



# **Prüfungsleistung ED im Fach Datenbankgestützte Informationssysteme (DBIS)**

## **Kassenbon-Data Warehouse, Umsätze und Analyse mit Aggregattabellen auf Wochenbasis**

Fachbereich: Management, Information, Technologie  
Prof. Dr. Alfred Wulff

Aufgabe:	13
Matrikelnummer:	6013868
Name:	Tokuc
Vorname:	Kübra
Email:	kuebra.tokuc@student.jade-hs.de
Studiengang:	Wirtschaftsinformatik
Semester:	WS2019/20
Abgabetermin:	16. Januar 2020

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Quelltextverzeichnis</b>	<b>IV</b>
<b>1. Aufgabenbeschreibung</b>	<b>1</b>
<b>2. Pflichtenheft</b>	<b>1</b>
2.1. Zielbestimmung . . . . .	1
2.2. Einsatz . . . . .	2
2.3. Umgebung . . . . .	2
2.3.1. Einfügen von Listings . . . . .	2
2.4. Daten . . . . .	3
2.5. Performance . . . . .	3
2.6. Qualitätsziele . . . . .	3
2.7. Ablieferung der Ergebnisse . . . . .	3
2.8. Selbstständige Erarbeitung der Ergebnisse . . . . .	3
2.9. Organisatorisches . . . . .	3
<b>3. Umsetzung</b>	<b>3</b>
3.1. Einrichtung der Datenbank . . . . .	3
3.2. Java-Programm . . . . .	3
3.3. Data Warehouse . . . . .	4
3.3.1. Erweiterung des Datenbestandes . . . . .	4
3.3.2. Aufbereitung des Datenbestandes . . . . .	4
3.3.3. Analyse des Starschemas . . . . .	4
3.3.4. ETL-Prozess . . . . .	5
3.3.5. Aggregate auf Wochenbasis . . . . .	6
3.4. Prozedur für das Reporting . . . . .	7
3.5. Reporting mit QlikSense . . . . .	7
<b>Literatur</b>	<b>9</b>
<b>A. Quelltext für das Java Programm</b>	<b>10</b>
A.1. ImportWindow . . . . .	10
A.2. DBConnection . . . . .	17
A.3. FileLine . . . . .	20
A.4. FileLineParser . . . . .	22
A.5. ImportRoutine . . . . .	27
A.6. DBImport . . . . .	32
A.7. DateFormat . . . . .	35

<b>B. Quelltext für SQL-Skripte</b>	<b>38</b>
B.1. 3-Insert-Student . . . . .	38
B.2. 4-ETL-DW . . . . .	50
B.3. 5-Fakten-Woche . . . . .	54
B.4. 6-DW-Report . . . . .	56
B.5. 7-Hilfstabelle . . . . .	58

## Abbildungsverzeichnis

1.	ER-Modell zur Auswertung von Kassenbons . . . . .	4
2.	Importfenster des Java-Programms . . . . .	5
3.	Konzeptionelles Datenmodell des Data-Warehouses . . . . .	6
4.	Physikalisches Datenmodell des Data-Warehouses . . . . .	7
5.	Datenmodell des DW mit Assoziierung . . . . .	8
6.	Aggregate auf Wochenbasis . . . . .	9
7.	Reporting-Tabelle . . . . .	61

## Listings

1.	Quelltext für ein Listing . . . . .	2
----	-------------------------------------	---

## 1. Aufgabenbeschreibung

In diesem Abschnitt (siehe auch Gubbi et al., 2013) der Arbeit wird das Ziel formuliert, in einen größeren Zusammenhang eingeordnet und gegen andere Themen abgegrenzt. Die wichtigsten Begriffe des Themas müssen in der Einleitung präzise definiert werden; eine sorgfältige Formulierung ist hier besonders wichtig. Weiterhin können Hinweise zur verwendeten Untersuchungsmethodik gegeben werden. Durch die Darstellung des Gangs der Untersuchung kann auch die Zweckmäßigkeit der gewählten Gliederung hervorgehoben werden. Nach Möglichkeit sollte dieses Kapitel nicht ‚Einleitung‘ heißen, sondern einen sinnvollen Titel mit Bezug zur Arbeit tragen.

## 2. Pflichtenheft

### 2.1. Zielbestimmung

Das Mischen von alter und neuer Rechtschreibung ist unzulässig.

Für die Erstellung der eigenen Arbeit kann es sinnvoll sein, dieses Dokument zu übernehmen und kontinuierlich die beispielhaften Bereiche gegen die eigenen neuen Passagen zu ersetzen; so bleibt der Aufbau erhalten und man verliert nicht versehentlich Formatierungen o. ä. Bei der Erstellung der Gliederung der eigenen wissenschaftlichen Arbeit sollten die beiden Kriterien Vollständigkeit und Überschneidungsfreiheit beachtet werden! Auf jeder eröffneten Gliederungsebene müssen jeweils mindestens zwei Gliederungspunkte existieren, also nicht:

---

2 Ist-Zustand

2.1 Ist-Zustand im Unternehmen XYZ

3 Soll-Konzept

---

Abkürzungen im Plural (Formatvorlagen) erhalten kein nachgestelltes „s“. Abkürzungen wie „PCs“ oder „CD-ROMs“ sind unzulässig.

Sollen einzelne Wörter im Text hervorgehoben werden, so ist eine kursive Hervorhebung dem Druck in fatter Schrift vorzuziehen.

## 2.2. Einsatz

Überschriften werden in L<sup>A</sup>T<sub>E</sub>X mit den Befehlen `\section{}`, `\subsection{}` und `\paragraph{}` erzeugt.

Jeder Überschrift sollte auf der tiefsten Gliederungsebene mindestens eine Seite Text folgen, davon mindestens zwei Zeilen auf derselben Seite. Es sollten nicht mehr als vier Gliederungsebenen verwendet werden.

Überschriften sollten in eine Zeile passen, damit Silbentrennungen vermieden werden können. Sollten Silbentrennungen in Ausnahmefällen erforderlich sein, ist sinn gemäß zu trennen, also z.B. nicht Umweltin-formatik, sondern Umwelt-informatik.

## 2.3. Umgebung

Bei der Erstellung von Abbildungen ist darauf zu achten, dass die erzeugten Grafiken selbstähnlich sein müssen, d. h. Größe, Schriftart, Schattierung, Linienart und -stärke, sowie die Art der Pfeilspitzen müssen in allen Grafiken gleich gewählt werden. Die serifenlose Schriftart Arial sollte in jedem Fall benutzt werden. Dabei sollte jedoch beachtet werden, dass auf Schatten, 3D-Effekte und Füllbereich zunächst zu verzichten ist. Sie dienen als Hervorhebung in einigen wenigen Grafiken; der Großteil der verwendeten Grafiken enthält diese Hervorhebungen nicht.

### 2.3.1. Einfügen von Listings

Abbildungen werden in L<sup>A</sup>T<sub>E</sub>X über die Umgebung `\begin{lstlisting}` eingefügt. Es ist zu beachten, dass in Listing 1 bewusst jede Zeile als Kommentar (Zeilenbeginn mit `%`) abgebildet ist, da L<sup>A</sup>T<sub>E</sub>X keine Listingdefinition als eigenes Listing zulässt.

Listing 1: Quelltext für ein Listing

```
1 %\begin{lstlisting}
2 % public static void main(args[]) {
3 % System.out.println("Hello World!");
4 % }
5 %\end{lstlisting}
```

**2.4. Daten****2.5. Performance****2.6. Qualitätsziele****2.7. Ablieferung der Ergebnisse****2.8. Selbstständige Erarbeitung der Ergebnisse****2.9. Organisatorisches****3. Umsetzung**

Zum Schluss der Arbeit kann in dem letzten Teil eine thesenartige Zusammenfassung der Untersuchungsergebnisse gegeben werden. Andere Möglichkeiten sind hier auch der Ausblick auf weitere – noch ungelöste – Fragestellungen im Zusammenhang mit dem Thema.

**3.1. Einrichtung der Datenbank****3.2. Java-Programm**

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

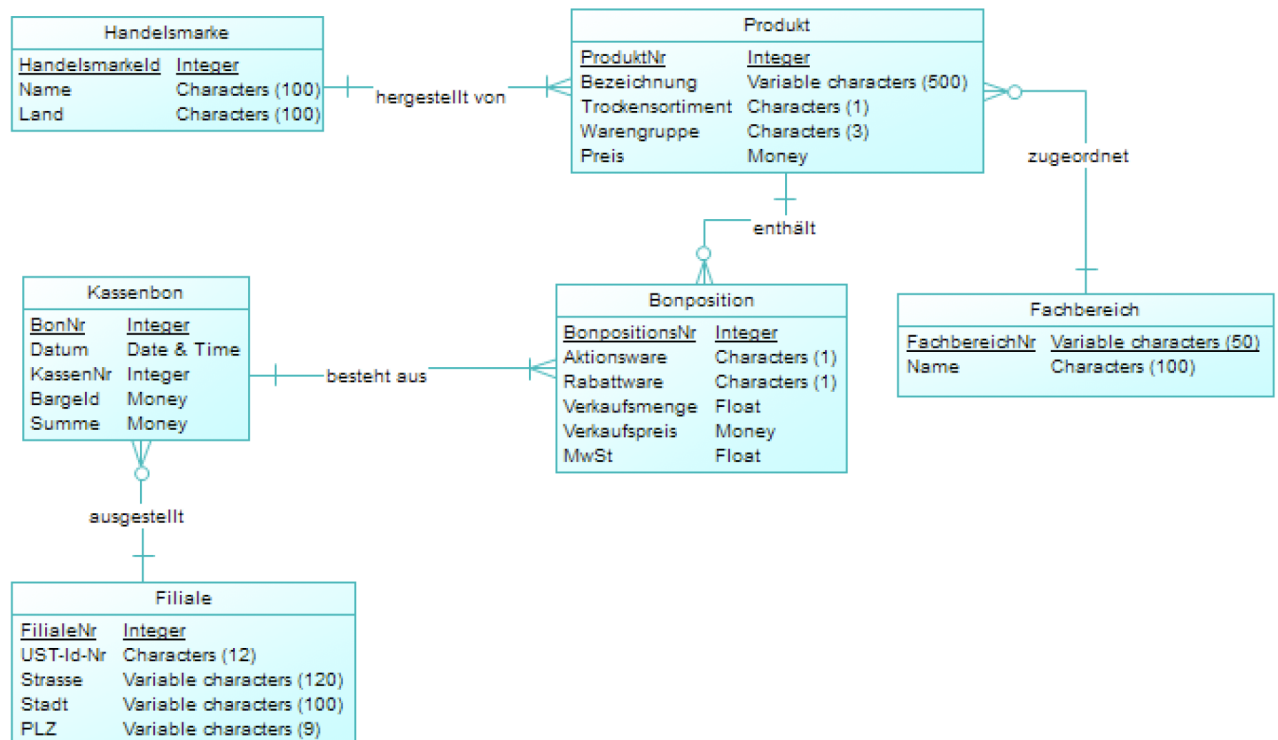


Abbildung 1: ER-Modell zur Auswertung von Kassenbons (Quelle: DBIS-Aufgaben)

