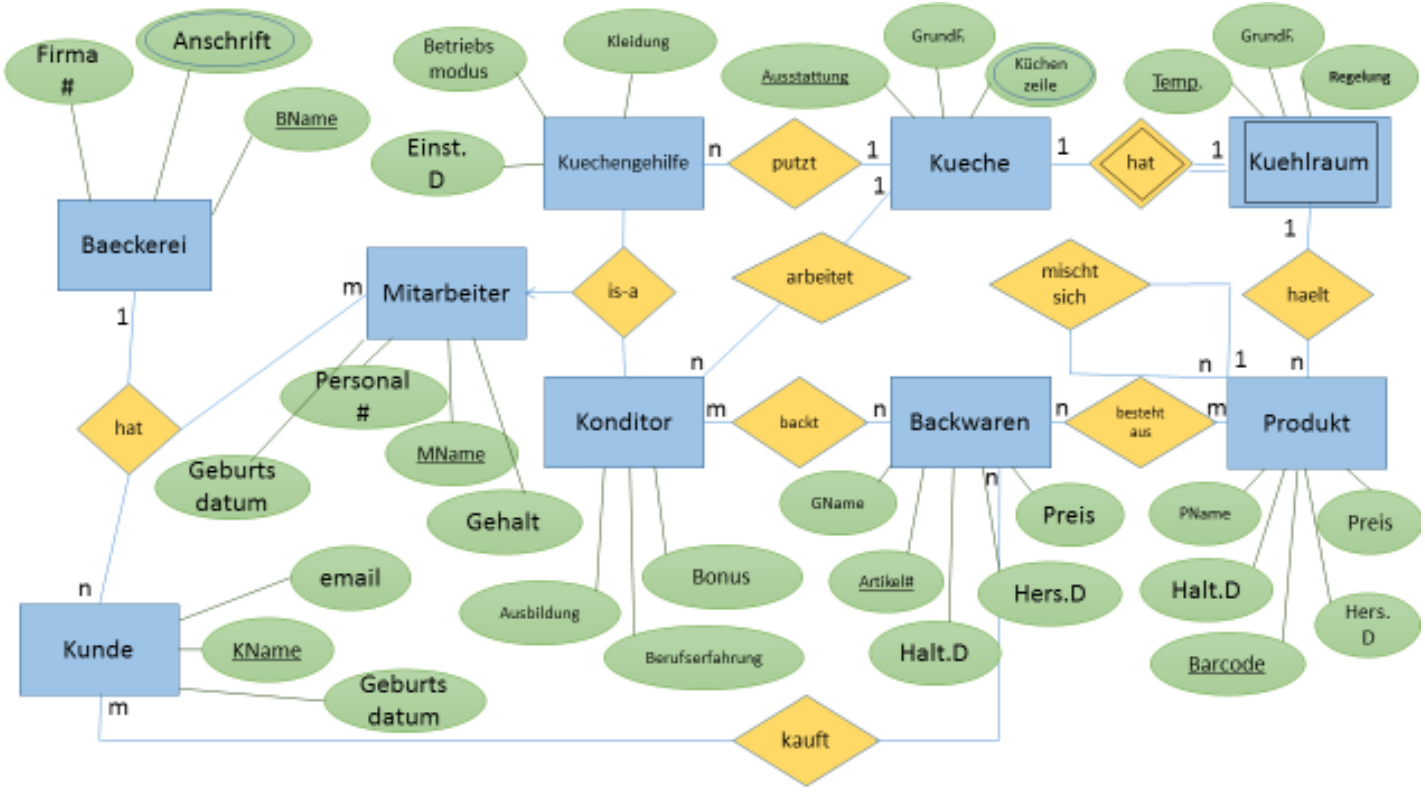
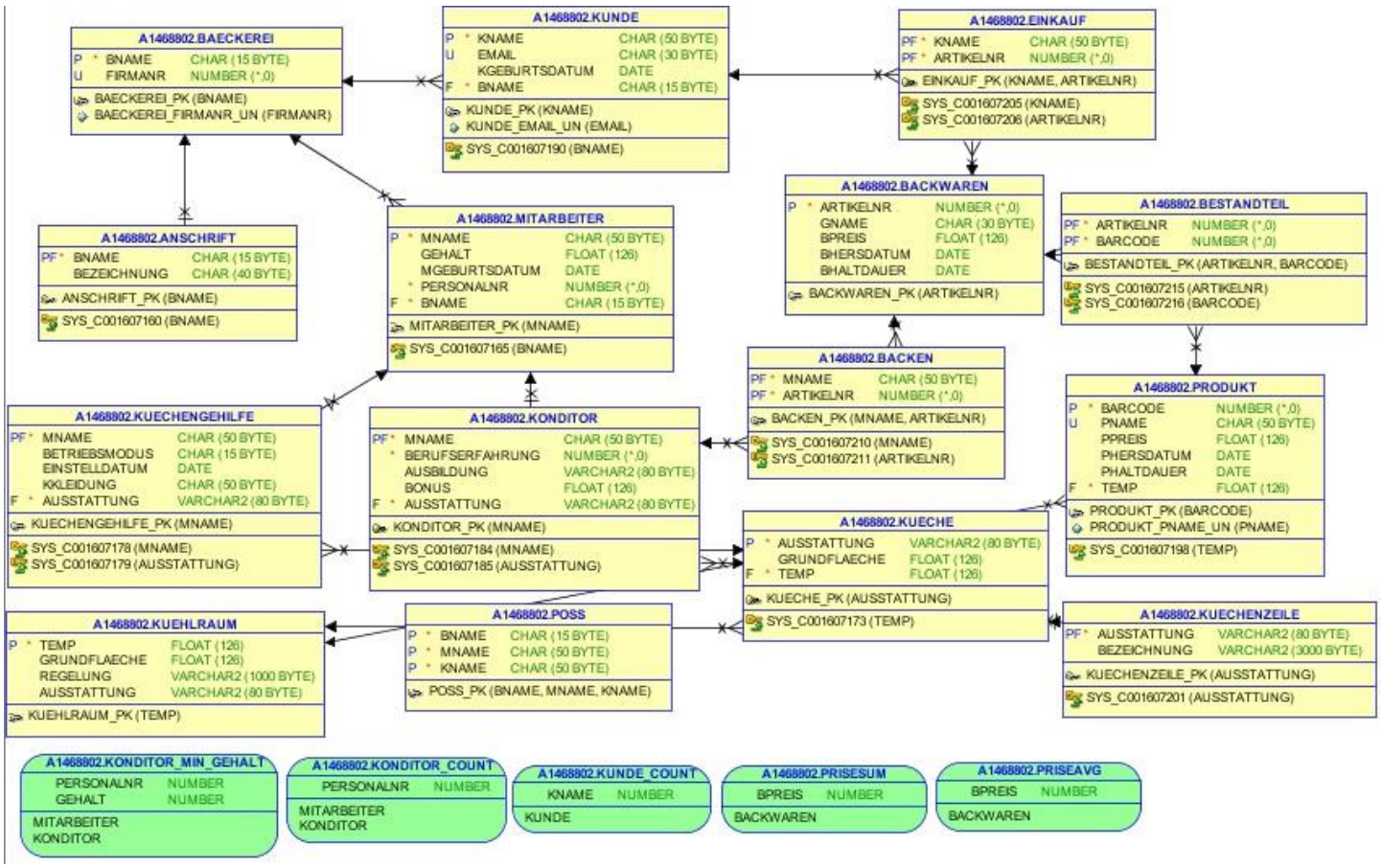
„Lecker“

