

# HÖHERE ABTEILUNG FÜR WIRTSCHAFTSINGENIEURWESEN

HTBLuVA Waidhofen/Ybbs  
Im Vogelsang 8, 3340 Waidhofen/Ybbs

## Absolventendatenbank



*Schuljahr:* 2006/2007

*Verfasser:* Bianca Ecker  
Michael Großberger

*Prüfungsfach:* Programm- und Projektentwicklung

*Prüfer:* Prof. Dipl.-Ing. Horst Dietrich

*Betreuer:* Dipl.-Ing. Gerald Pracherstorfer  
Prof. Dipl.-Ing. Horst Dietrich  
VL Christian Schneider

***Besonderen Dank an...***

*... . Dipl.-Ing. Gerald Pracherstorfer*

*... Prof. Dipl.-Ing. Horst Dietrich*

*... VL Christian Schneider*

*... Dr. MMag. Jürgen Krajsek*

## Inhaltsverzeichnis

<b>1. Vorwort</b>	<b>8</b>
<b>2. Zusammenfassung</b>	<b>9</b>
<b>3. Summary</b>	<b>11</b>
<b>4. Begriffe und Abkürzungen</b>	<b>12</b>
<b>5. Aufgabenstellung</b>	<b>13</b>
<b>6. Unser Team</b>	<b>14</b>
<b>7. Betreuerteam</b>	<b>15</b>
<b>8. Lastenheft</b>	<b>16</b>
8.1. Zielbestimmung	16
8.2. Produkteinsatz	16
8.3. Produktfunktionen	16
<b>9. Pflichtenheft</b>	<b>17</b>
9.1. Auftraggeber	17
9.2. IST-Zustand	17
9.3. Zielbestimmung	17
9.3.1. MUSSANFORDERUNGEN	17
9.3.2. WUNSCHANFORDERUNGEN	18
9.3.3. ABGRENZUNGSKRITERIEN	18
9.4. Produkteinsatz	19
9.4.1. ANWENDUNGSBEREICHE	19
9.4.2. ZIELGRUPPEN	19
9.4.3. BETRIEBSBEDINGUNGEN	19

9.5. Produktübersicht	19
9.6. Produktfunktionen	19
9.7. nicht funktionale Anforderungen	20
9.8. technische Umgebung	21
9.8.1. SOFTWARE	21
9.9. Dokumentationsanforderungen	21
<b>10. Aufgabenaufteilung</b>	<b>22</b>
<b>11. Ressourcen- &amp; Meilensteinplanung</b>	<b>23</b>
11.1. Ressourcenplanung	23
11.2. Meilensteinplanung	24
<b>12. Allgemeine Informationen</b>	<b>25</b>
12.1. Datenbanksystem	25
12.1.1. ALLGEMEIN	25
12.1.2. FUNKTIONSWEISE	25
12.1.3. GUNDSÄTZE DER DATENBANKPLANNUNG:	25
12.2. Normalformen	26
12.2.1. ERSTE NORMALFORM	26
12.2.2. ZWEITE NORMALFORM	26
12.2.3. DRITTE NORMALFORM	26
12.3. Visual Basic for Application	27
12.4. Structured Query Language	27
<b>13. Datenbankstruktur</b>	<b>28</b>
13.1. Tabelle Stammdaten	29
13.1.1. ALLGEMEINES	29
13.1.2. ELEMENTE	29

<b>13.2. Tabelle Absolventendaten</b>	<b>30</b>
13.2.1. ALLGEMEINES	30
13.2.2. ELEMENTE	30
<b>13.3. Tabelle Beiträge</b>	<b>30</b>
13.3.1. ALLGEMEINES	30
13.3.2. ELEMENTE	30
<b>13.4. Tabelle Aktuelle Adresse</b>	<b>31</b>
13.4.1. ALLGEMEINES	31
13.4.2. ELEMENTE	31
<b>13.5. Tabelle ursprüngliche Adresse</b>	<b>32</b>
13.5.1. ALLGEMEINES	32
13.5.2. ELEMENTE	32
<b>13.6. Tabelle Ort</b>	<b>32</b>
13.6.1. ALLGEMEINES	32
13.6.2. ELEMENTE	32
<b>13.7. Tabelle e-mail</b>	<b>33</b>
13.7.1. ALLGEMEINES	33
13.7.2. ELEMENTE	33
<b>13.8. Tabelle neue Absolventen</b>	<b>33</b>
13.8.1. ALLGEMEINES	33
13.8.2. ELEMENTE	34
<b>13.9. Tabelle laufende Nummer</b>	<b>34</b>
13.9.1. ALLGEMEINES	34
13.9.2. ELEMENTE	35
<b>14. Erlagscheinnummer</b>	<b>36</b>
<b>14.1. Aufbau der Erlagscheinnummer</b>	<b>36</b>

14.1.1. ABTEILUNG	36
14.1.2. JAHRGANG	36
14.1.3. KLASSE	37
14.1.4. LAUFENDE NUMMER	37
<b>15. Formulare und ihre Funktionen</b>	<b>38</b>
<i>15.1. Grundlegendes</i>	<i>39</i>
15.1.1. ALLGEMEIN	39
15.1.2. ARBEITEN MIT DEN FORMULAREN	39
<i>15.2. Das Hauptformular</i>	<i>40</i>
15.2.1. ALLGEMEIN	41
15.2.2. FELDER UND SCHALTFLÄCHEN	41
15.2.2.1. laufende Nummer	41
15.2.2.2. Erlagscheinnummer	41
15.2.2.3. Nachname & Vorname	42
15.2.2.4. Adressdaten	42
15.2.2.5. Titel, Bemerkung & Sonstiges	42
15.2.2.6. Unterformular Beiträge	43
15.2.2.7. Schaltfläche Beiträge - Jahrbücher	43
15.2.2.8. Schaltfläche Adressen	43
15.2.2.9. Schaltfläche Abschlussdaten	44
15.2.2.10. Schaltfläche Aktionen	44
<i>15.3. Beiträge - Jahrbücher</i>	<i>45</i>
15.3.1. ALLGEMEIN	46
15.3.2. FELDER, SCHALTFLÄCHE UND UNTERFORM.	46
15.3.2.1. laufende Nummer	46
15.3.2.2. Erlagscheinnummer	47
15.3.2.3. Nachname & Vorname	47
15.3.2.4. Adressdaten	47

15.3.2.5. Beitragsdaten	48
15.3.2.6. Schaltfläche Beitrag aktualisieren	48
15.3.2.7. Unterformular alte Beiträge	49
<b>15.4. Adressen</b>	<b>50</b>
15.4.1. ALLGEMEIN	50
<b>15.5. aktuelle Adresse</b>	<b>51</b>
15.5.1. ALLGEMEIN	52
15.5.2. FELDER	52
<b>15.6 Ursprungsadresse</b>	<b>54</b>
15.6.1. ALLGEMEIN	55
15.6.2. FELDER	55
<b>15.7. e-mail</b>	<b>56</b>
15.7.1. ALLGEMEIN	57
15.7.2. FELD UND UNTERFORMULAR	57
<b>15.8. Abschlussdaten</b>	<b>58</b>
15.8.1. ALLGEMEIN	59
15.8.2. FELDER	59
<b>15.9. Aktionen</b>	<b>60</b>
15.9.1. ALLGEMEIN	61
15.9.2. SCHALTFLÄCHE ABSOLVENTEN HINZUFÜGEN	61
15.9.2.1. Ausgangsdokument des Sekretariats	61
15.9.2.2. Einlesen in die Datenbank	62
15.9.2.3. Generieren der neuen Erlagscheinnummern	63
15.9.2.4. Schreiben der Tabellen	65
15.9.3. RUNDSCHREIBEN	67
15. 9.3.1. ALLGEMEINES	67
15.9. 3.2. SQL-Abfrage „A_Rundschreiben“	69

15. 9. 3.2.1. CODE	69
15.9.3.2.2. AUSGABE-Felder	69
15.9.3.2.3. Tabellen und Beziehungen	70
15.9.3.2.4. Auswahlbedingungen	70
15.9.3.2.5. Sortierung	71
15.9.3.3. Button „Rundschreiben Tabellenansicht“	72
15.9.3.4. Schaltfläche „Serienbrief“	72
15.9.5. ETIKETTENDRUCK	73
15.9.4.1. ALLGEMEINES	73
15. 9.4.2. SQL-Abfrage „A_Etikettendruck“	74
15. 9. 4.2.1. Code	74
15. 9. 4.2.2. Ausgabefelder	74
15. 9. 4.2.3. Tabellen und Beziehungen	75
15. 9. 4.2.4. Auswahlbedingungen	75
15. 9. 4.2.5. Sortierung	75
15. 9. 4.3. SCHALTFLÄCHE „Etikettendruck“	76
15. 9. 4.4. SCHALTFLÄCHE „Verschickt Abhaken“	77
15. 9. 4.4.1. SQL-Abfrage „A_Verschickt“	77
15. 9.5. KLASSENLISTE	78
15. 9.5.1. ALLGEMEINES	78
15. 9. 5.2. SQL-Abfrage „A_Klassenliste“	79
15. 9. 5.2.1. CODE	79
15. 9. 5.2.2. Ausgabefelder	79
15. 9. 5.2.3. Tabellen und Beziehungen	80
15. 9. 5.2.4. Auswahlbedingungen	80
15. 9. 5.2.5. Sortierung	81
15. 9. 5.3. Schaltfläche „Klassenliste – Bericht“	81
15. 9. 5.4. Schaltfläche „Klassenliste exportieren“	81
15. 9.6. ABSOLVENTINNEN	82



15. 9.6.1. Schaltfläche „Absolventinnen der HTL“	82
15. 9. 6.2. SQL-Abfrage „A_Absolventinnen“	82
15. 9. 6.2.1. Code	82
15. 9. 6.2.2. Ausgabefelder	82
15. 9. 6.2.3. Tabellen und Beziehungen	83
15.9. 6.2.4 Auswahlbedingung und Sortierung	83
<b>16. Einspielen der Absolventendaten</b>	<b>84</b>
16.1. Alte Datenbankstruktur	84
16.2. Nötige Änderungen an der Originaldatenbank	86
16.3. Programme zum Einlesen der Daten	87
16.3.7. TABELLE URSPRUNGSADRESSE	88
16.3.6. TABELLE ORT	88
16.3.5. TABELLE AKTUELLE ADRESSE	89
16.3.4. TABELLE BEITRÄGE	90
16.3.3. TABELLE E-MAIL	90
16.3.2. TABELLE ABSOLVENTENDATEN	91
16.3.1. TABELLE STAMMDATEN	92
<b>17. Sicherheit</b>	<b>93</b>
17.1. Login	93
17.2. Autoexec	93
<b>18. Backend &amp; Frontend</b>	<b>94</b>
<b>19. Resümee</b>	<b>94</b>
<b>20. Arbeitslisten</b>	<b>95</b>
<b>21. Besprechungsprotokolle</b>	<b>107</b>

## 1. Vorwort

Ganz herzlich möchten wir uns bei Herrn VL Christian Schneider bedanken. Er ist der Betreuer der Absolventendatenbank und von ihm stammt auch die Idee, die Datenbank neu zu gestalten. Er hat uns die regelmäßigen Arbeitsabläufe und Anforderungen des Absolventenvereins genau und klar geschildert. Weiters möchten wir betonen, dass er immer so freundlich war, sich Zeit für uns zu nehmen und sich stets bemüht hat alle offenen Fragen sofort zu beantworten.