### 3.3. Data Warehouse

#### 3.3.1. Erweiterung des Datenbestandes

#### 3.3.2. Aufbereitung des Datenbestandes

#### 3.3.3. Analyse des Starschemas

Nachdem der Datenbestand erweitert und korrekt aufbereitet worden ist, wird ein Star-Schema für ein Data-Warehouse zur Umsatzanalyse und Auswertung von Verkaufsdaten nach einem vorliegenden Schema (s. Abbildung 3) erstellt. Das Star-Schema besteht aus Dimensionen- und Faktentabellen, welche die Struktur und den Inhalt einer multidimensionalen Datenmenge definieren (Wulff, 2019). Die Werte in der Faktentabelle werden durch die Primary Keys der Dimensionstabelle mit ihr in Abhängigkeit gebracht (s. Abbildung 4). Allerdings ergibt die Analyse des Datenmodells Inkonsistenzen zwischen den verschiedenen Tabellen:



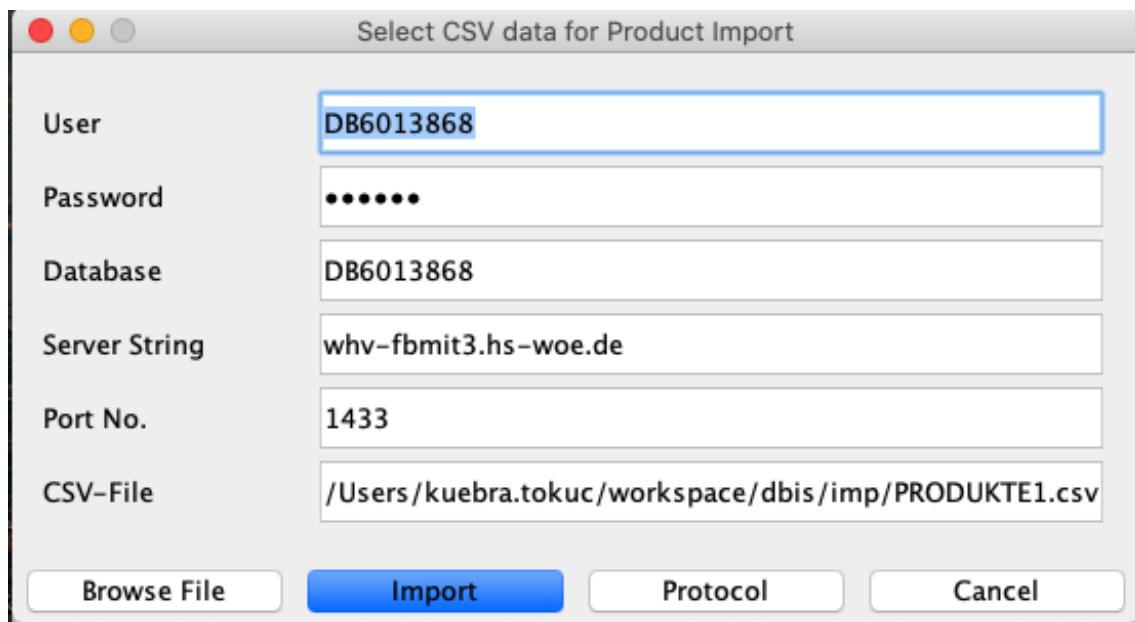


Abbildung 2: Importfenster des Java-Programms

1. Kein Fachbereich im Sternschema, stattdessen Abteilung
2. Kassenbon\_Fakten AbteilungsNr FK int und DIM\_Abteilung AbteilungsNr PK int
3. FilialeNr int der operativen DB und FilialeNr char(3) der DIM\_Filiale
4. Warengruppe char(3) der operativen DB (ODB) und Warengruppe char(1) der DIM\_Produkt
5. Bezeichnung des Produkts in ODB varchar(500) und in DIM\_Produkt varchar(100)

#### 3.3.4. ETL-Prozess

Mit dem ETL-Prozess (Extract, Transform, Load) werden die Daten aus der operativen Datenbank für die Auswertung und Analyse für das Laden in das Data-Warehouse aufbereitet.

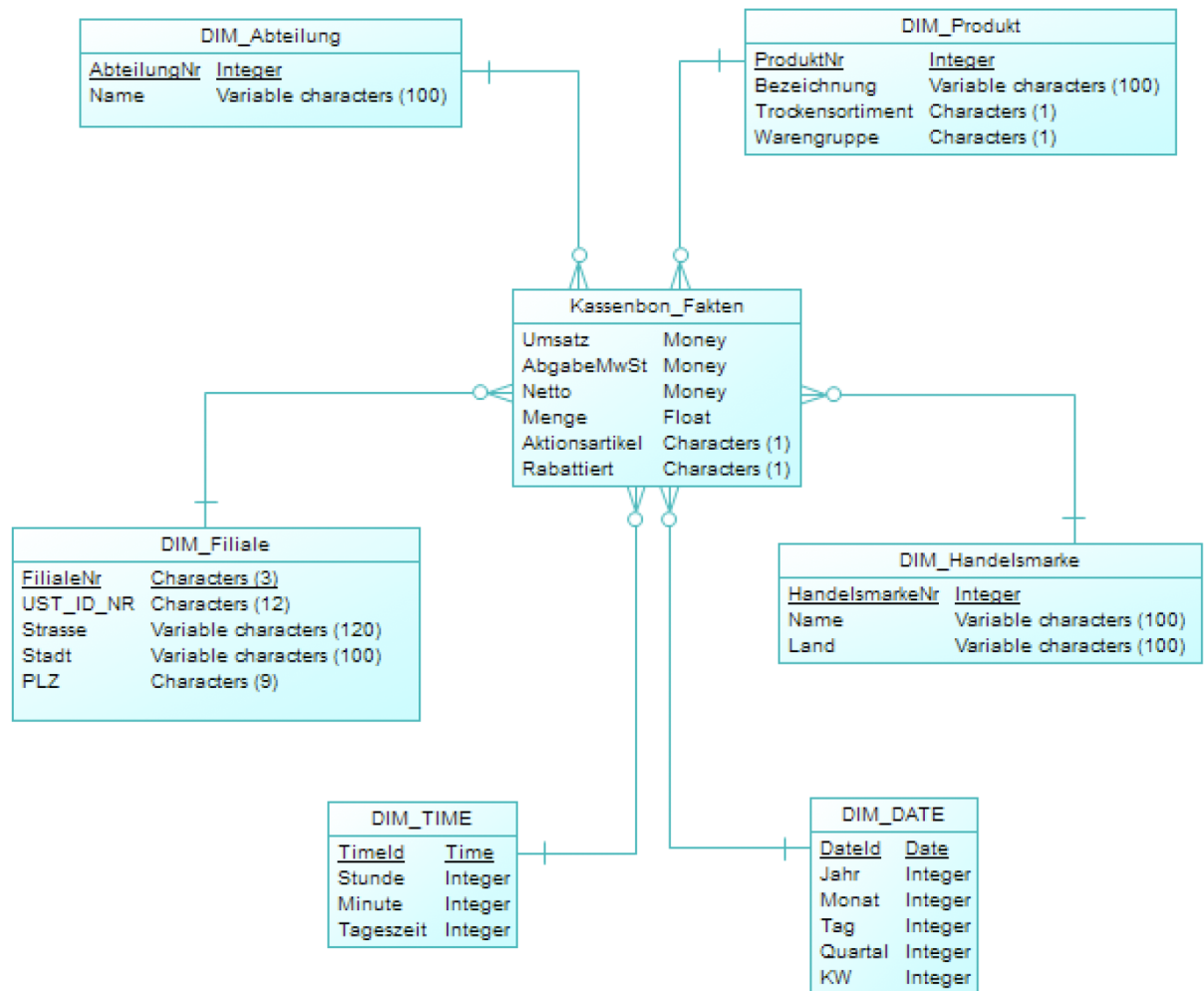


Abbildung 3: Konzeptionelles Datenmodell des Data-Warehouses (Quelle: DBIS-Aufgaben)

### 3.3.5. Aggregate auf Wochenbasis

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext

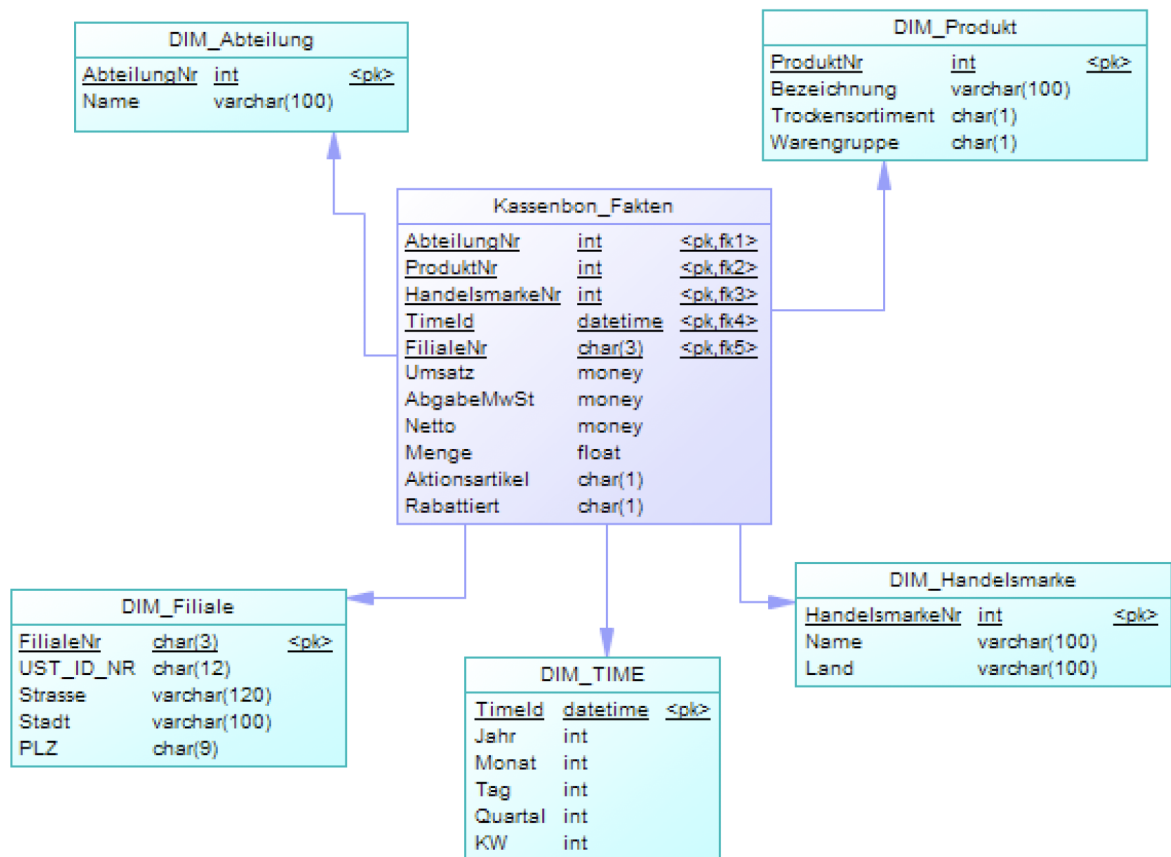


Abbildung 4: Physikalisches Datenmodell des Data-Warehouses (Quelle: DBIS-Aufgaben)

sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

### 3.4. Prozedur für das Reporting

### 3.5. Reporting mit QlikSense

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirk-

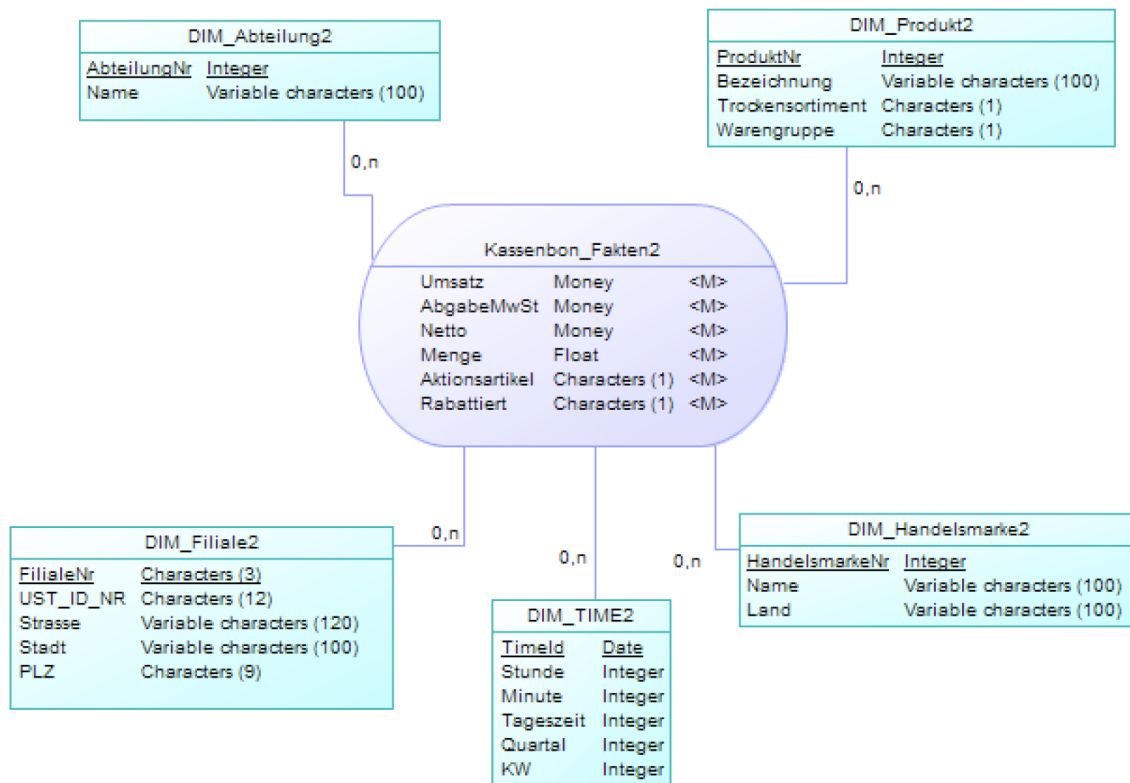


Abbildung 5: Datenmodell des DW mit Assoziierung (Quelle: DBIS-Aufgaben)

lich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

	ABTEILUNG NR	PRODUKT NR	HANDELSMARKEN NR	JAHR	WOCHE	TIMEID	FILIALE NR	Umsatz	ABGABEMWST	NETTO	MENGE	AKTIONSARTIKEL	RABATTIERT
1	1000	1	2	2019	1	08:45:00	1	6395	447.65	5947.35	500	N	N
2	1000	1	2	2019	1	09:00:00	1	6395	447.65	5947.35	500	N	N
3	1000	1	2	2019	1	09:12:00	1	6395	447.65	5947.35	500	N	N
4	1000	1	2	2019	1	08:05:00	1	1279	89.53	1189.47	100	N	N
5	1000	6	2	2019	1	08:21:00	1	124.9	8.743	116.157	10	N	Y
6	1014	10	22	2019	1	08:35:00	1	54	3.78	50.22	100	Y	Y
7	1014	11	22	2019	1	08:35:00	1	54	3.78	50.22	100	Y	Y
8	1014	12	22	2019	1	08:35:00	1	54	3.78	50.22	100	Y	Y
9	1014	13	22	2019	1	08:15:00	1	299	20.93	278.07	100	Y	N
10	1014	13	22	2019	1	08:35:00	1	104.5	7.315	97.185	50	Y	Y
11	1014	14	22	2019	1	08:15:00	1	299	20.93	278.07	100	Y	N
12	1014	15	22	2019	1	08:15:00	1	299	20.93	278.07	100	Y	N
13	1014	19	37	2019	1	08:45:00	1	665	46.55	618.45	1000	Y	Y
14	1014	19	37	2019	1	09:00:00	1	665	46.55	618.45	1000	Y	Y
15	1014	20	37	2019	1	08:45:00	1	745	52.15	692.85	1000	Y	Y
16	1014	20	37	2019	1	09:12:00	1	745	52.15	692.85	1000	Y	Y
17	1014	21	37	2019	1	08:45:00	1	745	52.15	692.85	1000	Y	Y
18	1014	21	37	2019	1	09:12:00	1	745	52.15	692.85	1000	Y	Y
19	1014	22	37	2019	1	08:15:00	1	233	16.31	216.69	1000	N	Y
20	1014	22	37	2019	1	08:21:00	1	116.5	8.155	108.345	50	N	N
21	1000	3	2	2019	6	08:22:00	14	174.3	12.201	162.099	70	Y	Y
22	1000	1	2	2019	6	08:22:00	14	89.53	6.2671	83.2629	7	N	N
23	1000	2	2	2019	6	08:22:00	14	83.93	5.8751	78.0549	7	N	N
24	1000	1	2	2019	7	08:22:00	14	89.53	6.2671	83.2629	7	N	N
25	1000	1	2	2019	7	10:22:00	14	89.53	6.2671	83.2629	7	N	N
26	1000	2	2	2019	7	10:22:00	14	16786	1175.02	15610.98	1400	Y	Y
27	1000	4	2	2019	7	10:22:00	14	146.3	10.241	136.059	70	Y	Y

Abbildung 6: Aggregate auf Wochenbasis

## Literatur

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

Wulff, A. (2019). Datenbankbasierte informationssysteme. Abgerufen 16.12.2019, von <https://moodle.jade-hs.de/moodle/course/view.php?id=5151>.