Die Bäckerei, hat eine FirmarNr, Anschrift, einen Name(*„Lecker“*) und beschäftigt viele MitarbeiterInnen. Sie sind entweder Küchengehilfen oder Konditoren und haben eine Personalnummer, einen Namen, ein Geburtsdatum und ein Gehalt. Die Küchengehilfen haben einen Betriebsmodus, ein Einstellungsdatum und eine Küchenkleidung. Die Konditoren haben eine Ausbildung, eine Berufserfahrung und einen Bonus. Die Bäckerei hat viele Kunden, die eine oder mehrere Backwaren kaufen. Die Kunden haben einen Namen, Email und ein Geburtsdatum. Die Backwaren bestehen aus vielen verschiedenen Produkten und haben einen Namen, eine Haltbarkeitsdauer, einen Preis, ein Artikelnummer und ein Herstellungsdatum. Die Produkte haben einen Namen, eine Haltbarkeitsdauer, einen Barcode, einen Preis und ein Herstellungsdatum. Ein Produkt mischt sich mit anderen Produkten. Die Küchengehilfen bereiten die Produkte und putzen der Küche, wo die Konditoren die frischen Backwaren backen. Die Küchengehilfen arbeiten auch in der Küche. Die Küche hat eine Grundfläche, eine Ausstattung und Küchenzeile. In der Küche gibt ein Kühlraum. Der Kühlraum hält die Produkte und hat eine bestimmte Temperatur, eine Grundfläche und eine Regelung.