Weiters möchten wir uns natürlich auch bei unseren Betreuern bedanken. Herr Dipl.-Ing. Gerald Pracherstofer ist uns eine wichtige Stütze. Er ließ uns sehr gute Unterlagen zukommen, durch die wir gleich zu Anfang ersehen konnten, was von uns verlangt wird. Herr Pracherstorfer unterstütze uns vor allem bei der Erstellung der neuen Grundstruktur der Datenbank und bei den Abfragen. Er war aber auch bei Programmierfragen immer mit Rat und Tat zur Stelle.

Herr Prof. Dipl.-Ing. Horst Dietrich unterstützte uns hauptsächlich bei speziellen Fragen zum Thema Programmierung. Ohne seine Tipps wäre wohl vieles langsamer und komplizierter von Statten gegangen. Im Speziellen betreute er auch die Erstellung des Backends.

Zuletzt richten wir einen speziellen Dank an unseren Klassenvorstand Herrn Dr. MMag. Jürgen Krajsek. Er war ebenfalls immer für uns da, achtete auf den Fortgang unserer Arbeit und beantwortete uns gerne Fachfragen.

## **2. Zusammenfassung**

### **Problemstellung**

Da die Absolventendatenbank unserer Schule schon seit Jahrzehnten existiert, ist sie vom anfänglichen Karteikartensystem bis zur heutigen computerverwalteten Datenbank, ständig gewachsen. So wurden wir gebeten den Umgang mit der Absolventendatenbank zu vereinfachen und zu automatisieren.

### **Aufgabenstellung**

Unsere Aufgabe war es, die Absolventendatenbank der HTBLuVA Waidhofen/Ybbs grundlegend umzustrukturieren. Dass hieß, eine klarere Tabellenstruktur zu erstellen, die Datenverwaltung professionell zu gestalten und eine einfache und schnelle Handhabung des Gesamtsystems zu ermöglichen. Dies war durch die Schaffung von Eingabemasken möglich, die den Anforderungen des Systems angepasst sind. Diesen Veränderungen waren notwendig, da das Datenbanksystem auf einem über zehn Jahre alten Stand beruhte und der Umgang mit den Daten sehr kompliziert und aufwendig war.

### **Ziel**

Unser Ziel war die Umstrukturierung der Tabellen und eine Zweiteilung der Datenbank.

Für den Frontendbereich wurde eine neue Eingabemaske mit einem optisch ansprechenden und klar strukturierten Layout geschaffen. Es beinhaltet die Sortier- und Suchfunktionen, die dem Benutzer die Arbeit mit der Datenbank erheblich erleichtern und auch für Laien leicht zu handhaben sind. Da die Absolventendatenbank laufend sehr aufwendig erweitert wurde, haben wir auch diesen Prozess automatisiert. Das Frontend ist also der Teil unserer Arbeit, der für den Benutzer sichtbar ist. Er liegt in Access.

Das Backend ist ein SQL-Server, der von dem Programm MS Desktop Engine verwaltet wird. Hier liegt die Datenbank, während von Access aus nur auf die Datensätze zugegriffen wird.

Weiters wurde eine Clientstruktur mit verschiedenen Zugriffsrechten geschaffen, um auch systemfremden Personen die Mitarbeit an der Datenbank zu ermöglichen.

<h2>Ergebnis</h2>
-------------------

Das Projektergebnis wird dem Betreuer des Absolventenvereins, VI. Ing. Christian Schneider, zur Verfügung gestellt. Wir hoffen, Herr Schneider wird mit der neuen Absolventendatenbank zufrieden sein, und bedanken uns in diesem Zuge auch für seine Aufgeschlossenheit und die gute Zusammenarbeit.

### 3. Summary

The task is to restructure the software database of the graduate's association of our school, HTBLuVA Waidhofen/Ybbs, fundamentally.

This means to build a clear structure in form of tables, to design the data administration professional and to make an easy and fast handling of the whole system possible.

This is possible with the creation of input masks, which are adapted to the demands of the system.

The changes are necessary, because the database system is based on a state more than ten years ago and the dealing with the data is complicated.

#### **Necessary data, that should be kept:**

- Address data including e-mail and title
- Opening - Mr/Mrs
- old address, when the address changes
- class and year of graduation; two times when the graduate finished the technical college after higher technical school
- subscription fee received
- annual report - yes/no
- remarks
- twelve-digit number on pay-in slip

#### **Scope of supply**

- Printing lists of classes for class reunions – in order of class
- Printing of pay-in slips for broadcasts – in order of post code
- Dispatch of the annual reports in stages – marking, when the annual report has been already sent this year
- Adding of the new graduates every year
- Simple input mask for address data, payment, and annual report
- Comfortable search function inside the input mask to search for names and serial numbers

#### **Miscellaneous**

- Records in the database never will be deleted (Even the dead and those, where the address is unknown have to be on the class list)

## 4. Begriffe und Abkürzungen

### BEGRIFFE:

DB	Datenbank, ist ein System zur elektronischen Datenverwaltung
Tabelle	Beinhaltet thematisch zusammengehörige Datensätze
Tupel	Zelle einer Tabelle
Datensatz	Zeile einer Tabelle
Feld	Spalte einer Tabelle
Attribut	Spaltenüberschrift
Relation	Die Daten werden in Tabellenform gespeichert.
Beziehung	Der Zusammenhang der einzelnen Tabellen
NF	Normalform
VBA	Visual Basic for Application, ist eine eigenständige, objektorientierte Programmiersprache für die MS Office-Anwendungen
SQL	Structured Query Language, ist eine Datenbankabfragesprache
MS DE	Microsoft SQL Desktop Engine, ist ein relationales Datenbankmanagementsystem auf der Microsoft-Windows-Plattform
MS ACCESS	Ist ein Datenbankverwaltungsprogramm, welches nicht nur Daten durch Abfragen und Makros filtern kann, sondern durch VBA-Programmierung auch neue Funktionen erstellt.

### ABKÜRZUNGEN:

JB	Jahrbuch
BT	Beiträge
lfdNr	fortlaufende Nummer
DB	Datenbank
DBMS	Datenbankmanagementsystem

## **5. Aufgabenstellung**

Unsere Aufgabe ist es, die Absolventendatenbank der HTBLuVA Waidhofen/Ybbs grundlegend umzustrukturieren. Das heißt, eine klarere Tabellenstruktur zu erstellen, die Datenverwaltung professionell zu gestalten und eine einfache und schnelle Handhabung des Gesamtsystems zu ermöglichen. Dies ist durch die Schaffung von Eingabemasken möglich, die den Anforderungen des Systems angepasst sind. Diesen Veränderungen sind notwendig, da das Datenbanksystem auf einem über zehn Jahre alten Stand beruhte und der Umgang mit den Daten sehr kompliziert und aufwendig war.

### **DIE NOTWENDIGEN DATEN, DIE ERHALTEN WERDEN SOLLEN:**

- Adressdaten inkl. E-Mail und Titel
- Anrede Herr/Frau
- alte Adresse bei Adressänderung
- Klasse und Jahr der Matura oder Abschlussprüfung - doppelt für Absolventen, die nach der Fachschule noch Matura gemacht haben
- Einbezahlter Beitrag
- Jahresbericht Ja/Nein
- Bemerkung
- 12-stellige Nummer für Erlagschein

### **LEISTUNGSUMFANG:**

- Drucken von Listen für Klassentreffen - Sortierung nach Klassen
- Drucken von Erlagscheinen für Rundschreiben - Serienbrief (sortiert nach Postleitzahlen)
- Versand der Jahresberichte - auf mehrere Etappen d.h. Kennzeichnung, wenn in diesem Jahr schon ein JB geschickt wurde
- Hinzufügen der neuen Absolventen jedes Jahr
- Einfache Eingabemaske für Adressendaten, Beitrag und JB
- komfortable Suchfunktion nach Namen und fortlaufender Nummer innerhalb der Eingabemaske

### **SONSTIGES:**

- Es werden niemals Datensätze gelöscht (auf Klassenliste müssen auch Verstorbene sein, bzw. jene wo Adresse unbekannt)

## 6. Unser Team



**Bianca Ecker**

Kronhobl 56  
3340 Waidhofen/Ybbs

Tel: 0650/9147918  
e-mail: [bianca.ecker@gmx.at](mailto:bianca.ecker@gmx.at)  
Klasse 5AHWII



**Michael Großberger**

Bertastr. 29/8  
3340 Waidhofen/Ybbs

Tel: 0650/2785666  
e-mail: [michael@grossberger.co.at](mailto:michael@grossberger.co.at)  
Klasse 5AHWII



## 7. Betreuer team



**DI Pracherstorfer Gerald**

e-mail:

[gerald.pracherstorfer@htlwy.ac.at](mailto:gerald.pracherstorfer@htlwy.ac.at)



**Prof. Dipl.-Ing. Horst Dietrich**



**VL Christian Schneider**

## **8. Lastenheft**

### **8.1. Zielbestimmung**

Unser Ziel ist die klare Umstrukturierung der Tabellen und eine Zweiteilung der Datenbank in einen Frontend- und einen Backendbereich. Es werden neue Eingabe- und Suchfunktionen geschaffen, die auch für Laien leicht zu handhaben sind. Weiters wird eine Clientstruktur mit verschiedenen Zugriffsrechten geschaffen, um auch systemfremden Personen die Mitarbeit an der Datenbank zu ermöglichen. Da die Absolventendatenbank laufend erweitert wird und dies bisher sehr aufwendig war, werden wir diesen Prozess automatisieren.