## A. Quelltext für das Java Programm

### A.1. ImportWindow

```
1
2  public class ImportWindow implements ActionListener{
3
4  // Datenbankverbindung mit JDBC Data Source
5
6  private DBConnection dbcon;
7  private Connection con;
8  private Statement statement;
9  //LoginFenster
10 public JFrame frmLoginWindow;
11 //Logger
12 private static final Logger LOG = Logger.getGlobal();
13
14 //Labels for Connection Screen
15 private JLabel userL;
16 private JLabel passwordL;
17 private JLabel databaseL;
18 private JLabel serverL;
19 private JLabel portL;
20 private JLabel fileL;
21 //Inputs in Connection Screen
22 private JTextField userF;
23 private JPasswordField passwordF;
24 private JTextField databaseF;
25 private JTextField fileF;
26 private JTextField serverF;
27 private JTextField portF;
28 //Buttons for Screens
29 private JButton btnImport;
```

```
30 private JButton btnFile;
31 private JButton btnCancel;
32 private JButton btnProtocol;
33 //Protocol
34 private String protocol_content;
35
36 //Constructor for starting Application Window
37 public ImportWindow() {
38     init();
39 }
40
41 public void init() {
42     frmLoginWindow = new JFrame("Select CSV data for Product Import");
43     frmLoginWindow.setBounds(100,100,550,300);
44     frmLoginWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
45     frmLoginWindow.setLocationRelativeTo(null);
46     frmLoginWindow.setResizable(false);
47
48     userL = new JLabel("User");
49     passwordL = new JLabel("Password");
50     databaseL = new JLabel("Database");
51     serverL = new JLabel("Server String");
52     portL = new JLabel("Port No.");
53     fileL = new JLabel("CSV-File");
54
55     //Buttons
56     btnImport = new JButton("Import");
57     btnImport.addActionListener(this);
58     btnFile = new JButton("Browse File");
59     btnFile.addActionListener(this);
60     btnCancel = new JButton("Cancel");
61     btnCancel.addActionListener(this);
62     btnProtocol = new JButton("Protocol");
```

```
63  btnProtocol.addActionListener(this);
64
65  // Input Fields
66
67  userF = new JTextField();
68  userF.setText("DB6013868");
69
70  passwordF = new JPasswordField();
71  passwordF.setText("3utbve");
72
73  databaseF = new JTextField();
74  databaseF.setText("DB6013868");
75
76  serverF = new JTextField();
77  serverF.setText("whv-fbmit3.hs-woe.de");
78
79  portF= new JTextField();
80  portF.setText("1433");
81
82  fileF = new JTextField();
83  fileF.setText(System.getProperty("user.dir")
84    + System.getProperty("file.separator") + "imp"
85    + System.getProperty("file.separator") + "PRODUKTE1.csv");
86
87  //JPanel for Labels
88
89  JPanel labelPane = new JPanel(new GridLayout(6, 1));
90  labelPane.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));
91
92  labelPane.add(userL);
93  labelPane.add(passwordL);
94  labelPane.add(databaseL);
95  labelPane.add(serverL);
```



```
96  labelPane.add(portL);
97  labelPane.add(fileL);
98
99  // JPanel for Inputs
100
101  JPanel fieldPane = new JPanel(new GridLayout(6, 1));
102  fieldPane.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));
103  fieldPane.add(userF);
104  fieldPane.add(passwordF);
105  fieldPane.add(databaseF);
106  fieldPane.add(serverF);
107  fieldPane.add(portF);
108  fieldPane.add(fileF);
109
110  // JPanel for Buttons
111
112  JPanel btnPane = new JPanel(new GridLayout(1,4));
113  btnPane.add(btnFile);
114  btnPane.add(btnImport);
115  btnPane.add(btnProtocol);
116  btnPane.add(btnCancel);
117
118  //Container
119  Container container = frmLoginWindow.getContentPane();
120  container.setBackground(Color.lightGray);
121
122  //Button Default
123  frmLoginWindow.getRootPane().setDefaultButton(btnImport);
124
125  //place panels
126  container.add(labelPane, BorderLayout.CENTER);
127  container.add(fieldPane, BorderLayout.LINE_END);
128  container.add(btnPane, BorderLayout.SOUTH);
```

```
129
130 frmLoginWindow.setVisible(true);
131 }
132
133 @Override
134 public void actionPerformed(ActionEvent e) {
135
136     if(e.getSource() == btnFile){
137         final JFileChooser fc = new JFileChooser();
138         int returnVal = fc.showOpenDialog(fc);
139
140         if (returnVal == JFileChooser.APPROVE_OPTION) {
141             File file = fc.getSelectedFile();
142             //This is where a real application would open the file.
143             fileF.setText(file.getAbsolutePath());
144             LOG.info("File selected with success");
145         }
146         System.out.println("Button geklickt!");
147     }
148     else if(e.getSource() == btnCancel) {
149         System.exit(0);
150         System.out.println("Canceled");
151         LOG.info("Canceled");
152     }
153     else if(e.getSource() == btnProtocol) {
154
155         JTextArea outputArea = new JTextArea(20, 40);
156         // Header and append
157         String header = getHeader();
158         String foot = getFoot();
159         protocol_content = header + protocol_content + foot;
160         outputArea.setText(protocol_content);
161         outputArea.setEditable(false);
```

```
162     JScrollPane scrollPane = new JScrollPane(outputArea);
163
164     JOptionPane.showMessageDialog(new JFrame(), scrollPane, "Protokoll",
165         JOptionPane.INFORMATION_MESSAGE);
166 }
167 // DB Import CSV
168 else if(e.getSource() == btnImport) {
169
170     LOG.info("Import clicked..");
171
172     try {
173         LOG.info("Connecting...");
174         dbcon = new DBConnection(serverF.getText(),
175             portF.getText(), databaseF.getText(),
176             userF.getText(), new String(
177                 passwordF.getPassword()));
178         con = dbcon.getConnection();
179
180         // Testen der Verbindung
181
182         boolean isOk = dbcon.testConnection();
183
184         //Test if connection is established --> true!!
185         System.out.println("Connection established"+ isOk);
186         protocol_content+= "Connection established to" + "
187             "+serverF.getText();
188
189         if (!isOk) {
190             JOptionPane.showMessageDialog(new JFrame(),
191                 "Die Anmeldung konnte nicht durchgeführt werden!"
192                 + "\n\nBitte überprüfen Sie Ihre Angaben!",
193                 "Anmeldung fehlgeschlagen!", JOptionPane.ERROR_MESSAGE);
```

```
194     } else {
195
196         // Das Hauptfenster für die Systemverwaltung wird geöffnet.
197         ImportRoutine dbisImport = new ImportRoutine(con, fileF
198             .getText());
199         try {
200             // Daten importieren
201             protocol_content = dbisImport.startImport();
202
203             if (dbisImport.isImportOk())
204                 JOptionPane.showMessageDialog(new JFrame(),
205                     "Datenimport wurde erfolgreich durchgeführt!", "Datenimport",
206                     JOptionPane.INFORMATION_MESSAGE);
207             else
208                 JOptionPane.showMessageDialog(new JFrame(),
209                     "Datenimport wurde abgebrochen!",
210                     "Datenimport fehlgeschlagen!", JOptionPane.ERROR_MESSAGE);
211         } catch (IOException e1) {
212             JOptionPane
213                 .showMessageDialog(
214                     new JFrame(),
215                     "Die Datei konnte nicht gefunden oder nicht gelesen werden!"
216                     + "\n\nBitte überprüfen Sie Pfad, Name und Rechte der
217                         Importdatei!",
218                     "Import fehlgeschlagen!", JOptionPane.ERROR_MESSAGE);
219             e1.printStackTrace();
220         }
221         //*****!!*****
222         dbcon.freeConnection();
223
224     } catch (SQLException e1) {
225         LOG.log(Level.SEVERE, "Fehler im Datensatz.", e1);
```

```
226     JOptionPane.showMessageDialog(new JFrame(),
227         "Keine Verbindung zur Datenbank!\n" +
228         "Bitte Überprüfen Sie Ihre Angaben!",
229         "Datenimport", JOptionPane.ERROR_MESSAGE);
230     }
231 }
232
233 }
234 private String getHeader() {
235     return "*****\n" + "Start: "
236         + Date_Format.getDdMMyyyyHHMi(System.currentTimeMillis())
237         + " Uhr \n" + "Benutzer: " + userF.getText() + "\n" + "Database: "
238         + databaseF.getText() + "\n"
239         + "*****\n\n";
240 }
241 private String getFoot() {
242     return "\n\n*****\n" + "End: "
243         + Date_Format.getDdMMyyyyHHMi(System.currentTimeMillis())
244         + " Uhr \n" + "*****\n\n";
245 }
246
247
248 }
```

## A.2. DBConnection

```
1
2     public class DBConnection {
3
4         // Establish Connection via DataSource --> "mittlerweile bevorzugt"
5         // i-net MERLIA.jar is JDBC 3.0 / 4.0 driver for MS SQL Server
6         // Quelle 6. Dezember 2019:
           https://www.inetsoftware.de/products/jdbc-driver/ms-sql/features
```

```
7 // Class.forName("com.inet.tds.TdsDriver") - MS-SQL-Server
8 // This class is an implementation of a simple javax.sql.DataSource
  for the driver i-net ...
9 // ... OPTA-xs and MERLIA-xs. For application servers you should use
  the PDataSource or DTCDataSource.
10 // Quelle 6. Dezember TDSDDataSource:
    https://www.inetsoftware.de/documentation/jdbc-driver/ms-sql/apispec/index.html?c
11 // public TdsDataSource () Methods:
12 // getConnection(username, password) throws java.sql.SQLException
    returns a Connection to database
13 // setUrl(java.lang.String.jdbcUrl) and getUrl()
14 // setServerName(String serverName)
15 // getInstanceName()
16 // setDatabaseName(String databaseName) etc ....
17 // setPort(String port) --> cast from Textfield not necessary, if not
    set, default ist 1433
18 // setUser and getUser as well as getPassword set Password both String
19
20
21 private final String serverName;
22 private final String port;
23 private final String databaseName;
24 private final String username;
25 private final String password;
26 private Connection con;
27 private static final Logger log = Logger.getGlobal();
28
29 // Constructor for DB-Connection-Object
30 public DBConnection(String serverName, String port, String
    databaseName, String username, String password) {
31
32     this.serverName = serverName;
33     this.port = port;
```

```
34  this.databaseName = databaseName;
35  this.username = username;
36  this.password = password;
37  }
38
39  public synchronized Connection getConnection() throws SQLException{
40
41      // Connection with DataSource
42
43      try {
44          if (con == null || con.isClosed()) {
45              DataSource ds = new TdsDataSource();
46              ((TdsDataSource) ds).setServerName(serverName);
47              ((TdsDataSource) ds).setPortNumber(Integer.parseInt(port));
48              con = ds.getConnection(username, password);
49              con.setCatalog(databaseName);
50          }
51      } catch (SQLException e) {
52          log.log(Level.WARNING, "DB Connection failed.", e);
53          JOptionPane.showMessageDialog(new JFrame(),
54              " DB Connection failed!\n" +
55              "Check your connection input",
56              "CSV Import", JOptionPane.ERROR_MESSAGE);
57      }
58      return con;
59
60  }
61  public synchronized void freeConnection() {
62      try {
63          System.out.println("Active DB-Verbindung wird geschlossen... ");
64          con.close();
65          con = null;
66      } catch (Exception e) {
```

```
67
68     }
69 }
70
71 public synchronized boolean testConnection() {
72     try {
73         // Connection worked!
74         log.info("Testing Connection in DBConnectionDataSource");
75         if (getConnection() != null)
76             return true;
77     } catch (Exception e) {
78         e.printStackTrace();
79     }
80     return false;
81
82 }
83 public String getSchemaname() {
84     return "dbo";
85 }
86
87
88 }
```

### A.3. FileLine

```
1     public class FileLine {
2
3     private int produktNummer;
4     private String fachbereichNr;
5     private int handelsmarkeId;
6     private String bezeichnung;
7     private String trockensortiment;
8     private String warengruppe;
```



```
9  private double preis;
10
11  // Constructor for FileLine
12
13  public FileLine(
14      String fachbereichNr,
15      int handelsmarkeId,
16      String bezeichnung,
17      String trockensortiment,
18      String warengruppe,
19      double preis
20  ) {
21      this.fachbereichNr = fachbereichNr;
22      this.handelsmarkeId = handelsmarkeId;
23      this.bezeichnung = bezeichnung;
24      this.trockensortiment = trockensortiment;
25      this.warengruppe = warengruppe;
26      this.preis = preis;
27  }
28
29  public int getProduktNummer() {
30      return produktNummer;
31  }
32
33  public String getFachbereichNr() {
34      return fachbereichNr;
35  }
36
37  public int getHandelsmarkeId() {
38      return handelsmarkeId;
39  }
40
41  public String getBezeichnung() {
```

```
42     return bezeichnung;
43 }
44
45 public String getTrockensortiment() {
46     return trockensortiment;
47 }
48
49 public String getWarengruppe() {
50     return warengruppe;
51 }
52
53 public double getPreis() {
54     return preis;
55 }
56
57
58 }
```

#### A.4. FileLineParser

```
1
2     public class FileLineParser {
3
4         private static final int SPALTENANZAHL = 4;
5
6         // Erzeugt aus der übergebenen Zeile ein Object
7
8         protected static FileLine getExpFileLine(String line)
9             throws NoSuchElementException, SQLException {
10
11             String bezeichnung = getValue(line, 1).trim();
12             String trockensortiment = getValue(line, 2).trim();
13             String warengruppe = getValue(line, 3).trim();
```

```
14  String preis = getValue(line, 4).trim();
15  String  fb_id;
16
17  //Versuch: Handelsmarke und Fachbereich integrieren
18  //Hardcoded Idee für Fachbereich: Methode getFachbereich wie
    getWarengruppe
19  //--> Fachbereichsnummer ist der Foreign Key
20
21  // Gewuerze sind nicht als Fachbereich vorhanden
22
23  //Schwieriger: Handelsmarke -> Bezeichnung nochmal Tokenizen (1.
    Stelle oder mit Regex wegen Dr Oetker und Kellogs)
24  //Muss auch ein insert für Handelsmarke geben, wenn nicht vorhanden
25  // Neue Handelsmarken: Weber, Kellog's, Nestle, Basic, Schaer. Nick
26
27
28
29  if(bezeichnung.length() >500) {
30      bezeichnung.substring(0, 500);
31  }
32  if(trockensortiment.length() >1) {
33      trockensortiment.substring(0, 1);
34  }
35
36
37
38  warengruppe = FileLineParser.getWarengruppe(warengruppe);
39  fb_id = FileLineParser.getFachbereichsNummer(warengruppe);
40
41  if(warengruppe.length() >3) {
42      warengruppe.substring(0,3);
43  }
44  FileLine fileLineObj = null;
```

```
45  try {
46
47      fileLineObj = new FileLine(fb_id,
                                9,bezeichnung,trockensortiment,warengruppe,
                                Float.parseFloat(preis));
48  } catch (Exception e) {
49      System.out
50          .println("Parsefehler: Bitte überprüfen Sie das Format in dieser
                    Zeile:"
                    + line);
51      throw new NoSuchElementException();
52  }
53  return fileLineObj;
54  }
55
56
57  /**
58   * Liefert einen Wert, für den entsprechenden Index
59   */
60
61  private static String getValue(String line, int keyIndex)
62      throws NoSuchElementException {
63      String value = null;
64      boolean isDub = false;
65
66      StringTokenizer lineValues = new StringTokenizer(line, ";");
67      int tokens = lineValues.countTokens();
68
69      // Überprüfung der Spaltenanzahl
70      if (tokens < SPALTENANZAHL) {
71          // Falls mehr Tokens in Zeile als notwendige Spaltenzahl, Throw
              NoSuchElementException
72          System.out.println("Falsches Format von Testdaten!"
73              + "Zu wenige Parameter gefunden: ")