Diagramme




A1468802.BAECKEREI
P * BNAME CHAR (15 BYTE)
U FIRMANR NUMBER (*,0)
BAECKEREI_PK (BNAME)
BAECKEREI_FIRMANR_UN (FIRMANR)
A1468802.KUNDE
P * KNAME CHAR (50 BYTE)
U EMAIL CHAR (30 BYTE)
KGEBURTSDATUM DATE
F * BNAME CHAR (15 BYTE)
KUNDE_PK (KNAME)
KUNDE_EMAIL_UN (EMAIL)
SYS_C001607190 (BNAME)
A1468802.EINKAUF
PF * KNAME CHAR (50 BYTE)
PF * ARTIKELNR NUMBER (*,0)
EINKAUF_PK (KNAME, ARTIKELNR)
SYS_C001607205 (KNAME)
SYS_C001607206 (ARTIKELNR)
A1468802.ANSCHRIFT
PF * BNAME CHAR (15 BYTE)
BEZEICHNUNG CHAR (40 BYTE)
ANSCHRIFT_PK (BNAME)
SYS_C001607160 (BNAME)
A1468802.MITARBEITER
P * MNAME CHAR (50 BYTE)
GEHALT FLOAT (126)
MGEBURTSDATUM DATE
* PERSONALNR NUMBER (*,0)
F * BNAME CHAR (15 BYTE)
MITARBEITER_PK (MNAME)
SYS_C001607165 (BNAME)
A1468802.BACKWAREN
P * ARTIKELNR NUMBER (*,0)
GNAME CHAR (30 BYTE)
BPREIS FLOAT (126)
BHERSDATUM DATE
BHALTDAUER DATE
BACKWAREN_PK (ARTIKELNR)
A1468802.BESTANDTEIL
PF * ARTIKELNR NUMBER (*,0)
PF * BARCODE NUMBER (*,0)
BESTANDTEIL_PK (ARTIKELNR, BARCODE)
SYS_C001607215 (ARTIKELNR)
SYS_C001607216 (BARCODE)
A1468802.BACKEN
PF * MNAME CHAR (50 BYTE)
PF * ARTIKELNR NUMBER (*,0)
BACKEN_PK (MNAME, ARTIKELNR)
SYS_C001607210 (MNAME)
SYS_C001607211 (ARTIKELNR)
A1468802.PRODUKT
P * BARCODE NUMBER (*,0)
U PNAME CHAR (50 BYTE)
PPREIS FLOAT (126)
PHERSDATUM DATE
PHALTDAUER DATE
F * TEMP FLOAT (126)
PRODUKT_PK (BARCODE)
PRODUKT_PNAME_UN (PNAME)
SYS_C001607198 (TEMP)
A1468802.KUECHENGEHILFE
PF * MNAME CHAR (50 BYTE)
BETRIEBSMODUS CHAR (15 BYTE)
EINSTELLDATUM DATE
KKLEIDUNG CHAR (50 BYTE)
F * AUSSTATTUNG VARCHAR2 (80 BYTE)
KUECHENGEHILFE_PK (MNAME)
SYS_C001607178 (MNAME)
SYS_C001607179 (AUSSTATTUNG)
A1468802.KONDITOR
PF * MNAME CHAR (50 BYTE)
* BERUFSERFAHRUNG NUMBER (*,0)
AUSBILDUNG VARCHAR2 (80 BYTE)
BONUS FLOAT (126)
F * AUSSTATTUNG VARCHAR2 (80 BYTE)
KONDITOR_PK (MNAME)
SYS_C001607184 (MNAME)
SYS_C001607185 (AUSSTATTUNG)
A1468802.KUECHE
P * AUSSTATTUNG VARCHAR2 (80 BYTE)
GRUNDFLAECHE FLOAT (126)
F * TEMP FLOAT (126)
KUECHE_PK (AUSSTATTUNG)
SYS_C001607173 (TEMP)
A1468802.KUEHLRAUM
P * TEMP FLOAT (126)
GRUNDFLAECHE FLOAT (126)
REGELUNG VARCHAR2 (1000 BYTE)
AUSSTATTUNG VARCHAR2 (80 BYTE)
KUEHLRAUM_PK (TEMP)
A1468802.POSS
P * BNAME CHAR (15 BYTE)
P * MNAME CHAR (50 BYTE)
P * KNAME CHAR (50 BYTE)
POSS_PK (BNAME, MNAME, KNAME)
A1468802.KUECHENZEILE
PF * AUSSTATTUNG VARCHAR2 (80 BYTE)
BEZEICHNUNG VARCHAR2 (3000 BYTE)
KUECHENZEILE_PK (AUSSTATTUNG)
SYS_C001607201 (AUSSTATTUNG)
A1468802.KONDITOR_MIN_GEHALT
PERSONALNR NUMBER
GEHALT NUMBER
MITARBEITER
KONDITOR
A1468802.KONDITOR_COUNT
PERSONALNR NUMBER
MITARBEITER
KONDITOR
A1468802.KUNDE_COUNT
KNAME NUMBER
KUNDE
A1468802.PRISESUM
BPREIS NUMBER
BACKWAREN
A1468802.PRISEAVG
BPREIS NUMBER
BACKWAREN