### **8.2. Produkteinsatz**

Der Produkteinsatz erfolgt rein Schulintern, da die Absolventendaten dem Datenschutzgesetz unterliegen und dementsprechend vertraulich zu behandeln sind.

### **8.3. Produktfunktionen**

Da die Vorgängerdatenbank keinerlei Funktionen enthalten hat, müssen alle Funktionen neu geschaffen werden. Die wichtigsten Funktionen beziehen sich auf das Anlegen neuer Absolventen, den laufenden Umgang mit den Daten, Abfragen verschiedener Arten, das Verschicken von Briefen und Jahrbüchern und natürlich die Datensicherung.

## **9. Pflichtenheft**

### **9.1. Auftraggeber**

Unser Auftraggeber ist der Absolventenverein der HTLuVA Waidhofen an der Ybbs. Herr VL Christian Schneider ist der Hauptverantwortliche und für die Führung der Absolventendatenbank verantwortlich. Er verwaltet die Daten aller Schulabgänger.

### **9.2. IST-Zustand**

Die Absolventendatenbank wird seit einigen Jahren im MS ACCESS geführt und besitzt kein Backend. Da die Absolventendatenbank auf dem Schulrechner der CRW liegt, wird fast nur in der Schule gearbeitet. Wenn Herr Schneider zu Hause arbeiten will, muss er den aktuellen Datenstand über Datenträger zu sich mitnehmen. Die Absolventendatenbank umfasst bereits ca. 4.600 Absolventen. Sämtliche Daten befinden sich in einer Tabelle. Bis auf ein alles umfassendes Formular gibt es keinerlei bestehende Funktionen zum Abrufen dieser Daten.

### **9.3. Zielbestimmung**

#### **9.3.1. MUSSANFORDERUNGEN:**

- Es sollen Listen für Klassentreffen abgefragt und anschließend gedruckt werden -> Sortierung nach Klassen
- Weiters soll das automatische Drucken von Erlagscheinen für Rundschreiben ermöglicht werden, wobei die Ausgabe nach Postleitzahlen sortiert erfolgen soll.
- Versand der Jahresberichte erfolgt auf mehrere Etappen, d.h. es müssen die noch nicht verschickten aber angeforderten JB gekennzeichnet und ausgeschickt werden.
- Hinzufügen der neuen Absolventen jedes Jahr
- Einfache Eingabemaske für Adressendaten, Beitrag und JB
- komfortable Suchfunktion nach Namen und fortlaufender Nummer innerhalb der Eingabemaske

### **9.3.2. WUNSCHANFORDERUNGEN:**

Durch die zusätzlichen Veränderungen wird der Umgang mit der Datenbank in Zukunft klarer, übersichtlicher und genauer sein. Dadurch wird die Handhabung komfortabler und man wird sich viel Zeit ersparen.

- Die Daten werden thematisch auf mehrere Tabellen aufgegliedert um die drei Normalformen zu gewährleisten.
- Manche Daten werden nicht mehr mit aufgenommen, da sie nicht unbedingt nötig sind.
- Es werden mehrere Eingabemasken geschaffen, die den üblichen Anwendungen entsprechen, untereinander verknüpft sind und einem übersichtlichen, angenehmem Layout entsprechen.
- Die geforderten Funktionen sind über Schaltflächen des Hauptformulars aufzurufen.
- Es wird auch ein Backend geschaffen (SQL Server), um die Datensicherheit besser gewährleisten zu können.
- Es werden User angelegt und Rechte vergeben
- Es werden komfortable Such- und Sortierfunktionen erschaffen
- Die fortlaufende Nummer wird automatisch generiert und hinzugefügt
- Durch die Schaffung einer AUTOEXEC- Funktion wird immer der letzte Datenstand mitgesichert.
- Es wird der Zugriff auf die Datenbank für den Administrator auch via Internet über den Schulserver möglich sein.

### **9.3.3. ABGRENZUNGSKRITERIEN:**

- Es kann nicht garantiert werden, dass alle Bildschirmauflösungen eine optimale Darstellung bereitstellen.
- Es ist nicht sicher ob das Programm auf höheren Betriebssystem- oder Programmversionen fehlerfrei läuft.
- Hundertprozentige Ausfallsicherheit kann nicht gewährleistet werden jedoch absolute Datensicherheit.

## 9.4. Produkteinsatz

### 9.4.1. ANWENDUNGSBEREICHE:

- Der Produkteinsatz erfolgt rein Schulintern, da die Absolventendaten dem Datenschutzgesetz unterliegen und dementsprechend vertraulich zu behandeln sind. Es wird der Zugriff auf die Datenbank für den Administrator auch via Internet über den Schulserver möglich sein, da sich auf diesem das Backend befindet. Es muss lediglich die neue DB-Struktur auf dem Home PC vorhanden sein.

### 9.4.2. ZIELGRUPPEN:

Der Zugriff auf die DB ist vorrangig für den Absolventendatenbankverwalter (Administrator) vorgesehen. Es ist jedoch möglich auch Systemfremde Personen beschränkt mitarbeiten zu lassen (Clients).

### 9.4.3. BETRIEBSBEDINGUNGEN:

Die Absolventendatenbank soll immer aufrufbar sein, ausgenommen wenn es am Server Probleme gibt, da sich das Backend auf diesem befindet.

## 9.5. Produktübersicht

Es wird ein Verwaltungssystem für die Absolventendatenbank erstellt. Dieses soll von jedem PC, der das Frontend installiert hat, ausführbar sein. Dazu benötigt man jedoch die Administratorrechte um auf das Backend zugreifen zu können. Dies wird vorrangig vom Lehrer PC der CMW und dem privaten PC von Herrn Schneider aus geschehen.

## 9.6. Produktfunktionen

- Automatisches Übernehmen der neuen Absolventen in die DB:
  - Generieren einer neuen Erlagscheinnummer
  - Übernahme der vorhandenen Daten in die jeweiligen Tabellen
- e-mail:
  - Bei e-mail Adressen ist eine Mehrfachangabe möglich.

- Jahresberichte auf Etappen verschicken:
  - Es wird eine Abfrage vorgenommen, in der ermittelt wird wer einen Jahresbericht wünscht und noch keinen erhalten hat.
  - Die Namen der Interessenten werden automatisch im Etikettenformat dargestellt.
  - Danach wird das Feld verschickt auf true gesetzt.
- Adressen:
  - Unterteilung in aktuelle und Ursprungsadresse
  - Die Post des Absolventenvereins geht nur an aktuelle Adressen.
  - Für Klassentreffen wird die aktuelle Adresse angegeben, wenn diese nicht vorhanden ist die Ursprungsadresse.
- Rundschreiben:
  - Angabe der benötigten Personendaten
  - Es wird an alle mit aktueller Adresse verschickt
  - Vor dem Erstellen wird nach 2 Kriterien aussortiert: PLZ und Land – erst wird das Ausland und dann Österreich bearbeitet.
  - Es wird auf Erlagscheine gedruckt
- komfortable Suchfunktion nach Namen und fortlaufender Nummer innerhalb der Eingabemaske
- Schaffung einer Administrator- / Clientstruktur
- Automatisches Erzeugen eines neuen Datensatzes bei der Eingabe eines Beitrages
- AUTOEXEC-Funktion, um den letzten Datenstand zu sichern
- Beim Ändern eines Datensatzes wird dieser sofort gespeichert und beim Schließen des Programms wird alles gespeichert.

## **9.7. nicht funktionale Anforderungen**

- Plattformunabhängigkeit
- Backup – System für die gesamte Datenbank

## 9.8. technische Umgebung

### 9.8.1. SOFTWARE:

- MS ACCESS
- MS Desktop Engine

## 9.9. Dokumentationsanforderungen

Die Dokumentation soll so genau wie möglich, jedoch nicht mit überschüssigen Informationen erstellt werden. Sie dient dazu, um sowohl Entwickler als auch dem User über die Umgebung vertraut zu machen. Die Dokumentation wird sowohl in elektronischer Form, also auch in Papierform zur Verfügung stehen.

## 10. Aufgabenaufteilung

Gemeinsam: Erstellen der neuen Datenbankstruktur

Ecker: Erstellen der Formulare und des Layouts

Ecker: **VBA Programmierung:**

komfortable Suchfunktion nach Namen und fortlaufender Nummer innerhalb der Eingabemaske

Automatisches Übernehmen der neuen Absolventen in die DB:

- Generieren einer neuen Erlagscheinnummer
- Übernahme der vorhandenen Daten in die jeweiligen Tabellen

Automatisches Erzeugen neuer Datensätze

Beim Ändern eines Datensatzes wird dieser sofort gespeichert und beim Schließen des Programms wird alles gespeichert.

Großberger: **Abfragen:**

Jahresberichte auf Etappen verschicken:

Es wird ermittelt wer einen Jahresbericht wünscht und noch keinen erhalten hat.

Die Namen der Interessenten werden automatisch im Etikettenformat dargestellt.

Klassentreffenabfragen

Rundschreiben:

Angabe der benötigten Personendaten

Es wird an alle mit aktueller Adresse verschickt

Vor dem Erstellen wird nach 2 Kriterien aussortiert:

PLZ und Land – erst wird das Ausland und dann

Österreich bearbeitet.