```

```
74     + tokens + " von " + SPALTENANZAHL + "Spalten.");
75     throw new NoSuchElementException("Falsches Format der Daten!\n");
76 } else if (tokens > SPALTENANZAHL) {
77     System.out.println("Falsches Format von Testdaten!"
78         + " Zu viele Parameter gefunden: " + tokens + " von "
79         + SPALTENANZAHL + " Spalten.");
80     throw new NoSuchElementException("Falsches Format der Daten!\n");
81 }
82
83 for (int i = 0; i < keyIndex; i++) {
84     value = lineValues.nextTok();
85 }
86
87 // Checkt, ob der Wert redundante Werte hat
88
89 isDub = isDub(value);
90 if(isDub == true) {
91     System.out.println("Falsches Format von Testdaten!"
92         + "Redundante Daten");
93     throw new NoSuchElementException("Falsches Format der Daten!\n");
94 }
95
96 return value;
97 }
98
99 // Übergabe der Warengruppennummer
100
101 public static String getWarengruppe(String s_warengruppe) {
102
103     if (s_warengruppe == null)
104         return null;
105
106     String warengruppe_id = null;
```

```
107
108     if (s_warengruppe.equalsIgnoreCase("Milch"))
109         warengruppe_id = "002";
110     else if (s_warengruppe.equalsIgnoreCase("Musli & Cerealien"))
111         warengruppe_id = "003";
112     else if (s_warengruppe.equalsIgnoreCase("Saucen"))
113         warengruppe_id = "004";
114     else if (s_warengruppe.equalsIgnoreCase("Gewurze"))
115         warengruppe_id = "005";
116     else
117         warengruppe_id = "006";
118
119     return warengruppe_id;
120 }
121
122 // Übergabe der Fachbereichsnummer
123 // wird auch Fachbereich fuer Gewurze erstellen
124
125 public static String getFachbereichsNummer(String warengruppe_id)
126     throws SQLException {
127
128     if (warengruppe_id==null) {
129         return null;
130     }
131     String fb_id= null;
132     if (warengruppe_id=="002") {
133         fb_id= "1014";
134     }else if(warengruppe_id == "003") {
135         fb_id= "1023";
136     }else if(warengruppe_id == "004") {
137         fb_id= "1015";
138     }else if(warengruppe_id == "005") {
139         ImportRoutine.createNewFB("Gewurze", "1039");
```

```
139     fb_id="1039";
140 }
141
142     return fb_id;
143 }
144
145 // Methode zum Prüfen von Redundanten Einträgen
146
147 public static boolean isDub(String bezeichnung) {
148
149     String[] words = bezeichnung.split("[\\s+]");
150
151     Map<String, Integer> occurrences = new HashMap<String, Integer>();
152
153     boolean isdub =false;
154     Integer oldCount=0;
155     for ( String word : words ) {
156         oldCount = occurrences.get(word);
157         if ( oldCount == null ) {
158             oldCount = 0;
159         }
160         occurrences.put(word, oldCount + 1);
161         if(oldCount>2) isdub = true;
162         System.out.println("oldcount : "+ oldCount+ "word"+ word + "isdub"
163             + isdub);
164     }
165     System.out.println("IS DUB? :"+ isdub);
166     return isdub;
167 }
168 }
```

## A.5. ImportRoutine

```
1
2 public class ImportRoutine {
3
4     //Schnittstelle zum Importieren von Daten
5
6     protected BufferedReader in;
7     protected final int DOPPELTER_PRIMARY_KEY_FEHLER = 2601;
8     protected final int DOPPELTER_UNIQUE_SCHLUSSEL_FEHLER = 2627;
9     protected String fileIn;
10    protected static Connection con;
11    private boolean istImportOk = true;
12
13    public ImportRoutine(Connection con, String fileIn) {
14        this.con = con;
15        this.fileIn = fileIn;
16    }
17
18    /**
19     * Liest Datei aus und speichert sie in die Datenbank.
20     */
21    public String startImport() throws IOException {
22        String protocol = "";
23        if (con == null) {
24            System.out
25                .println("Datenbankverbindung muss initialisiert
26                        werden!");
27        }
28        try {
29            try {
30                // Buffer für Zeilen
31                in = new BufferedReader(new FileReader(fileIn));
32            }
33            catch (Exception e) {e.printStackTrace();}
```



```
33
34     String line = null;
35     int lineCounter = 0;
36     int objCounter = 0;
37
38     protocol += "Datenimport wird gestartet...\n\n";
39
40     // Abbruch des Exports bei falschen Zeilen
41
42     try {
43         line = in.readLine();
44     }
45     catch (IOException e1) {
46         // TODO Auto-generated catch block
47         e1.printStackTrace();
48     }
49
50     while (line != null && istImportOk) {
51         lineCounter++;
52
53         // Kommentare ausschliessen
54         if (!line.trim().equals("") && !line.substring(0,
55             1).equals("#")) {
56             try {
57                 // Zeile ueberspringen
58                 objCounter++;
59                 FileLine fileLine = FileLineParser.getExpFileLine(line);
60                 DBImport.insert(fileLine, con);
61             } catch (SQLException e) {
62
63                 // Fehler ignorieren, wenn Zeile existiert
64
```

```
65         if (e.getErrorCode() != DOPPELTER_PRIMARY_KEY_FEHLER
66             && e.getErrorCode() !=
67                 DOPPELTER_UNIQUE_SCHLUSSEL_FEHLER) {
68             // Der Import soll weiter laufen, wenn der Datensatz
69             // existiert.
70             // istImportOk = false;
71             e.printStackTrace();
72             protocol += "SQL-Exception in der Zeile: "
73                 + lineCounter + "\n";
74             protocol += line + "\n\n";
75             JOptionPane
76                 .showMessageDialog(
77                 new JFrame(),
78                 "Daten aus der Zeile "
79                 + lineCounter
80                 + " verursachten beim Ausführen der
81                 SQL-Anweisung Fehler."
82                 + "\nBitte überprüfen Sie die SQL-Anweisung
83                 oder zu importierenden Daten!",
84                 "Datenimport",
85                 JOptionPane.ERROR_MESSAGE);
86         } else {
87             protocol += "Datensatz in der Zeile " + lineCounter +
88                 " ist bereits vorhanden:\n";
89             protocol += line + "\n\n";
90         }
91     } catch (ClassNotFoundException e) {
92         // istImportOk = false;
93         JOptionPane
94             .showMessageDialog(
95             new JFrame(),
96             "Keine Verbindung zur Datenbank!"
97             + "\nBitte überprüfen Sie Ihre Angaben!",
```

```
94         "Datenimport",
95         JOptionPane.ERROR_MESSAGE);
96     e.printStackTrace();
97     protocol += "Keine Verbindung zur Datenbank!\n\n";
98     break;
99 } catch (NoSuchElementException e) {
100     protocol += "Parse-Exception in der Zeile: "
101         + lineCounter + "\n";
102     protocol += line + "\n\n";
103     JOptionPane
104         .showMessageDialog(
105         new JFrame(),
106         "Daten aus der Zeile "
107         + lineCounter
108         + " konnten nicht importiert werden."
109         + "\nBitte überprüfen Sie die zu
            importierenden Daten!",
110         "Datenimport",
111         JOptionPane.ERROR_MESSAGE);
112     e.printStackTrace();
113     // break;
114 }
115 }
116 try {
117     line = in.readLine();
118 } catch (IOException e) {
119     // TODO Auto-generated catch block
120     e.printStackTrace();
121 }
122 }
123 protocol += "Anzahl der verarbeiteten Datensätze: " +
124     objCounter + "\n\n";
125 protocol += "Datenimport ist abgeschlossen.\n\n";
```

```
125     } finally {
126         if (in != null)
127             try {
128                 in.close();
129             } catch (IOException e) {
130                 // TODO Auto-generated catch block
131                 e.printStackTrace();
132             }
133     }
134     return protocol;
135 }
136
137 // Neuer Fachbereich für Gewürze erstellen, bevor der Import
138 // vollzogen wird
139
140 public static void createNewFB(String name, String fb_nr) throws
141     SQLException {
142
143     DBImport.insertFB(name, fb_nr, con);
144
145 }
146
147 /**
148  * Liefert true, wenn der Import ohne Fehler durchgeführt wurde.
149  */
150 public boolean isImportOk() {
151     return istImportOk;
152 }
```

## A.6. DBImport

```
1
2  // Klasse zum Importieren der CSV-Zeilen in die Datenbank
3
4  public class DBImport{
5
6  /**
7   * Speichert Daten der übergebenen Zeile in der Datenbank.
8   *
9   * @throws ClassNotFoundException
10  */
11
12  // Schreiben der Daten in die Datenbank
13
14  public static int insert(FileLine fileLine, Connection con)
15      throws SQLException, ClassNotFoundException, NoSuchElementException
16      {
17
18
19      int ergebnisZeilen = 0;
20
21
22      if (fileLine.getFachbereichNr() != null) {
23          stmt = con
24              .prepareStatement("INSERT INTO "
25                  + "dbo.PRODUKT"
26                  + "(fachbereichNr, handelsmarkeID, bezeichnung, trockensortiment, warengruppe, preis)
27                  + "VALUES (?, ?, ?, ?, ?, ?)");
28
29
30      // Werte setzen:
31      stmt.setString(1, fileLine.getFachbereichNr());
32      stmt.setInt(2, fileLine.getHandelsmarkeId());
```

```
33     stmt.setString(3, fileLine.getBezeichnung());
34     stmt.setString(4, fileLine.getTrockensortiment());
35     stmt.setString(5, fileLine.getWarengruppe());
36     stmt.setDouble(6, fileLine.getPreis());
37
38     System.out.println(fileLine.toString());
39     // Insert Ausführen
40     ergebnisZeilen = stmt.executeUpdate();
41
42     // Transaktion beenden:
43     con.commit();
44     con.setAutoCommit(true);
45 } else
46     throw new NoSuchElementException();
47
48 if (stmt != null)
49     try {
50         stmt.close();
51     } catch (Exception ex) {
52     }
53 stmt = null;
54
55
56
57 return ergebnisZeilen;
58 }
59
60 // Method for creating new Fachbereich
61
62 @SuppressWarnings("null")
63 public static void insertFB (String name, String fb_nr, Connection
        con) throws SQLException{
64
```

```
65
66  int ergebnisZeilen = 0;
67
68
69  String query = "INSERT INTO dbo.FACHBEREICH"
70    + "(FACHBEREICHNR, NAME)"
71    + "VALUES (?,?)" ;
72
73  PreparedStatement statement = con.prepareStatement(query);
74
75  // Fachbereich will be created with number and name
76
77  statement.setString(1,fb_nr);
78  statement.setString(2,name);
79  ergebnisZeilen = statement.executeUpdate();
80
81  con.commit();
82  con.setAutoCommit(true);
83  if (statement != null)
84    try {
85      System.out.println("Inserting FB");
86      statement.close();
87    } catch (Exception ex) {
88    }
89  statement = null;
90
91
92  }
93  }
```