Logische Entwurf

Baeckerei (BName, FirmaNr)
PK: BName

Anschrift (*BName*, Bezeichnung)
PK: BName
FK: Anschrift.BName◆ Baeckerei.BName

Mitarbeiter (PersonalNr, MName, Gehalt, Geburtsdatum, *Hat-BName*)
PK: PersonalNr
FK: Mitarbeiter.BName ◆ Baeckerei.BName

Kuechengehilfe (Betriebsmodus, EinstellDatum, KKleidung, *MName*)
PK: Betriebsmodus
FK: Kuechengehilfe. MName ◆ Mitarbeiter.MName

Konditor (Berufserfahrung, Ausbildung, Bonus, *MName, Arbeitet-Ausstattung*)
PK: Berufserfahrung
FK: Konditor. MName◆ Mitarbeiter. MName,
Konditor.Ausstattung◆Kueche.Ausstattung

Kunde (KName, SvNr, Geburtsdatum, *Hat-BName*)
PK: KName
FK: Kunde.BName ◆ Baeckerei.BName

Backwaren (ArtikelNr, GName, BPreis, BHersDatum, BHaltDauer)
PK: ArtikelNr

Produkt (Barcode, PName, PPreis, PHersDatum, PHaltDauer, *Haelt-Temp, Mischung-Barcode*)
PK: Barcode
FK: Produkt.Temp◆Kuehlraum.Temp,
Produkt.Barcode◆ Produkt.Barcode

Kueche (Ausstattung, Grundflaeche, *Temp*)
PK: Ausstattung
FK: Kueche.Temp ◆ Kuehlraum.Temp

Kuechenzeile (*Ausstattung*, Bezeichnung)
PK: Ausstattung
FK: Kuechenzeile.Ausstattung◆Kueche.Ausstattung

Kuehlraum (Temp, Grundflaeche, Regelung, *Ausstattung*)
PK: Temp, Ausstattung
FK: Kuehlraum.Ausstattung ◆ Kueche.Ausstattung

Einkauf (*KName, ArtikelNr*)
PK: KName, ArtikelNr
FK: Einkauf.KName ◆ Kunde.KName,
Einkauf.ArtikelNr◆ Backwaren.ArtikelNr

Backen (*MName, ArtikelNr*)
PK: MName, ArtikelNr
FK: Backen. MName ◆ Konditor. MName,
Backen.ArtikelNr ◆ Backwaren.ArtikelNr

Bestandteil (*ArtikelNr, Barcode*)
PK: ArtikelNr, Barcode
FK: Bestandteil.ArtikelNr ◆ Backwaren.ArtikelNr,
Bestandteil.Barcode ◆ Produkt.Barcode

Putzen (*MName, Ausstattung*)
PK: Betriebsmodus, Ausstattung
FK: Putzen.MName ◆ Kuechengehilfe.MName,
Putzen.Ausstattung ◆ Kueche.Ausstattung

Possesionem (*MName, KName,BName*)
PK: MName, KName, BName
FK: Possesionem.MName ◆ Mitarbieter.MName,
Possesionem.KName ◆ Kunde.KName,
Possesionem.BName ◆ Baeckerei.BName