Es wird auf Erlagscheine gedruckt

Ecker: Einlesen Absolventendatenbank & Backend

Großberger: Administrator- /Clientstruktur

Ecker: AUTOEXEC

Gemeinsam: Dokumentation



## 11. Ressourcen- & Meilensteinplanung

### 11.1. Ressourcenplanung

**Bianca Ecker:**

- Tabellenstruktur und Einhaltung der Normalformen
- Formularerstellung und Verknüpfung
- Layout
- Suchfunktion nach laufender Nummer
- Zufügen neuer Daten und Übernahme dieser in die entsprechenden Tabellen
- Automatisches hinzufügen der neuen Absolventen
- Programme zum Einlesen der Absolventendaten
- Autoexec
- Dokumentation

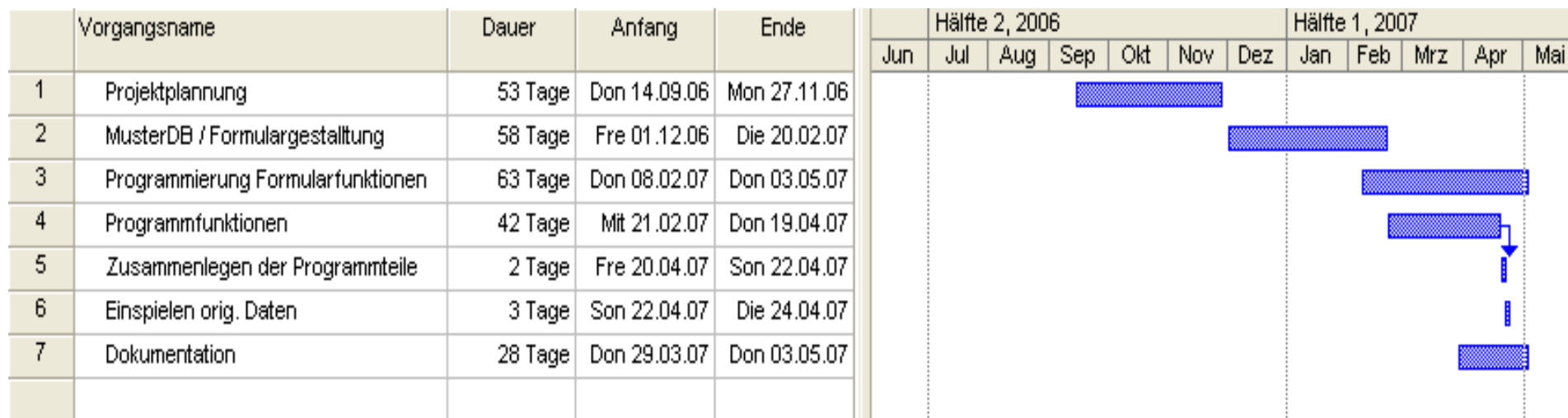
**Michael Großberger:**

- Tabellenstruktur und Einhaltung der Normalformen
- Verschicken der Jahresberichte auf Etappen
- Klassentreffenabfrage
- Etappenweises abschicken der Rundschreiben
- Benutzer
- Dokumentation

**Software:**

- Access
- SQL
- Word
- Visio
- Project

## 11.2. Meilensteinplanung



## 12. Allgemeine Informationen

### 12.1. Datenbanksystem

#### 12.1.1. ALLGEMEIN:

Eine Datenbank besteht aus Dateien, die speziell organisiert aufbewahrt und von einem Datenbanksystem gemeinsam verwaltet werden. Dadurch ermöglichen sie dem Benutzer bestimmte Informationen rasch und zuverlässig aufzufinden.

Zur Nutzung und Verwaltung der in der Datenbank gespeicherten Daten, benötigt man ein Datenbankmanagementsystem. Die Kombination aus Datenbank und Datenbankmanagementsystem bildet das Datenbanksystem. Das DBMS stellt dem Anwender Funktionen zur Verfügung, um Daten eingeben, verändern, abfragen und ausgeben zu können, z.B. MS Access.

Der Anwender greift auf das Datenbanksystem mit einer Datenbankabfragesprache wie z.B. SQL oder einer Anwendung, z.B. www-Seite, zu.

#### 12.1.2. FUNKTIONSWEISE:

Die Struktur einer DB basiert auf dem relationalen Datensystem. Dies ist folgendermaßen aufgebaut. Die Daten werden in Tabellenform gespeichert. Jede Tabelle ist in Spalten (sog. Felder) und Zeilen (entspricht Datensätzen) gegliedert.

Jede Tabelle wird als Relation bezeichnet und hat einen eindeutigen Namen. Jede einzelne Zeile wird als Tupel bezeichnet und die Spaltenüberschriften als Attribute. In jeder Tupel steht genau ein Wert. In jeder Reihe muss ein Feld vorhanden sein, welches die Reihe eindeutig kennzeichnet. Dieses Feld wird als Primärschlüssel bezeichnet. Wird der Primärschlüssel einer anderen Tabelle aufgenommen, so bezeichnet man diesen als Sekundärschlüssel oder Fremdschlüssel.

#### 12.1.3. GUNDSÄTZE DER DATENBANKPLANNUNG:

- Keine Redundanz: Unter Redundanz versteht man das doppelte Vorhandensein einzelner Daten.

- Eindeutigkeit: Jedes Tupel muss durch einen Schlüssel eindeutig identifizierbar sein.
- Keine Prozessdaten: Prozessdaten sind Daten, die durch einen Rechenprozess aus Attributen gewonnen werden.
- In einer Tupel (= am Schnittpunkt von Zeile und Spalte) darf immer nur ein Wert stehen.
- Die Normalformen müssen eingehalten werden.

## 12.2. Normalformen

Das Ziel der Normalformen ist es, die Grundsätze der Datenbankplanung einzuhalten. Dazu gehören das Erstellen von Tabellen und das Herstellen von Beziehungen zwischen diesen Tabellen gemäß Regeln, die dazu dienen, die Daten zu schützen und die Flexibilität der Datenbank zu erhöhen.

### 12.2.1. ERSTE NORMALFORM:

Eine Relation ist in erster NF, wenn alle Attributswerte eindeutig und einelementig sind. Das heißt, zusammengesetzte, mengenwertige oder geschachtelte Wertebereiche sind nicht erlaubt. Nicht eindeutige Attributswerte müssen in mehrere Attribute aufgeteilt werden, z.B. das Attribut Adresse in PLZ, Ort, Strasse.

### 12.2.2. ZWEITE NORMALFORM:

Eine Relation ist in zweiter NF, wenn sie in der ersten NF ist und jedes Nicht-Schlüssel-Attribut vom gesamten Primärschlüssel, nicht nur von einem Teil des Schlüssels, abhängt.

### 12.2.3. DRITTE NORMALFORM:

Die zweite NF muss gelten. Eine Relation ist in der dritten NF, wenn alle Nicht-Schlüssel-Attribute nur vom Schlüssel, nicht aber voneinander, abhängig sind. Z.B die PLZ ist der Primärschlüssel und der Ort das Attribut.

### 12.3. Visual Basic for Application

**VBA** (Visual Basic for Application) ist eine eigenständige, objektorientierte Programmiersprache für die MS Office-Anwendungen. Sie dient der Erweiterung des Funktionsumfangs der MS-Office-Produkte und ist in diese integriert. Ursprünglich hatten die einzelnen Office-Anwendungen eigene, unterschiedliche Makro-Sprachen. Um eine einheitliche Programmierung im Office-Paket zu erreichen, wurde eine einheitliche Programmiersprache für alle Anwendungen entwickelt: VBA.

VBA ist eine objektorientierte, vollwertige Programmiersprache, die sich an Basic anlehnt. Sie verfügt über umfassende Objektmodelle für alle Office-Anwendungen und erlaubt eine strukturierte und modulatorientierte Programmierung. Auch wenn VBA speziell auf die Eigenschaften und Anforderungen für den Einsatz innerhalb einer Office-Anwendung ausgerichtet ist, verfügt es über viele Fähigkeiten und Merkmale einer normalen Programmiersprache, wie beispielsweise Delphi oder Visual Basic. Um die Programmierung zu erleichtern, verfügt das Office-Paket über eine VBA-Entwicklungsumgebung, die als eigenständige Anwendung ausgeführt wird. Die Entwicklungsumgebung bietet eine übersichtliche Arbeitsumgebung zum Programmieren, die darüber hinaus noch einige Programmierhilfen und Arbeitserleichterungen zur Verfügung stellt.

### 12.4. Structured Query Language

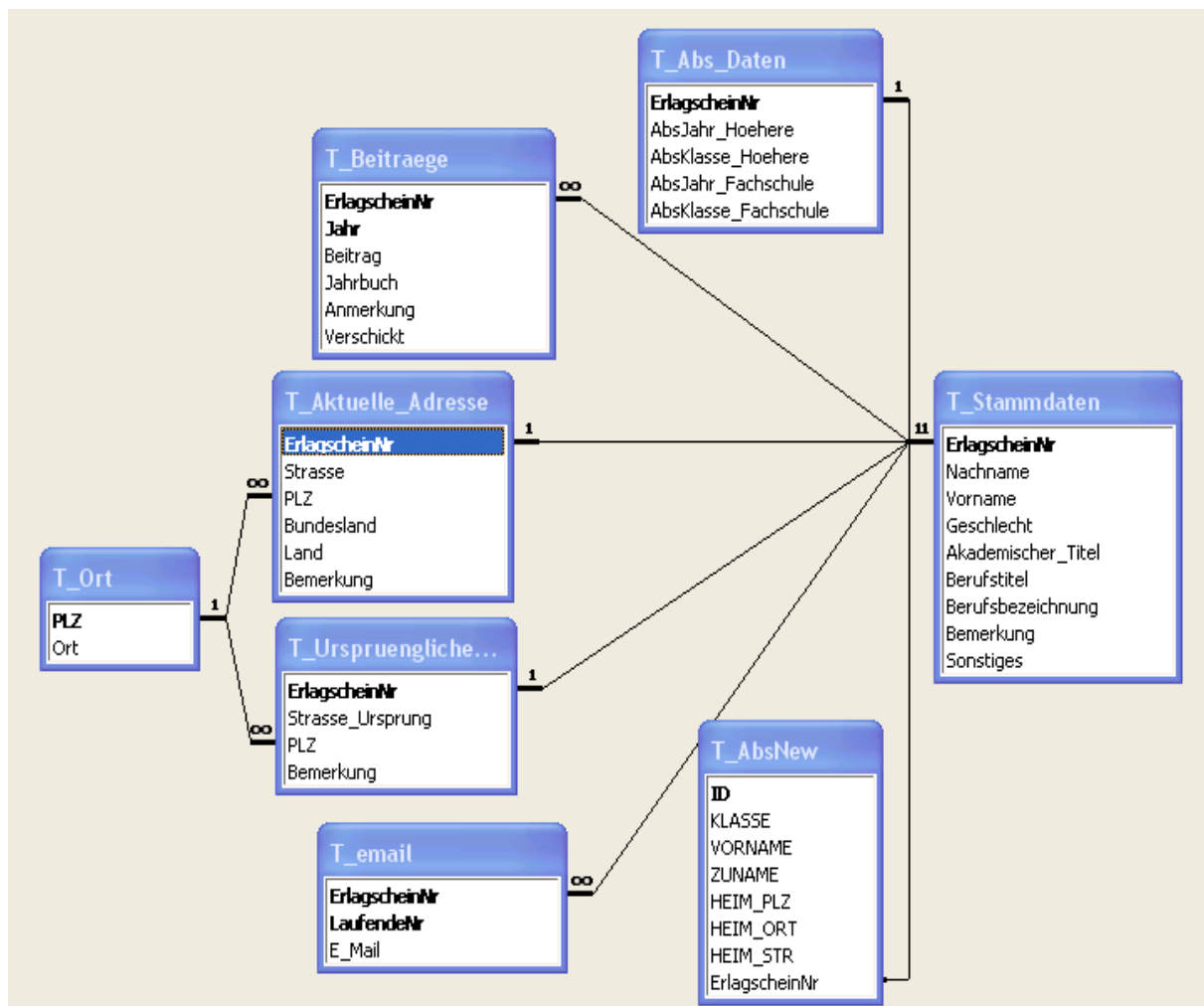
SQL heißt **S**tructured (Strukturiert) **Q**uery (Abfrage) **L**anguage (Sprache). Es ist eine Standardsprache, mit der man auf Datenbanken zugreifen kann. Genauer gesagt kann man