## A.7. DateFormat

```
2
3 public class Date_Format {
4
5     /**
6      * Liefert das Datum im Format "dd.MM.yyyy" zurück.
7      * @param time long
8      * @return String
9      */
10    public static String getDdMMyyyyHHMi(long time) {
11        java.util.Date d = new java.util.Date(time);
12        TimeZone z = TimeZone.getDefault();
13
14        Calendar c = Calendar.getInstance(z, Locale.GERMANY);
15        SimpleDateFormat sd = new SimpleDateFormat("dd.MM.yyyy HH:mm:ss");
16        sd.setCalendar(c);
17        return sd.format(d);
18    }
19
20    /**
21     * Wandelt Datum als String in einen Longwert.
22     */
23    public static long getDdMMyyyy(String sDate) throws ParseException {
24        if ( (sDate == null) || sDate.trim().equals("")) {
25            return 0;
26        }
27        SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy");
28        // Prüft, dass es sich tatsächlich um die richtigen Datumsangaben
29        // handelt.
30        dateFormat.setLenient(false);
31        GregorianCalendar date = new GregorianCalendar();
32        date.setTime(dateFormat.parse(sDate));
33        return date.getTime().getTime();
34    }
35 }
```



```
34
35  /**
36   * Wandelt Datum als String in ein GregorianCalendar-Objekttyp.
37   */
38  public static GregorianCalendar getDayAsGregorianCalendar(String
        sDate) throws ParseException {
39      if ( (sDate == null) || sDate.trim().equals("")) {
40          return null;
41      }
42      SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy");
43      // Prüft, dass es sich tatsächlich um die richtigen Datumsangaben
        handelt.
44      dateFormat.setLenient(false);
45      GregorianCalendar date = new GregorianCalendar();
46      date.setTime(dateFormat.parse(sDate));
47      return date;
48  }
49
50  /**
51   * Wandel Date-Objekt in einen String im Format 'YYYY-MM-dd'
52   */
53  public static String getYyyyMMdd(java.sql.Date time) {
54      TimeZone z = TimeZone.getDefault();
55
56      Calendar c = Calendar.getInstance(z, Locale.GERMANY);
57      SimpleDateFormat sd = new SimpleDateFormat("yyyy-MM-dd");
58      sd.setCalendar(c);
59      return sd.format(time);
60  }
61
62  /**
63   * Wandel Long-Objekt in einen Date-Objekt
64   */
```

```
65     public static java.sql.Date getDate(long time) {
66         return new java.sql.Date(time);
67     }
68 }
```