Create Tables, Views und Procedures

```
CREATE TABLE baeckerei(
bname char(15) NOT NULL,
firmanr integer UNIQUE,
PRIMARY KEY (bname) );

CREATE TABLE anschrift(
bname char(15) NOT NULL,
bezeichnung char(40),
PRIMARY KEY (bname),
FOREIGN KEY(bname) REFERENCES baeckerei ON DELETE CASCADE );

CREATE TABLE mitarbeiter(
mname char(50) NOT NULL,
gehalt double precision,
mgeburtsdatum DATE,
personalnr integer NOT NULL,
bname char(15) NOT NULL,
PRIMARY KEY (mname),
FOREIGN KEY(bname) REFERENCES baeckerei ON DELETE CASCADE );

CREATE SEQUENCE mitar_persnr
START WITH 1000
INCREMENT BY 1;

CREATE OR REPLACE trigger mitar_persnr_trg
BEFORE INSERT ON mitarbeiter
FOR EACH ROW
BEGIN
    SELECT mitar_persnr.nextval INTO :new.personalnr FROM dual;
END;
/

CREATE TABLE kuehlraum(
temp double precision DEFAULT '4' NOT NULL,
grundflaeche double precision,
regelung varchar(1000),
ausstattung varchar(80),
PRIMARY KEY (temp),
CHECK (temp<8 AND temp>=0)
);

CREATE TABLE kueche(
ausstattung varchar(80) NOT NULL,
grundflaeche double precision,
temp double precision NOT NULL,
PRIMARY KEY (ausstattung),
CHECK (temp<8 AND temp>=0),
FOREIGN KEY (temp) REFERENCES kuehlraum ON DELETE CASCADE );
```

```
CREATE TABLE kuechengehilfe(
mname char(50) NOT NULL,
betriebsmodus char(15),
einstelldatum DATE,
kkleidung char(50),
ausstattung varchar(80) NOT NULL,
PRIMARY KEY (mname),
CHECK (betriebsmodus='Vormittag' OR betriebsmodus='Nachmittag'),
FOREIGN KEY (mname) REFERENCES mitarbeiter ON DELETE CASCADE,
FOREIGN KEY (ausstattung) REFERENCES kueche ON DELETE CASCADE );

CREATE TABLE konditor(
mname char(50) NOT NULL,
berufserfahrung integer NOT NULL,
ausbildung varchar(80),
bonus double precision DEFAULT '30',
ausstattung varchar(80) NOT NULL,
PRIMARY KEY (mname),
FOREIGN KEY (mname) REFERENCES mitarbeiter ON DELETE CASCADE,
FOREIGN KEY (ausstattung) REFERENCES kueche ON DELETE CASCADE );

CREATE TABLE kunde(
kname char(50) NOT NULL,
email char(30) UNIQUE,
kgeburtsdatum DATE,
bname char(15) NOT NULL,
PRIMARY KEY (kname),
FOREIGN KEY (bname) REFERENCES baeckerei ON DELETE CASCADE );

CREATE TABLE backwaren(
artikelnr integer NOT NULL,
gname char(30),
bpreis double precision,
bhersdatum DATE,
bhaltdauer DATE,
PRIMARY KEY (artikelnr) );

CREATE TABLE produkt(
barcode integer NOT NULL,
pname char(50) UNIQUE,
ppreis double precision,
phersdatum DATE,
phaltdauer DATE,
temp double precision NOT NULL,
PRIMARY KEY (barcode),
CHECK (temp<8 AND temp>=0),
FOREIGN KEY (temp) REFERENCES kuehlraum ON DELETE CASCADE );

CREATE TABLE kuechenzeile(
ausstattung varchar(80) NOT NULL,
bezeichnung varchar(3000),
```

```
PRIMARY KEY (ausstattung),
FOREIGN KEY (ausstattung) REFERENCES kueche ON DELETE CASCADE );

CREATE TABLE einkauf(
kname char(50) NOT NULL,
artikelnr integer NOT NULL,
PRIMARY KEY (kname, artikelnr),
FOREIGN KEY (kname) REFERENCES kunde ON DELETE CASCADE,
FOREIGN KEY (artikelnr) REFERENCES backwaren ON DELETE CASCADE );

CREATE TABLE backen(
mname char(50) NOT NULL,
artikelnr integer NOT NULL,
PRIMARY KEY (mname, artikelnr),
FOREIGN KEY (mname) REFERENCES konditor ON DELETE CASCADE,
FOREIGN KEY (artikelnr) REFERENCES backwaren ON DELETE CASCADE );

CREATE TABLE bestandteil(
artikelnr integer NOT NULL,
barcode integer NOT NULL,
PRIMARY KEY (artikelnr, barcode),
FOREIGN KEY (artikelnr) REFERENCES backwaren ON DELETE CASCADE,
FOREIGN KEY (barcode) REFERENCES produkt ON DELETE CASCADE );

CREATE TABLE poss(
bname char(15) NOT NULL,
mname char(50) NOT NULL,
kname char(50) NOT NULL,
PRIMARY KEY (bname,mname,kname),
FOREIGN KEY (bname) REFERENCES baeckerei ON DELETE CASCADE,
FOREIGN KEY (mname) REFERENCES mitarbeiter ON DELETE CASCADE,
FOREIGN KEY (kname) REFERENCES kunde ON DELETE CASCADE);


CREATE VIEW konditor_count (personalnr)
AS
SELECT personalnr
FROM mitarbeiter INNER JOIN konditor
ON mitarbeiter.mname=konditor.mname;

CREATE VIEW konditor_min_gehalt (personalnr, gehalt)
AS
SELECT
personalnr, Min(gehalt)
FROM mitarbeiter INNER JOIN konditor
ON mitarbeiter.mname=konditor.mname
GROUP BY personalnr, gehalt
HAVING MIN(gehalt)<1000;

CREATE VIEW priseAVG(bpreis)
AS
SELECT AVG(bpreis)
```

```
FROM backwaren;

CREATE VIEW priseSUM(bpreis)
AS
SELECT SUM(bpreis)
FROM backwaren;

CREATE VIEW kunde_count(kname)
AS
SELECT COUNT (DISTINCT kname)
FROM kunde;

create or replace PROCEDURE kon(name CHAR)
IS
BEGIN
DELETE FROM konditor WHERE mname=name;
END kon;
/

create or replace PROCEDURE kuch(name CHAR)
IS
BEGIN
DELETE FROM kuechengehilfe WHERE mname=name;
END kuch;
/

create or replace PROCEDURE persnr(nn IN CHAR, abt OUT INTEGER) IS
BEGIN
  Select personalnr INTO abt FROM mitarbeiter
  where mname=nn;
END persnr;
/
```