- Datenbanken selbst anlegen, ändern und löschen
- Tabellen erzeugen, ändern und löschen
- Daten hinzufügen, abrufen, ändern und löschen
- Aktionen aufgrund von Änderungen der Datenbank ausgelöst werden

SQL wurde in den 1970ern bei IBM entwickelt. Jede Abfrage und jeder Eintrag, jedes Löschen einer Datenbank geschieht über SQL oder ist zumindest mit SQL auch möglich. Der Sprachumfang von SQL ist gewaltig, einer der Gründe, warum es offensichtlich keine Konkurrenz zu diesem Standard gibt. SQL hat eine relativ einfache Syntax, die an die englische Umgangssprache angelehnt ist. Durch seine Rolle als Quasi-Standard ist SQL von großer Bedeutung, da eine weitgehende Unabhängigkeit von der benutzten Software erzielt werden kann.

## 13. Datenbankstruktur

Wir haben die Absolventendatenbank mit der Struktur einer einzigen, alle Daten umfassenden Tabelle erhalten. Um einen besseren Umgang und die drei Normalformen zu gewährleisten, haben wir die Datenbank grundlegend umstrukturiert. Das heißt wir haben sie auf mehrere Tabellen zerspalten, unnötige Attribute entfernt und andere hinzugefügt, sowie die Schlüssel vergeben.



## 13.1. Tabelle Stammdaten

### 13.1.1. ALLGEMEINES:

Die Tabelle Stammdaten ist die zentrale Tabelle im relationalen Datenbankaufbau. Sie stellt die Schnittstelle zwischen sämtlichen anderen Personendaten enthaltenden Tabellen dar. Sie muss im Gegensatz zu anderen Tabellen alle Absolventen enthalten.

### 13.1.2. ELEMENTE:

<b>ErlagscheinNr</b>	Die Erlagscheinnummer ist der Primärschlüssel und damit das Herzstück der Tabelle. Information über ihren Aufbau finden Sie unter Punkt 14.
<b>Nachname</b>	Dieses Attribut muss immer enthalten sein. Änderungen sind jedoch jeder Zeit möglich.
<b>Vorname</b>	Muss ebenfalls enthalten sein und ist änderbar.
<b>Geschlecht</b>	Das Geschlecht scheint in der Tabelle auf, um bei den Abfragen die Anreden Frau oder Herr beifügen zu können.
<b>Akademischer_Titel</b>	Kann eingetragen werden. Bei HTL-Absolventen wird bei der Neuabsolventenübernahme automatisch „Ing.“ hinzugefügt.
<b>Berufstitel</b>	Kann enthalten sein.
<b>Berufsbezeichnung</b>	Kann ebenfalls enthalten sein.
<b>Bemerkung</b>	In der Tabelle Stammdaten werden nur entscheidende Informationen zur Person im Feld Bemerkung eingetragen, z.B. ein Todesfall.
<b>Sonstiges</b>	Hier hat der Administrator die Möglichkeit allfällige, für ihr wichtige, Daten einzutragen.

## 13.2. Tabelle Absolventendaten

### 13.2.1. ALLGEMEINES:

Hie werden das Abschlussjahr und die Abschlussklasse erfasst. Nachdem es möglich ist nach einem Fachschulabschluss die HTL zu absolvieren, ist ein Mehrfacheintrag möglich.

### 13.2.2. ELEMENTE:

<b>ErlagscheinNr</b>	Als Unsiversalprimärschlüssel
<b>AbsJahr_Hoehere</b>	Hier wird das Abschlussjahr der HTL eingetragen.
<b>AbsKlasse_Hoehere</b>	Abschlussklassenbezeichnung und die Klasse (a, b) der HTL vermerkt.
<b>AbsJahr_Fachschule</b>	Hier wird das Abschlussjahr der Fachschule eingetragen.
<b>AbsKlasse_Fachschule</b>	Abschlussklassenbezeichnung und die Klasse (a, b) der Fachschule vermerkt.

## 13.3. Tabelle Beiträge

### 13.3.1. ALLGEMEINES:

In der Tabelle Beiträge werden alle erhaltenen Spenden an den Absolventenverein festgehalten. Ebenfalls wird der Wunsch des Absolventen nach dem Erhalt eines aktuellen Jahrbuches vermerkt.

### 13.3.2. ELEMENTE:

<b>ErlagscheinNr</b>	Als Primärschlüssel
<b>Jahr</b>	Das Jahr stellt den Sekundärschlüssel dar. Es werden Einträge und damit Jahre in die DB aufgenommen, in denen der Absolvent an den Absolventenverein gespendet hat.



<b>Beitrag</b>	Hier wird die Höhe der Spende vermerkt. Aus ihr geht in der Regel auch hervor, ob der Absolvent ein Jahrbuch wünscht.
<b>Jahrbuch</b>	Wird als Kontrollkästchen angezeigt, weil nicht jeder Spender auch ein Jahrbuch wünscht.
<b>Anmerkung</b>	Hier hat der Administrator die Möglichkeit eventuelle Zusatzinformationen zu vermerken.
<b>Verschickt</b>	Dieses Attribut dient zur Administration des Ausschickens der Jahrbücher. Da der Absolventenverein die Erlagscheine nur allmählich zurückerhält, müssen die Jahrbücher auf Etagen ausgeschickt werden. Um hier über eine Abfrage den Überblick zu behalten, ist das Attribut Verschickt nötig.

## 13.4. Tabelle Aktuelle Adresse

### 13.4.1. ALLGEMEINES:

Diese Tabelle zeigt den zuletzt erfassten Wohnort des Absolventen auf. Das heißt, dass sie bei Schulabgängern die Heimatadresse zum Zeitpunkt des Schulabschlusses enthält und durch Zahlung des Absolventenmitgliedsbeitrags eine Aktualisierung der Adresse möglich ist.

### 13.4.2. ELEMENTE:

<b>ErlagscheinNr</b>	Als Primärschlüssel
<b>Strasse</b>	Hier werden die Strassenbezeichnung und die Hausnummer angegeben.
<b>PLZ</b>	Postleitzahl der zuletzt bekannten Wohnadresse
<b>Bundesland</b>	Die Angabe des Bundeslandes ist auslandsbezogen.
<b>Land</b>	Das Land wird ebenfalls nur im Falle einer bekannten Auswanderung angegeben.
<b>Bemerkung</b>	Im Feld Bemerkung gibt der Administrator sonstige Informationen wie z.B. „verzogen“ an.

## 13.5. Tabelle ursprüngliche Adresse

### 13.5.1. ALLGEMEINES:

In diese Tabelle wird die Wohnadresse zum Zeitpunkt des Schulabschlusses erfasst. Diese Adresse wird nicht mehr verändert, da sie für die spätere Ausmachung des Absolventen für ein Klassentreffen von Interesse sein kann.

### 13.5.2. ELEMENTE:

<b>ErlagscheinNr</b>	Primärschlüssel
<b>Strasse_Ursprung</b>	Die Heimatstrasse des Absolventen
<b>PLZ</b>	Postleitzahl des ursprünglichen Wohnorts
<b>Bemerkung</b>	Dieses Feld kann wieder den Wortlaut „verzogen“ beinhalten, falls eine neue Wohnadresse nicht bekannt ist.

## 13.6. Tabelle Ort

### 13.6.1. ALLGEMEINES:

Die Tabelle Ort besteht, um die Normalformen zu gewährleisten. Sie steht mit den Tabellen aktuelle Adresse und ursprüngliche Adresse in Beziehung. Durch die in diesen Tabellen enthaltenen Postleitzahlangaben kann der dazugehörige Ortsname automatisch ergänzt werden.

### 13.6.2. ELEMENTE:

<b>PLZ</b>	Ist der Primärschlüssel der Tabelle und enthält die Postleitzahl. Sie ist als Text definiert, weil die kanadischen Postleitzahlen Buchstaben beinhalten.
<b>Ort</b>	Enthält den dazugehörigen Ortsnamen

## 13.7. Tabelle e-mail

### 13.7.1. ALLGEMEINES:

Diese Tabelle wird neu in die Datenbank eingegliedert, da in Zukunft alle Absolventen eine e-mail-Adresse hinterlassen sollen und bald auch elektronische Rundschreiben verschickt werden. Bei der e-mail-Adresse ist eine Mehrfachangabe möglich.