## B. Quelltext für SQL-Skripte

### B.1. 3-Insert-Student

```
1
2  -- ***** Kassenbonerstellung
   ***** --
3
4  insert into KASSEN BON (FILIALENR, DATUM, KASSEN NR, BARGELD, SUMME)
5  VALUES
6
7  (14, CONVERT([datetime], '2019-02-10 08:22:00.000', 20), 1, 400.00,
   400.00),
8  (14, CONVERT([datetime], '2019-02-11 10:22:00.000', 20), 1, 33.00,
   33.00),
9  (14, CONVERT([datetime], '2019-02-16 08:22:00.000', 20), 1, 12.00,
   12.00),
10 (14, CONVERT([datetime], '2019-02-17 10:22:00.000', 20), 1, 123.00,
   150.00),
11 (14, CONVERT([datetime], '2019-02-20 11:29:00.000', 20), 1, 134.00,
   200.00),
12 (14, CONVERT([datetime], '2019-02-22 13:22:00.000', 20), 1, 22.00,
   100.00),
13 (14, CONVERT([datetime], '2019-02-23 14:25:00.000', 20), 1, 43.00,
   50.00),
14 (14, CONVERT([datetime], '2019-02-25 08:30:00.000', 20), 1, 40.00,
   100.00),
15 (14, CONVERT([datetime], '2019-03-10 12:29:00.000', 20), 1, 100.00,
```

```
100.00),
16 (14, CONVERT([datetime], '2019-04-22 14:20:00.000', 20), 1, 223.00,
    230.00),
17 (14, CONVERT([datetime], '2019-04-11 13:20:00.000', 20), 1, 433.00,
    450.00),
18 (14, CONVERT([datetime], '2019-05-22 15:20:00.000', 20), 1, 333.00,
    335.00),
19 (14, CONVERT([datetime], '2019-05-23 14:40:00.000', 20), 1, 555.00,
    555.00),
20 (14, CONVERT([datetime], '2019-05-24 16:22:00.000', 20), 1, 244.00,
    250.00),
21 (14, CONVERT([datetime], '2019-05-25 15:30:00.000', 20), 1, 260.00,
    260.00),
22 (14, CONVERT([datetime], '2019-05-25 16:23:00.000', 20), 1, 50.00,
    50.00),
23 (14, CONVERT([datetime], '2019-05-25 16:45:00.000', 20), 1, 78.44,
    100.00),
24 (14, CONVERT([datetime], '2019-05-25 17:44:00.000', 20), 1, 444.00,
    450.00),
25 (14, CONVERT([datetime], '2019-05-26 17:55:00.000', 20), 1, 50.00,
    50.00),
26 (14, CONVERT([datetime], '2019-05-27 12:42:00.000', 20), 1, 33.50,
    35.00),
27 (14, CONVERT([datetime], '2019-05-28 17:50:00.000', 20), 1, 447.00,
    450.00),
28 (14, CONVERT([datetime], '2019-06-02 12:44:00.000', 20), 1, 1000.00,
    1000.00),
29 (14, CONVERT([datetime], '2019-06-12 10:44:00.000', 20), 1, 23.00,
    34.00),
30 (14, CONVERT([datetime], '2019-06-12 12:44:00.000', 20), 1, 25.00,
    37.00),
31 (14, CONVERT([datetime], '2019-06-13 10:44:00.000', 20), 1, 399.00,
    400.00),
```

```
32 (14, CONVERT([datetime], '2019-06-14 12:44:00.000', 20), 1, 23.00,  
    23.00),  
33 (14, CONVERT([datetime], '2019-06-17 12:30:00.000', 20), 1, 12.00,  
    50.00),  
34 (14, CONVERT([datetime], '2019-06-20 10:33:00.000', 20), 1, 66.00,  
    100.00),  
35 (14, CONVERT([datetime], '2019-06-25 10:44:00.000', 20), 1, 33.00,  
    100.00),  
36 (14, CONVERT([datetime], '2019-06-25 10:55:00.000', 20), 1, 13.00,  
    23.00),  
37 (14, CONVERT([datetime], '2019-06-25 10:20:00.000', 20), 1, 124.00,  
    200.00),  
38 (14, CONVERT([datetime], '2019-06-28 09:55:00.000', 20), 1, 45.00,  
    50.00),  
39 (14, CONVERT([datetime], '2019-06-30 13:45:00.000', 20), 1, 66.00,  
    70.00),  
40 (14, CONVERT([datetime], '2019-07-01 16:50:00.000', 20), 1, 898.00,  
    1000.00),  
41 (14, CONVERT([datetime], '2019-08-02 10:44:00.000', 20), 1, 87.00,  
    100.00),  
42 (14, CONVERT([datetime], '2019-08-24 12:44:00.000', 20), 1, 34.00,  
    35.00),  
43 (14, CONVERT([datetime], '2019-08-25 15:26:00.000', 20), 1, 400.00,  
    500.00),  
44 (14, CONVERT([datetime], '2019-09-11 09:22:00.000', 20), 1, 298.00,  
    300.00),  
45 (14, CONVERT([datetime], '2019-09-11 12:33:00.000', 20), 1, 500.00,  
    700.00),  
46 (14, CONVERT([datetime], '2019-09-12 11:20:00.000', 20), 1, 500.00,  
    100.00)  
47  
48 -- ***** BONPOSITIONEN ***** --  
49
```

```
50
51 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
    VERKAUFSMENGE, VERKAUFSPREIS, MWST)
52 values
53 (10, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
54 (10, 2, 'N', 'N', 1, 11.9900, 11.9900 * 0.07),
55 (10, 3, 'Y', 'Y', 10, 24.900, 24.900 * 0.07);
56
57 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
    VERKAUFSMENGE, VERKAUFSPREIS, MWST)
58 values
59 (11, 2, 'Y', 'Y', 200, 2398.00, 2398.00 * 0.07),
60 (11, 15, 'Y', 'N', 10, 2.9900, 2.9900 * 0.07),
61 (11, 4, 'Y', 'Y', 10, 20.9, 20.9 * 0.07);
62
63 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
    VERKAUFSMENGE, VERKAUFSPREIS, MWST)
64 values
65 (12, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
66 (12, 21, 'Y', 'N', 1, 2.4900, 2.4900 * 0.07),
67 (12, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07);
68
69 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
    VERKAUFSMENGE, VERKAUFSPREIS, MWST)
70 values
71 (13, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
72 (13, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07),
73 (13, 15, 'Y', 'N', 1, 2.9900, 2.9900 * 0.07);
74
75 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
    VERKAUFSMENGE, VERKAUFSPREIS, MWST)
76 values
77 (14, 10, 'Y', 'Y', 20, 10.8, 10.8 * 0.07),
```

```
78 (14, 20, 'Y', 'Y', 200, 149.0, 149.0 * 0.07),
79 (14, 15, 'N', 'N', 1, 2.9900, 2.9900 * 0.07);
80
81 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
82 values
83 (15, 14, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
84 (15, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
85 (15, 15, 'N', 'N', 1, 2.9900, 2.9900 * 0.07);
86
87 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
88 values
89 (16, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
90 (16, 14, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
91 (16, 15, 'Y', 'Y', 1, 2.9900, 2.9900 * 0.07);
92
93 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
94 values
95 (17, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07),
96 (17, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
97 (17, 161, 'N', 'N', 1, 9.9000, 9.9000 * 0.07);
98
99 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
100 values
101 (18, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
102 (18, 13, 'Y', 'Y', 10, 20.9, 20.9 * 0.07),
103 (18, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07);
104
105 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
```

```
106 values
107 (19, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
108 (19, 12, 'Y', 'Y', 10, 7.900, 7.900 * 0.07),
109 (19, 15, 'Y', 'Y', 1, 2.9900, 2.9900 * 0.07);
110
111 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
112 values
113 (20, 15, 'Y', 'N', 1, 12.7900, 12.7900 * 0.07),
114 (20, 169, 'N', 'N', 1, 3.7900, 3.7900 * 0.07),
115 (20, 172, 'Y', 'Y', 10, 29.900, 29.900 * 0.07);
116
117
118 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
119 values
120 (21, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
121 (21, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
122 (21, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07);
123
124 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
125 values
126 (22, 167, 'Y', 'Y', 10, 28.900, 28.900 * 0.07),
127 (22, 169, 'Y', 'Y', 1, 3.7900, 3.7900 * 0.07),
128 (22, 172, 'Y', 'Y', 200, 2.9900, 2.9900 * 0.07);
129
130 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
131 values
132 (23, 12, 'Y', 'Y', 10, 7.90, 7.90 * 0.07),
133 (23, 158, 'Y', 'Y', 1, 3.9500, 3.9500 * 0.07),
134 (23, 160, 'Y', 'Y', 10, 45.500, 45.500 * 0.07);
```

```
135
136
137 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
138 values
139 (24, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
140 (24, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
141 (24, 16, 'Y', 'Y', 20, 59.800, 59.800 * 0.07);
142
143
144 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
145 values
146 (25, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
147 (25, 16, 'Y', 'Y', 1, 2.9900, 2.9900 * 0.07),
148 (25, 19, 'Y', 'Y', 200, 133.0, 133.0 * 0.07);
149
150 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
151 values
152 (26, 12, 'Y', 'Y', 10, 7.900, 7.900 * 0.07),
153 (26, 163, 'Y', 'Y', 1, 3.9900, 3.9900 * 0.07),
154 (26, 164, 'Y', 'Y', 10, 19.900, 19.900 * 0.07);
155
156
157 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
158 values
159 (27, 169, 'Y', 'N', 10, 37.900, 37.900 * 0.07),
160 (27, 172, 'Y', 'Y', 15, 44.85, 44.85 * 0.07),
161 (27, 159, 'Y', 'Y', 10, 45.500, 45.500 * 0.07);
162
163 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
```



```
        VERKAUFSMENGE, VERKAUFSPREIS, MWST)
164 values
165 (28, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
166 (28, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07),
167 (28, 7, 'N', 'Y', 20, 27.8, 27.8 * 0.07);
168
169 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
        VERKAUFSMENGE, VERKAUFSPREIS, MWST)
170 values
171 (29, 6, 'Y', 'N', 10, 144.900, 144.900 * 0.07),
172 (29, 4, 'Y', 'Y', 10, 127.900, 127.900 * 0.07),
173 (29, 3, 'N', 'Y', 1, 2.4900, 2.4900 * 0.07);
174
175
176 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
        VERKAUFSMENGE, VERKAUFSPREIS, MWST)
177 values
178 (30, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
179 (30, 19, 'Y', 'Y', 200, 133.0, 133.0 * 0.07),
180 (30, 6, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
181
182 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
        VERKAUFSMENGE, VERKAUFSPREIS, MWST)
183 values
184 (31, 12, 'Y', 'Y', 20, 10.8, 10.8 * 0.07),
185 (31, 19, 'Y', 'Y', 200, 133.0, 133.0 * 0.07),
186 (31, 170, 'Y', 'Y', 200, 2.1900 * 200, 2.1900 * 200 * 0.07);
187
188 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
        VERKAUFSMENGE, VERKAUFSPREIS, MWST)
189 values
190 (32, 169, 'Y', 'N', 43, 3.7900 * 43, 3.7900 * 43 * 0.07),
191 (32, 21, 'Y', 'Y', 10, 24.900, 24.900 * 0.07),
```

```
192 (32, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07);
193
194 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
195 values
196 (33, 12, 'Y', 'Y', 30, 0.7900 * 30, 0.7900 * 30 * 0.07),
197 (33, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07),
198 (33, 6, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
199
200
201 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
202 values
203 (34, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
204 (34, 19, 'Y', 'Y', 200, 133.0, 133.0 * 0.07),
205 (34, 160, 'Y', 'Y', 1, 4.5500, 4.5500 * 0.07);
206
207 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
208 values
209 (35, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
210 (35, 12, 'Y', 'Y', 10, 7.900, 7.900 * 0.07),
211 (35, 6, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
212
213 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
214 values
215 (36, 170, 'Y', 'Y', 21, 2.1900 * 21, 2.1900 * 21 * 0.07),
216 (36, 15, 'Y', 'Y', 200, 133.0, 133.0 * 0.07),
217 (36, 9, 'N', 'Y', 20, 0.7900 * 20, 0.7900 * 20 * 0.07);
218
219 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
```

```
220 values
221 (37, 172, 'Y', 'N', 20, 2.9900 * 20, 2.9900 * 20 * 0.07),
222 (37, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
223 (37, 164, 'Y', 'Y', 200, 1.9900 * 200, 1.9900 * 200 * 0.07);
224
225 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
226 values
227 (38, 18, 'Y', 'Y', 10, 21.900, 21.900 * 0.07),
228 (38, 12, 'Y', 'Y', 30, 0.7900 * 30, 0.7900 * 30 * 0.07),
229 (38, 15, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
230
231
232 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
233 values
234 (39, 165, 'Y', 'Y', 200, 1.9900 * 200, 1.9900 * 200 * 0.07),
235 (39, 168, 'Y', 'Y', 100, 100 * 2.8900, 100 * 2.8900 * 0.07),
236 (39, 6, 'N', 'Y', 3, 14.4900 * 3, 14.4900 * 3 * 0.07);
237
238 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
239 values
240 (40, 166, 'Y', 'N', 50, 14.9900 * 50, 14.9900 * 50 * 0.07),
241 (40, 165, 'N', 'Y', 8, 1.9900 * 8, 1.9900 * 8 * 0.07),
242 (40, 171, 'Y', 'Y', 15, 1.9900 * 15, 1.9900 * 15 * 0.07);
243
244 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
245 values
246 (41, 2, 'Y', 'N', 10, 11.9900 * 10, 11.9900 * 10 * 0.07),
247 (41, 3, 'N', 'N', 45, 2.4900 * 45, 2.4900 * 45 * 0.07),
248 (41, 4, 'Y', 'Y', 1, 12.7900, 12.7900 * 0.07);
```

```
249
250 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
251 values
252 (42, 163, 'N', 'N', 5, 3.9900 * 5, 3.9900 * 5 * 0.07),
253 (42, 17, 'Y', 'Y', 50, 2.1900 * 50, 2.1900 * 50 * 0.07),
254 (42, 18, 'Y', 'Y', 30, 2.1900 * 30, 2.1900 * 30 * 0.07);
255
256 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
257 values
258 (43, 169, 'Y', 'Y', 20, 3.7900* 20, 3.7900* 20 * 0.07),
259 (43, 20, 'Y', 'Y',10, 24.900, 24.900 * 0.07),
260 (43, 21, 'N', 'Y', 1, 2.4900, 2.4900 * 0.07);
261
262 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
263 values
264 (44, 170, 'Y', 'Y', 10, 21.900, 21.900 * 0.07),
265 (44, 162, 'Y', 'Y', 20, 2.9900 * 20, 2.9900 * 20 * 0.07),
266 (44, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07);
267
268 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
269 values
270 (45, 15, 'Y', 'N', 10, 29.9, 29.9 * 0.07),
271 (45, 1, 'N', 'N', 1, 12.79, 12.79 * 0.07),
272 (45, 162, 'Y', 'Y', 20, 2.9900 * 20, 2.9900 * 20 * 0.07);
273
274 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
275 values
276 (46, 12, 'Y', 'Y', 20, 10.8, 10.8 * 0.07),
```

```
277 (46, 162, 'Y', 'Y', 20, 2.9900 * 20, 2.9900 * 20 * 0.07),
278 (46, 170, 'Y', 'Y', 10, 21.900, 21.900 * 0.07)
279
280 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
281 values
282 (47, 13, 'Y', 'Y', 10, 20.9, 20.9 * 0.07),
283 (47, 21, 'Y', 'Y', 200, 149.0, 149.0 * 0.07),
284 (47, 6, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
285
286 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
287 values
288 (48, 170, 'Y', 'Y', 10, 21.900, 21.900 * 0.07),
289 (48, 162, 'Y', 'Y', 20, 2.9900 * 20, 2.9900 * 20 * 0.07),
290 (48, 15, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
291
292 insert into BONPOSITION (BONNR, PRODUKTNR, AKTIONSWARE, RABATTWARE,
      VERKAUFSMENGE, VERKAUFSPREIS, MWST)
293 values
294 (49, 13, 'Y', 'Y', 10, 20.9, 20.9 * 0.07),
295 (49, 170, 'Y', 'Y', 10, 21.900, 21.900 * 0.07),
296 (49, 6, 'N', 'Y', 2, 24.98, 24.98 * 0.07);
297
298 -- ***** Tabelle HANDELSMARKE ***** --
299
300
301 insert into HANDELSMARKE (NAME, LAND)
302 values
303 ('Schar', 'Deutschland'),
304 ('Nestle', 'Schweiz'),
305 ('Kellogg's', 'USA'),
306 ('Coca Cola', 'USA'),
```

```

307 ('Goutess', 'Deutschland'),
308 ('Gutfried', 'Deutschland'),
309 ('Jever', 'Deutschland'),
310 ('Nick', 'USA'),
311 ('Basic', 'USA');
312
313 -- ***** --
314 -- ***** Tabelle FACHBEREICH ***** --
315 -- ***** --
316
317 -- ** Gewurze wurden aus dem Java-Programm heraus hinzugefugt *** --
318 insert into FACHBEREICH (FACHBEREICHNR, NAME)
319 values
320 (1040, 'Snacks'),
321 (1041, 'Oriental'),
322 (1042, 'Asia'),
323 (1043, 'Italienisch'),
324 (1044, 'Drogerie'),
325 (1045, 'Haushalt'),
326 (1046, 'Getranke'),
327 (1047, 'Desserts');

```

## B.2. 4-ETL-DW

```

1  -- Dim-Tabellen mit Werten aus der ODB befullen --
2  --DIM_Abteilung--
3  insert into DIM_Abteilung(AbteilungNr,Name)
4  select FB.FachbereichNr,FB.Name
5  from Fachbereich as FB
6
7  --DIM_Produkt--
8  insert into
    DIM_PRODUKT(PRODUKTNR,BEZEICHNUNG,TROCKENSORTIMENT,WARENGRUPPE)