Drop Tables und Views

```
DROP TABLE baeckerei CASCADE CONSTRAINTS;
DROP TABLE anschrift CASCADE CONSTRAINTS;
DROP TABLE mitarbeiter CASCADE CONSTRAINTS;
DROP TABLE kuechengehilfe CASCADE CONSTRAINTS;
DROP TABLE konditor CASCADE CONSTRAINTS;
DROP TABLE kunde CASCADE CONSTRAINTS;
DROP TABLE backwaren CASCADE CONSTRAINTS;
DROP TABLE produkt CASCADE CONSTRAINTS;
DROP TABLE kueche CASCADE CONSTRAINTS;
DROP TABLE kuechenzeile CASCADE CONSTRAINTS;
DROP TABLE kuehlraum CASCADE CONSTRAINTS;
DROP TABLE einkauf CASCADE CONSTRAINTS;
DROP TABLE backen CASCADE CONSTRAINTS;
DROP TABLE bestandteil CASCADE CONSTRAINTS;
DROP SEQUENCE mitar_persnr;
DROP VIEW konditor_count CASCADE CONSTRAINTS;
DROP VIEW kunde_count CASCADE CONSTRAINTS;
DROP VIEW konditor_min_gehalt CASCADE CONSTRAINTS;
DROP VIEW priseAVG CASCADE CONSTRAINTS;
DROP VIEW priseSUM CASCADE CONSTRAINTS;
```

Kurze Beschreibung der Herangehensweise bei der Programmierung

1. SQL Tables Create + Drop:
15 Tables;
5 Views;

2. PHP Dateien erstellen:
Select, Insert : in jede php;
Insgesamt 11 .php Dateien

3. CSS erstellen;

4. Stored Procedure:
konditor.php -> “Delete” funktion;
kuechengehilfe.php -> “Delete” funktion;
mitarbeiter.php -> Input: Mitarbeiter Name,
Output: Mitarbeiter PersonalNr;

5. Java „TestDataGenerator1“ erstellen:
Kunde – 1000 Rows;
Produkt – 1000 Rows;
Mitarbeiter – 100 Rows;
Backwaren – 100 Rows;
Konditor – 10 Rows;
Kuechengehilfe – 10 Rows;
Backen – 10 Rows;
Einkauf – 10 Rows;
Bestandteil – 10 Rows;