### 13.7.2. ELEMENTE:

<b>ErlagscheinNr</b>	Primärschlüssel
<b>LaufendeNr</b>	Ist der Sekundärschlüssel. Er kennzeichnet Zweit- und Drittadressen eines Absolventen.
<b>E_Mail</b>	Diese Feld beinhaltet die Adresse

## 13.8. Tabelle neue Absolventen

### 13.8.1. ALLGEMEINES:

Diese Tabelle ist eine temporäre Tabelle. Das heißt sie wird durch das Drücken der Schaltfläche „Absolventen hinzufügen“ geschrieben. Es werden die Daten der letzten Absolventen vom Sekretariat übernommen. Aus ihnen, den in der Absolventendatenbank bestehenden Daten (fortlaufende Nummer) und aus dem Betriebssystemdaten (Jahr) werden die neuen Erlagscheinnummern errechnet und in der Tabelle angefügt. Anschließend werden Die Daten der Tabelle neue Absolventen den Stammtabellen hinzugefügt. Danach wird der Tabelleninhalt wieder gelöscht.

Alle Felder der Tabelle müssen enthalten sein!

### **13.8.2. ELEMENTE:**

<b>ID</b>	Da die Tabelle mit dem Feld ID als Primärschlüssel von Sekretariat übernommen wird, bleibt ID auch der Primärschlüssel.
<b>KLASSE</b>	Beinhaltet die volle Klassenbezeichnung mit Jahrgang, Klasse und Abteilungsbezeichnung.
<b>VORNAME</b>	
<b>ZUNAME</b>	
<b>HEIM_PLZ</b>	Die Postleitzahl der beim Abschluss gemeldeten Adresse.
<b>HEIM_ORT</b>	Die Ortsbezeichnung der beim Abschluss gemeldeten Adresse.
<b>HEIM_STR</b>	Die Strasse und Hausnummer der beim Abschluss gemeldeten Adresse.
<b>ErlagscheinNr</b>	Neue zusammengestellte Erlagscheinnummern bestehend aus Abteilung, Jahrgang, Klasse und neuer laufender Nummer.

## **13.9. Tabelle laufende Nummer**

### **13.9.1. ALLGEMEINES:**

Die Tabelle laufende Nummer ist ebenfalls temporär. Jedoch wird sie durch das Betätigen der Schaltfläche „Absolventen hinzufügen“ erst erzeugt, im Laufe des Rechenprozesses geschrieben und nach Abschluss des Schreibens der neuen Absolventendaten in die Stammtabellen wieder gelöscht. Sie wird rein dazu benötigt, um die bereits in der Absolventendatenbank bestehenden, laufenden Nummern in die Tabelle zu schreiben und durch die Abfrage „laufende Nummer“ absteigend zu sortieren. Anschließend kann die höchste bereits existierende laufende Nummer aus der Tabelle laufende Nummer ausgelesen, erhöht und einem neuen Absolventen zugewiesen werden.

### 13.9.2. ELEMENTE:

#### **lfdNr**

Diese Tabelle benötigt nur ein Attribut, da sie rein dazu dient die bestehenden laufenden Nummern der Absolventendatenbank zu sortieren. Das Attribut lfdNr ist kein Primärschlüssel, da durch fehlende Abfragen oder Rechenprozesse keiner benötigt wird.

## 14. Erlagscheinnummer

### 14.1. Aufbau der Erlagscheinnummer

Die Erlagscheinnummer in fast allen Tabellen als Primärschlüssel eingesetzt. Sie ist eindeutig und besteht aus 12 Ziffern. Diese sind wie folgt logisch gegliedert.

11	1984	1	12345
Abteilung	Jahrgang	Klasse	lfd. Nummer

#### 14.1.1. ABTEILUNG:

Die verschiedenen Abteilungen sind durch Ziffern gekennzeichnet. Dabei ist natürlich zu beachten, dass frühere Abteilungsbezeichnungen auch in die Kennzeichnungsliste mit aufzunehmen sind.

**ACHTUNG:** Wenn die Klassenbezeichnungen erneut verändert werden, müssen auch neue Kennziffern in die Liste und den Funktionscode für das Importieren neuer Absolventen aufgenommen werden.

11 = MF  
12 = M  
13 = A  
14 = EF  
15 = E  
16 = B  
17 = W

23 = HMIA  
24 =  
25 =  
26 = HETI  
27 = HETR  
28 = HWII  
29 = HWIM

21 = FET  
22 = FMBF

99 = keine Abteilung oder Abteilung  
unbekannt

#### 14.1.2. JAHRGANG:

Der Jahrgang steht für das Abschlussjahr. Für das automatische Generieren der Erlagscheinnummer beim Importieren neuer Absolventen kann das Jahr aus dem Betriebssystem gelesen werden.

### 14.1.3. KLASSE:

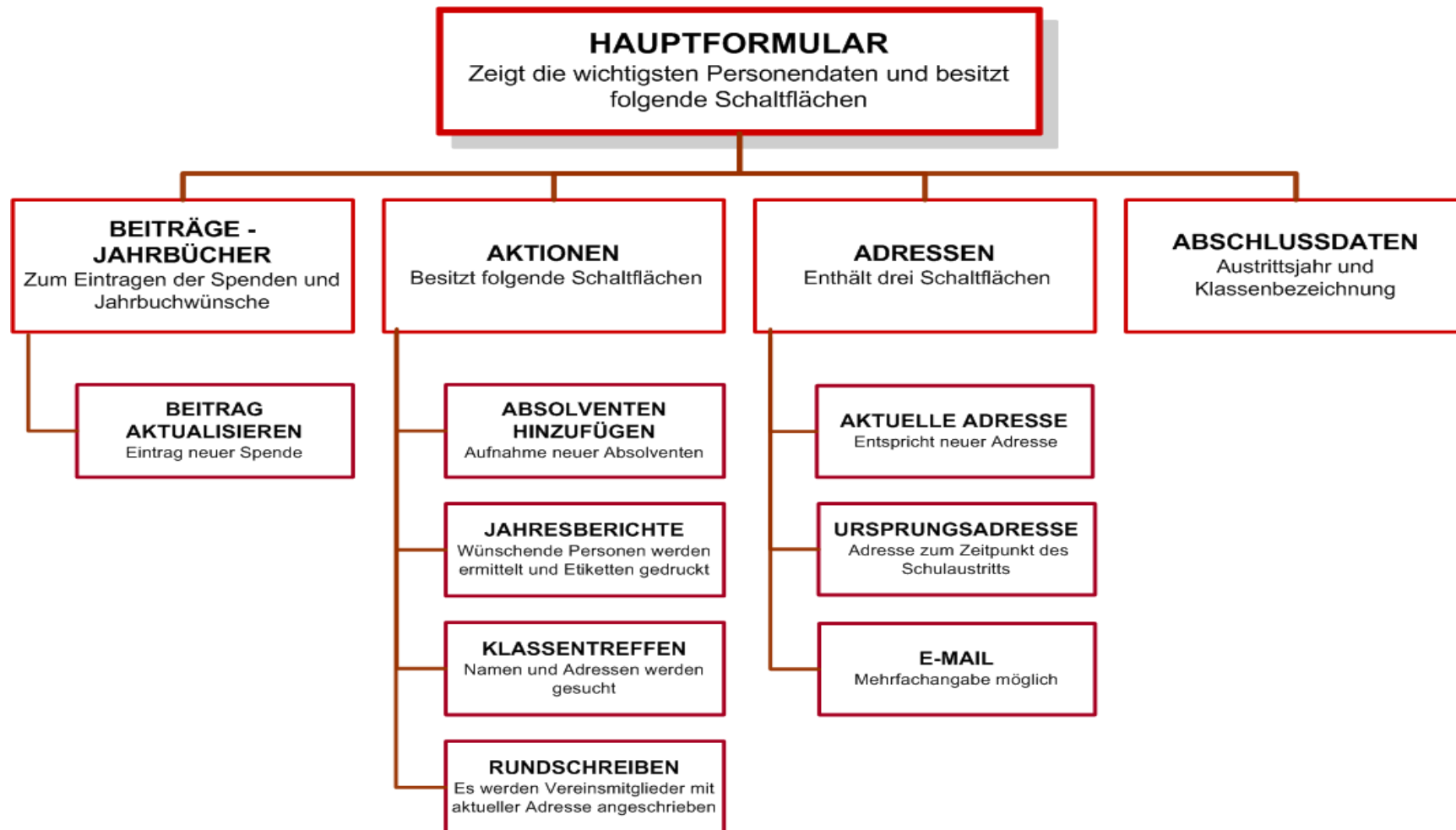
Gibt an, ob es sich um die Klasse a (=1) oder b (=2) des Jahrganges und der Abteilung handelt.

**ACHTUNG:** Wenn sich Klassenbezeichnungen ändern muss die Importfunktion umgeschrieben werden.

### 14.1.4. LAUFENDE NUMMER:

Die laufende Nummer ist eindeutig. Das heißt, dass jeder Absolvent eine neue lfd. Nr. erhält (fortlaufend). Sie wurde in der alten DB als eigenes Attribut geführt. Dies ist in dieser DB nicht mehr so, weil dies zu Redundanzen führen würde (siehe Normalformen).

## 15. Formulare und ihre Funktionen





## 15.1. Grundlegendes

### 15.1.1. ALLGEMEIN:

Für unseren Auftraggeber Herrn VL Schneider war eine Umstrukturierung der Formulare ein sehr wichtiger Punkt bei der Gestaltung unseres Projektes. Da für ihn die Arbeit mit seiner alten Maske sehr umständlich war, haben wir versucht, die Formulare so zu strukturieren, dass sie einerseits übersichtlich, klar und ansprechend für das Auge sind, andererseits sollten sie sich aber natürlich seinen Arbeitsgewohnheiten anpassen. Deshalb kommt es bei den Formularen auch zu Mehrfachnennungen der Felder. So haben wir z.B. die am häufigsten benötigten Daten am Hauptformular untergebracht. Diese kommen aber in anderen Formularen auch wieder vor, da sie dort thematisch zu Hause sind.