```

```
9  select P.ProduktNr, P.Bezeichnung, P.Trockensortiment,
      substring(P.Warengruppe,3,1)
10 from Produkt as P
11
12 --DIM_Filiale--
13 insert into DIM_FILIALE(FILIALENR, UST_ID_NR,STRASSE,STADT,PLZ)
14 select F.FILIALENR,F.UST_ID_NR, F.STRASSE,F.STADT, F.PLZ
15 from FILIALE as F
16
17 --DIM_Handelsmarke--
18 insert into DIM_HANDELSMARKE(HANDELSMARKENR,NAME, LAND)
19 select HM.HANDELSMARKEID, HM.NAME, HM.LAND
20 from HANDELSMARKE AS HM
21
22
23 --DIM_Date Parameter für die Berechnung der Attribute--
24 declare @vonDatum date,
25         @bisDatum date,
26         @Jahr integer, -- Attribut "Jahr"
27         @Monat integer, -- Attribut "Monat"
28         @Tag integer, -- Attribut "Tag"
29         @Quartal integer, -- Attribut "Quartal"
30         @KW integer -- Attribut "Kalenderwoche"
31
32 select @vonDatum=min(CONVERT(date,datum)) from KASSENBON
33 select @bisDatum=max(CONVERT(date,datum)) from KASSENBON
34
35 -- Zeitspanne: Schleife, bis Enddatum Erreicht wurde --
36
37 while @vonDatum <= @bisDatum
38 begin
39     set @Jahr =year(@vonDatum)
40     set @Monat =month(@vonDatum)
```

```
41     set @Tag  =day(@vonDatum)
42     set @Quartal =datepart(Quarter,@vonDatum)
43     set @KW    =datepart(WK,@vonDatum)
44
45     insert into DIM_DATE(DATEID, JAHR, MONAT, TAG, QUARTAL, KW)
46     values (@vonDatum, @jahr, @monat, @tag, @Quartal, @KW)
47
48     set @vonDatum = DATEADD(DD, 1,@vonDatum)
49 end
50
51 -- DIM_Time aud den Tag bezogen --
52 declare
53     @vonZeit time,
54     @bisZeit time,
55     @stunde int,
56     @minute int,
57     @tageszeit int,
58     @flag int
59     --Tag --
60     set @vonZeit = '00:00'
61     set @bisZeit = '23:59'
62     set @flag = 0 --unterbricht die Schleife --
63
64 while @flag = 0
65 begin
66     -- Tageszeiten bestimmen (1,2,3,4)
67     -- Schleife, bis Tagesende
68     set @stunde = DATEPART(HOUR, @vonZeit)
69     set @minute = DATEPART(MINUTE, @vonZeit)
70
71     if @stunde > 5 and @stunde < 12
72         set @tageszeit = 1
73     else if @stunde > 11 and @stunde < 18
```



```
74     set @tageszeit = 2
75     else if @stunde > 17 and @stunde <= 24
76         set @tageszeit = 3
77     else
78         set @tageszeit = 4
79
80     insert into DIM_TIME(TIMEID,STUNDE,MINUTE,TAGESZEIT)
81         values (@vonZeit, @stunde, @minute, @tageszeit)
82
83     if @vonZeit = @bisZeit
84         set @flag = 1
85
86     set @vonZeit = DATEADD(MINUTE, 1, @vonZeit)
87 end
88
89 -- Kassenbon_Fakten mit Zeiten verbinden --
90
91 INSERT INTO KASSEN_BON_FAKTEN
92 select fbr.FACHBEREICHNR, P.PRODUKTNR, hm.HANDELSMARKEID,
93 convert(date,DATEID), convert(time(0),TIMEID),
94     F.FILIALENR, sum(BP.Verkaufspreis), sum(BP.MWST),
95 sum(BP.Verkaufspreis) - sum(BP.MWST),
96 sum(BP.Verkaufsmenge),
97 bp.AKTIONSWARE,
98 bp.RABATTWARE
99
100 from
101 FACHBEREICH as fbr
102 join Produkt as P
103     on fbr.FACHBEREICHNR= P.FachbereichNr
104 join
105     HANDELSMARKE as hm
106     on P.HandelsmarkeId = hm.HANDELSMARKEID
```

```
107 join
108   Bonposition as bp
109   on P.ProduktNr = bp.ProduktNr
110 join
111   Kassenbon as KB
112   on bp.BonNr = KB.BonNr
113 join
114   FILIALE as F
115   on KB.FilialeNr = F.FILIALENR
116 join
117   Dim_Date as DD
118   on convert(date,KB.Datum) = DD.DateID
119 join
120   Dim_Time as DT
121   on convert(time(0),KB.Datum) = DT.TimeID
122 group by
123   fbr.FACHBEREICHNR, P.PRODUKTNR, hm.HANDELSMARKEID,
124   DD.DATEID, DT.TIMEID, F.FILIALENR,
125   BP.AKTIONSWARE, BP.RABATTWARE,BP.BONNR
```

### B.3. 5-Fakten-Woche

```
1  create table KASSEN BON_FAKTEN_WOCHEN (
2  ABTEILUNG NR int not null references DIM_ABTEILUNG (ABTEILUNG NR),
3  PRODUKT NR int not null references DIM_PRODUKT (PRODUKT NR),
4  HANDELS MARKEN NR int not null references DIM_HANDELS MARKE
   (HANDELS MARKEN NR),
5  JAHR int not null,
6  WOCH E int not null,
7   TIMEID time not null,
8  FILIALE NR char(3) not null references DIM_FILIALE (FILIALE NR),
9  Umsatz float not null,
10 ABGABEMWST float not null,
```

```
11 NETTO float not null,
12 MENGE int not null,
13 AKTIONSARTIKEL char(1) not null,
14 RABATTIERT char(1) not null,
15 constraint PK_FAKTEN_WOCHEN primary key
    (ABTEILUNGNR,PRODUKTNR,HANDELSMARKENR, JAHR, TIMEID,
     WOCHEN,FILIALENR))
16
17
18
19 INSERT INTO KASSEN_BON_FAKTEN_WOCHEN
20 select
21 FB.FACHBEREICHNR,
22 P.PRODUKTNR,
23 H.HANDELSMARKEID,
24 DD.JAHR,
25 DD.KW,
26     convert(time(0),TIMEID),
27 F.FILIALENR,
28 sum(BP.Verkaufspreis),
29 sum(BP.MWST),
30 sum(BP.Verkaufspreis)-sum(BP.MWST),
31 sum(BP.Verkaufsmenge),
32 BP.AKTIONSWARE,
33 BP.RABATTWARE
34 from
35 FACHBEREICH as FB
36 join Produkt as P
37   on FB.FACHBEREICHNR= P.FachbereichNr
38 join
39   HANDELSMARKE as H
40   on P.HandelsmarkeId = H.HANDELSMARKEID
41 join
```

```
42 Bonposition as BP
43 on P.ProduktNr = BP.ProduktNr
44 join
45 Kassenbon as KB
46 on BP.BonNr = KB.BonNr
47 join
48 FILIALE as F
49 on KB.FilialeNr = F.FILIALENR
50 join
51 Dim_Date as DD
52 on DATEPART(yyyy,KB.Datum) = DD.JAHR and
53 DATEPART(wk,KB.DATUM) = DD.KW
54 join
55 Dim_Time as DT
56 on convert(time(0),KB.Datum) = DT.TIMEID
57 group by
58 FB.FACHBEREICHNR,
59 P.PRODUKTNR,
60 H.HANDELSMARKEID,
61 DD.JAHR,
62 DD.KW,
63 DT.TIMEID,
64 F.FILIALENR,
65 BP.AKTIONSWARE,
66 BP.RABATTWARE
```

#### B.4. 6-DW-Report

```
1 --- *** PROZEDUR mit Cursor für das Datum ** ---
2
3 --- *** Parameter zum Aufruf der Prozedur: Produktnummer und
   Zeitspanne ** --
4
```

```
5 -- *** drop procedure DW_REPORT
6
7 CREATE PROCEDURE DW_REPORT (@produkt int ,@vonDatum date, @bisDatum
    date) AS
8 BEGIN
9     BEGIN TRANSACTION
10    if @vonDatum > @bisDatum
11        print 'ERROR. Eingabe überprüfen'
12    else
13        insert into DW_REPORT_TABLE
14        select PRODUKTNR, DATEID, sum(UMSATZ) as umsatz, sum(MENGE) as menge
            from KASSEN BON_FAKTEN
15    WHERE
16    DATEID BETWEEN @vonDatum AND @bisDatum
17    and Produktnr =@produkt
18    group by DATEID, PRODUKTNR
19
20    declare @ProduktNr char(2), @von date, @bis date, @umsatz char(15),
        @menge char(4), @gesums char(10);
21    set @von = @vonDatum
22    set @bis = @bisDatum
23
24    declare KB_Cursor CURSOR FOR
25    SELECT Datum as Datum FROM DW_REPORT_TABLE
26    OPEN KB_Cursor
27    FETCH NEXT FROM KB_Cursor into @von
28    print 'ProduktNr Menge in St. Datum Umsatz in EU'
29    while @@FETCH_STATUS = 0
30    begin
31        select @umsatz = cast(umsatz as char) from DW_Report_TABLE where
            Datum = @von
32        select @menge= cast(menge as char) from DW_Report_TABLE where Datum
            = @von
```

```
33  select @ProduktNr = cast(ProduktNr as char) from DW_Report_TABLE
      where Datum = @von
34  print @ProduktNr + ' | '+@menge+ ' | '+cast(@von as char)+' |
      '+@umsatz+' | '
35  print
      '-----
36  FETCH NEXT FROM KB_Cursor into @von
37  end
38  select @gesums = cast(sum(umsatz) as char) from DW_REPORT_TABLE
39  print' | Gesamtumsatz (EU) : '+@gesums+' | '
40  close KB_Cursor
41  deallocate KB_Cursor
42  delete from DW_REPORT_TABLE
43  commit transaction
44  END
45
46
47  -- ** fur den Aufruf der Prozedur Beispielparameter auswählen ** --
48
49
50  declare @produktnummer int, @von date, @bis date
51  execute DW_REPORT 15, '2019-02-01', '2019-08-01'
```

## B.5. 7-Hilfstabelle

```
1
2
3  -- Hilfstabelle fur den Report --
4
5  create table DW_REPORT_TABLE(
6  ProduktNr int,
7  Datum datetime,
8  Umsatz float,
```

9 Menge `int`)

## **Abschließende Erklärung**

Ich versichere hiermit, dass ich meine Masterarbeit selbständig und ohne fremde Hilfe angefertigt habe, und dass ich alle von anderen Autoren wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren eng anlegenden Ausführungen meiner Arbeit besonders gekennzeichnet und die Quellen zitiert habe.

<ORT>, den 13. Januar 2020

<AUTOR>



ProduktNr	Menge in St.	Datum	Umsatz in €
15	10	2019-02-11	2.99
15	10	2019-02-16	29.9
15	1	2019-02-17	2.99
15	1	2019-02-20	2.99
15	1	2019-02-22	2.99
15	1	2019-02-23	2.99
15	10	2019-02-25	29.9
15	10	2019-03-10	29.9
15	1	2019-04-11	12.79
15	1	2019-04-22	2.99
15	10	2019-05-22	29.9
15	10	2019-05-25	29.9
15	10	2019-06-13	29.9
15	200	2019-06-17	133
15	2	2019-06-25	24.98
Gesamtumsatz (€) :			368.11

Abbildung 7: Reporting-Tabelle