### 15.1.2. ARBEITEN MIT DEN FORMULAREN:

Da die Arbeitswege immer wieder über das Hauptformular laufen, ist diese als Suchformular vorgesehen. Die Daten des gesuchten Absolventen werden beim Öffnen eines anderen Formulars automatisch auf diesem angezeigt. Lediglich das Formular für die Beiträge und Jahrbücher besitzt ebenfalls eine Suchfunktion. Da man von diesem aus keinen Zugriff auf andere Formulare hat, beschränkt sich die Suche auf dieses Formular.

Damit es zu keinen Verwechslungen kommt, ist es also wichtig immer die eingegebene Nummer auf dem angezeigten Formular zu kontrollieren. Ein sicheres, übersichtliches und fehlerfreies Arbeiten ist dann ganz sicher möglich, wenn alle Zusatzformulare gleich nach Gebrauch wieder geschlossen werden!

Man gelangt von Hauptformular ausgehend in die anderen Formulare indem man die jeweilige Schaltfläche anklickt. Dies ist auch mit der Enter- oder Tabulatortaste möglich. In jedem Formular springt man durch das Betätigen der Taste in das nächste Feld. Nach dem letzten Feld springt der Cursor auf die erste Schaltfläche. Drückt man auf Enter, so öffnet sich das gewählte Formular.

## 15.2. Das Hauptformular

## Hauptformular

laufende Nr	01676
-------------	-------

ErlagscheinNr	111929001676
---------------	--------------

Nachname	Kneidinger
----------	------------

Vorname	Johann
---------	--------

PLZ 3340

Ort	Waidhofen/Ybbs
-----	----------------

<b>Strasse</b>	Ybbsitzerstrasse 55
----------------	---------------------

Akademischer Titel Ing

<b>Berufstitel</b>	Kommerzialrat
--------------------	---------------

**Berufsbezeichnung** Landmaschinen,Kfz

<b>Bemerkung</b>	verstorben 17.10.89
------------------	---------------------

Sonstiges	
-----------	--

Beiträge -  
Jahrbücher

## Adressen

## Abschlussdaten

## Aktionen

## F\_Beitraege

	Jahr	Beitrag	Jahrbuch	Verschickt	Anmerkung
▶	2003	450,00 €	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Datensatz:   1   von 1

### **15.2.1. ALLGEMEIN:**

Wie man am Organigramm auf einen Blick erkennt, gehen alle Formulare vom Hauptformular aus. Dies ist deswegen so aufgebaut, weil Herr Schneider hauptsächlich mit den Daten, die auf dem Hauptformular aufgeführt sind, arbeitet. Hier sind also alle Daten ersichtlich, die am häufigsten benötigt werden. Das Formular ist aber nicht nur Übersicht, sondern auch das Suchformular. Hier können über die laufende Nummer alle Absolventen ermittelt werden.

Weiters kann man über die rechts angeführten Schaltflächen alle anderen Formulare und die Funktionen, die die neue Datenbank bietet, erreichen.

### **15.2.2. FELDER UND SCHALTFLÄCHEN:**

#### **15.2.2.1. laufende Nummer:**

Dieses Feld wurde eigens angelegt, um die Arbeit mit dem Programm zu erleichtern und zu rationalisieren. Da die Suche nach einem Absolventen über die Erlagscheinnummer läuft, diese aber sehr lange ist, wurde das Feld laufende Nummer eingeführt. Man gibt also nur die letzten fünf Ziffern der Erlagscheinnummer ein, muss anschließend auf Enter drücken und alle Daten des gewünschten Absolventen erscheinen auf jedem Formular, das neu geöffnet wird.

Wird die laufende Nummer falsch eingegeben (mehr oder weniger als fünf Ziffern) oder gibt es die Zahlenkombination noch nicht, so wird der erste in der Datenbank stehende Absolvent angezeigt. Drückt man Enter ohne etwas in das Feld laufende Nummer einzutragen so erscheint die Fehlermeldung: „Erlagscheinnummer nicht vorhanden!“.

```
strLfdnr = Forms!F_Hauptformular!lfdnr  
stamm.FindFirst "[ErlagscheinNr] Like '*' & strLfdnr & ''"
```

#### **15.2.2.2. Erlagscheinnummer:**

Da die Suche nach einem Absolventen künftig über die laufende Nummer erfolgt, dient das Feld Erlagscheinnummer lediglich der Kontrolle. Wenn das Ende der angezeigten Erlagscheinnummer (also die fortlaufende Nummer) nicht mit der Ziffernkombination übereinstimmt, die im Feld laufende Nummer eingegeben wurde, so ist von einem Fehler bei der Eingabe der laufenden Nummer zu rechnen. Das heißt, dass diese entweder noch nicht vorhanden ist oder falsch Eingegeben wurde.

### **15.2.2.3. Nachname & Vorname:**

Die Felder Nachname und Vorname müssen immer ausgefüllt sein. Bei Namensänderung ist es möglich diese hier einzutragen. Die veränderten Daten werden in die Tabelle Stammdaten übernommen.

### **15.2.2.4. Adressdaten:**

Es ist möglich die Adresse eines Absolventen über die Eingabemaske im Hauptformular zu ändern. Uneingeschränkt änderbar sind die Daten des Feldes Strasse, in dem die Straßenbezeichnung, Hausnummer und eventuell die Türnummer stehen.

Das Feld Postleitzahl ist ebenfalls änderbar. Wenn eine PLZ eingetragen wird, kontrolliert das Programm, ob diese bereits in der Tabelle Ort vorhanden ist. Wenn ja, wird automatisch der in der Tabelle Ort abgespeicherte Wortlaut der Ortsbezeichnung übernommen. Wenn die PLZ noch nicht in der Tabelle Ort angelegt ist, so wird sie hinzugefügt. Die Ortsbezeichnung, die in das Feld Ort eingetragen wird, wird der neuen PLZ angefügt.

Wenn von einem Absolventen keine Adresse mehr bekannt ist, so in der Tabelle aktuelle Adresse das Feld Strasse leer bleiben. Bei der Postleitzahl wird 0 eingetragen, und damit ergänzt sich im Feld Ort automatisch die Bezeichnung „Unbekannt“.

### **15.2.2.5. Titel, Bemerkung & Sonstiges:**

All diese Felder können, aber müssen nicht ausgefüllt sein. Sie sind in der Maske frei veränderbar. Sie werden in der Tabelle Stammdaten aktualisiert sobald auf Enter gedrückt oder wo anders angeklickt wird.

Im Feld akademischer Titel wird bei allen Absolventen der HTL bei der Datenübernahme automatisch „Ing.“ zugefügt. Natürlich können später auch andere Titel ergänzt werden. Die Felder Berufstitel und Berufsbezeichnung sprechen für sich. Im Feld Bemerkung werden nur wesentliche Stichworte wie verzogen oder verstorben vermerkt. Das Feld Sonstiges lässt Notizen aller Art zu.

#### **15.2.2.6. Unterformular Beiträge:**

Dieses Formular zeigt, wann und in welcher Höhe der abgefragte Absolvent an den Absolventenverein gespendet hat, ob er einen Jahresbericht wünschte und dieser auch verschickt wurde. Das Unterformular dient rein zur Übersicht. Da alte Datenstände bestehen bleiben sollen, können die Daten in dem Fenster nicht verändert werden.

#### **15.2.2.7. Schaltfläche Beiträge - Jahrbücher:**

Durch einfaches Klicken gelangt man in das Formular Beiträge – Jahrbücher. Man kann das Formular auch öffnen indem man mit der Enter- oder Tabulatortaste alle Felder durchspringt bis man die gewünschte Schaltfläche erreicht. Wenn man nun auf Enter drückt, öffnet sich das gewünschte Formular.

Das Formular Beiträge – Jahrbücher dient dazu, die neuen Spenden der Absolventen einzutragen. Da häufig nur Erlagscheine bearbeitet werden, verfügt dieses Formular über eine Suchfunktion und alle für die Kontrolle und Eingabe benötigten Daten.

#### **15.2.2.8. Schaltfläche Adressen:**

Wenn man die Schaltfläche Adressen betätigt, öffnet sich kein Arbeitsformular sonder ein Zwischenformular bei dem man drei Schaltflächen zur Auswahl hat. Es wurden drei Schaltflächen angelegt, um die einzelnen Adresssorten klar auseinander zu halten. Man hat die Wahl zwischen der „aktuellen Adresse“, der „Ursprungsadresse“ und der „e-mail“-Adresse.

Im Formular aktuelle Adresse befindet sich die zuletzt bekannte Adresse des Absolventen. Das Formular Ursprungsadresse beinhaltet die Adressdaten, die der Absolvent zum Zeitpunkt seines Schulabschlusses hatte. In Zukunft sehr wichtig werden die e-mail-Adressdaten, da ab 2007 jeder Absolvent beim Abschluss eine e-mail-Adresse bekannt geben soll und der Absolventenverein diese zukünftig vermehrt nützen wird.

#### 15.2.2.9. Schaltfläche Abschlussdaten:

Wenn diese Schaltfläche betätigt wird, öffnet man das Formular Abschlussdaten. In ihm sind das Jahr und Klassenbezeichnung der Austrittsklasse/n enthalten.

```
Private Sub Abschlussdaten_Click()  
    DoCmd.OpenForm "F_Abschlussdaten"  
End Sub
```

#### 15.2.2.10. Schaltfläche Aktionen:

Hier gelangt man zum Formular Aktionen, einem wesentlichen Kernstück unserer Diplomarbeit. Von den vier Schaltflächen dieses Formulars aus können sämtliche **Funktionen** gestartet werden, die nicht direkt in einem Verwaltungsformular zur Anwendung kommen.

Diese Funktionen sind:

- Absolventen hinzufügen: Zur Aufnahme der neuen Absolventen in die Datenbank
- Jahresberichte: Über diese Funktion werden die Absolventen, die spendeten, einen Jahresbericht wünschen und noch keinen erhalten haben ermittelt, und anschließend ihre Namen und Adressdaten im Etikettenformat gedruckt. So ist das Ausschicken auf Etappen möglich.
- Klassentreffen: Zum Suchen von Klassenmitgliedern für Klassentreffen
- Rundschreiben: Es werden nur die Vereinsmitglieder mit aktueller Adresse angeschrieben. Sie erhalten einen Brief inklusive Erlagschein.

## 15.3. Beiträge - Jahrbücher

### Beiträge - Jahrbücher

laufende Nr

ErlagscheinNr

Nachname

Vorname

PLZ

Ort

Strasse

Bundesland

Land

Jahr

Beitrag

Jahrbuch

Anmerkung

**Beitrag aktualisieren**

#### F\_Unterformular\_Beitraege

2005	12,00 €	Ja	Ja	nicht extra JB 2006
2004	4,50 €	Nein	Nein	nicht extra JB 2006
2002	4,50 €	Nein	Nein	nicht extra JB 2006
2000	60,00 €	Nein	Nein	nicht extra JB 2006
1999	100,00 €	Nein	Nein	nicht extra JB 2006
1998	200,00 €	Ja	Ja	nicht extra JB 2006

### 15.3.1. ALLGEMEIN:

Das Formular Beitrag – Jahrbücher dient dem Hinzufügen eines neuen Spendenbeitrags. Dieses Formular wird über eine Schaltfläche im Hauptformular erreicht. Die eingetragenen Spenden werden immer dem aktuellen Jahr zugewiesen.

Das Formular Beiträge – Jahrbücher besitzt eine Suchfunktion. Durch diese hat der Benützer die Möglichkeit beim Durchsehen der Erlagscheine nur mit diesem Formular zu arbeiten.

### 15.3.2. FELDER, SCHALTFLÄCHE UND UNTERFORMULAR:

#### 15.3.2.1. laufende Nummer:

Dieses Formular besitzt das Feld laufende Nummer, da es hier ebenfalls möglich ist, einen Datensatz über die letzten fünf Stellen der Erlagscheinnummer zu suchen. Diese Funktion wurde im Formular Beiträge – Jahrbücher ergänzt, um dem Administrator seine Arbeit zu erleichtern. Natürlich besitzt es auch die Funktion, dass die im Hauptformular stehende lfdNr. beim Öffnen des Formulars Beiträge – Jahrbücher automatisch abgefragt wird.

```
Forms!F_BTJB!lfdnr = Forms!F_Hauptformular!lfdnr
strLfdnr = Forms!F_BTJB!lfdnr
stamm.FindFirst "[ErlagscheinNr] Like '*' & strLfdnr & '"

On Error Resume Next
If strLfdnr = True Then
    Forms!F_BTJB!ErlagscheinNr = stamm!ErlagscheinNr
    Forms!F_BTJB!Nachname = stamm!Nachname
    Forms!F_BTJB!Vorname = stamm!Vorname
Else:
End If

aktadr.FindFirst "[ErlagscheinNr] Like '*'&strLfdnr&'"

If strLfdnr = True Then
    Forms!F_BTJB!PLZ = aktadr!PLZ
    If IsNull(aktadr!Strasse) Then
        Forms!F_BTJB!Strasse = " "
    End If
    Forms!F_BTJB!Strasse = aktadr!Strasse
    Forms!F_BTJB!Bundesland = aktadr!Bundesland
    Forms!F_BTJB!Land = aktadr!Land
Else: MsgBox "Fehler bei Schleife aktadr!"
End If
```



```
strPLZ = aktadr!PLZ
Ort.FindFirst "[PLZ] = '" & strPLZ & "'"

If strPLZ = True Then
    If IsNull(Ort!Ort) Then
        Forms!F_BTJB!Ort = " "
    End If
    Forms!F_BTJB!Ort = Ort!Ort
Else: Forms!F_BTJB!Ort = "Unbekannt"
End If
```

### **15.3.2.2. Erlagscheinnummer:**

Dieses Feld dient der Kontrolle. Es sollen die letzten fünf Stellen der Erlagscheinnummer mit der darüber stehenden laufenden Nummer verglichen werden. Stimmen die Zahlen nicht überein, so wurde ein Fehler bei der Eingabe der laufenden Nummer gemacht. Das heißt, dass die laufende Nummer weniger oder mehr als fünf Stellen aufweist, oder dass die eingegebene Nummer noch nicht vergeben wurde.

### **15.3.2.3. Nachname & Vorname:**

Die Namensfelder sind ein wichtiger Bestandteil des Formulars, der es ermöglicht beim Eintragen der Spenden nur mit diesem Formular zu arbeiten. Der Name dient natürlich der Kontrolle. Er kann bei Namensänderungen aber auch im Formular verändert werden.

### **15.3.2.4. Adressdaten:**

Beim Durchsehen der Erlagscheine soll auch immer ein Blick auf die Adresse fallen. Wenn ein Vereinsmitglied eine andere Anschrift besitzt, als in der Datenbank angegeben, so kann die Adresse im Formular einfach geändert werden. Die Daten werden automatisch den richtigen Tabellen zugeführt oder in ihnen ergänzt. Die alte Adresse wird überschrieben, sie ist also nicht mehr verfügbar.

### 15.3.2.5. Beitragsdaten:

Das Jahr wird bereits beim Öffnen und bei jeder Aktualisierung eingetragen. Es entspricht immer dem aktuellen Kalenderjahr. Es soll auf keinen Fall verändert werden. Auch wenn ein Absolvent erst Monate später etwas Spendet und so bereits das nächste Kalenderjahr begonnen hat, ist das für die weitere Abwicklung kein Problem. Beim Ausschicken der Jahrbücher wird es deshalb nicht zu Fehlern kommen.

Da es möglich ist, dass ein Mitglied des Absolventenvereins das Jahrbuch des Vorjahres im Frühjahr bestellt, und im Herbst des selben Jahres den aktuellen Jahresbericht wünscht, muss es möglich sein das Jahr zu verstellen. Dies wird durch die Schaltflächen Plus und Minus erreicht. Drückt man auf Plus, so erhöht sich das Jahr um eins, auf Minus wird es um eins niedriger. So gibt nicht das Spendenjahr an sondern das Erscheinungsjahr des Jahrbuchs.

Der gespendete Beitrag wird in der Währung Euro übernommen. Es muss lediglich eine Zahl eingegeben werden. Das Komma und die Währung werden automatisch ergänzt. Der Spendenbeitrag kann beliebig hoch oder niedrig sein.

Das Feld Jahrbuch wird gekennzeichnet, wenn der Absolvent das Zusenden eines Jahrbuchs will. Es wird gekennzeichnet indem man es anklickt. Wenn durch die Tabulatortaste oder die Entertaste vorgerückt wurde, kann das Feld durch die Leertaste auf true gesetzt werden.

Das Feld Anmerkung erlaubt kurze Kommentare.

### 15.3.2.6. Schaltfläche Beitrag aktualisieren:

Die in den Beitragsdatenfeldern neu angegebenen Daten werden erst in die Datenbank übernommen, wenn die Schaltfläche „Beitrag aktualisieren“ betätigt wurde. Bis dort hin können die Daten noch beliebig verändert werden. Wird die Maske geschlossen ohne diese Schaltfläche betätigt zu haben, so gehen die Informationen verloren.

```
strLfdnr = Forms!F_BTJB!lfdnr  
btjb.FindFirst "[ErlagscheinNr] Like '*' & strLfdnr & '"  
  
If strLfdnr = True Then  
    If Forms!F_BTJB!Jahr = (year(Date)) Then  
        btjb.MoveLast  
        btjb.AddNew
```

```
Forms!F_BTJB!Jahr = (year(Date))
btjb!ErlagscheinNr = Forms!F_BTJB!ErlagscheinNr
btjb!Jahr = Forms!F_BTJB!Jahr

btjb.Update
Else: MsgBox "ois fian oasch!"
End If

Else: MsgBox "Fehler beim reinschreiben der Beiträge!"
End If

aktuelles_Jahr = Forms!F_BTJB!Jahr
btjb.FindFirst "[ErlagscheinNr] Like '*' & strLfndr & '' And [Jahr] like ''
& aktuelles_Jahr & '' "
    btjb.Edit

    btjb!Beitrag = Forms!F_BTJB!Beitrag
    btjb!Jahrbuch = Forms!F_BTJB!Jahrbuch
    btjb!Verschickt = False
    btjb!Anmerkung = Forms!F_BTJB!Anmerkung

    btjb.Update

Forms!F_BTJB!Jahr = " "
Forms!F_BTJB!Beitrag = " "
Forms!F_BTJB!Jahrbuch = False
Forms!F_BTJB!Anmerkung = " "
```

### 15.3.2.7. Unterformular alte Beiträge:

Dieses Formular zeigt, wann und in welcher Höhe der abgefragte Absolvent an den Absolventenverein gespendet hat, ob er einen Jahresbericht wünschte und dieser auch verschickt wurde. Das Unterformular dient rein zur Übersicht. Da alte Datenstände bestehen bleiben sollen, können die Daten in dem Fenster nicht verändert werden. Das Datenfenster dient rein der Information.

## 15.4. Adressen

The image shows a software interface for managing addresses. At the top is a title bar with the text 'Adressen' in white on a dark blue background, and a small square button with a white 'x' on the right. Below the title bar, there are three light blue buttons with black text, arranged vertically. The first button is labeled 'aktuelle Adresse' and has a dashed border. The second button is labeled 'Ursprungsadresse'. The third button is labeled 'e-mail'.

### 15.4.1. ALLGEMEIN:

Dieses Formular dient zur Übersicht. Über die drei Schaltflächen kann man auf alle unterschiedlichen Formulare, die Adressdaten beinhalten, zugreifen.

### 15.5. aktuelle Adresse

**aktuelle Adresse**

x

<b>ErlagscheinNr</b>	161980002494
<b>PLZ</b>	98201
<b>Ort</b>	Everett
<b>Strasse</b>	2130 Cedar Street
<b>Bundesland</b>	Washington
<b>Land</b>	USA
<b>Bemerkung</b>	

### 15.5.1. ALLGEMEIN:

Das Formular aktuelle Adresse ist ein Anzeigeformular. Das heißt, dass es keine Suchfunktion besitzt. Der Anwender kann hier die aktuelle Adresse anzeigen lassen oder diese verändern. Zum gewünschten Datensatz gelangt man, indem man im Hauptformular die entsprechende laufende Nummer angibt und anschließend über das Formular Adressen in das Formular aktuelle Adresse geht. Man benötigt dieses Formular, um Auslandsadressen anzugeben.

### 15.5.2. FELDER:

Die Erlagscheinnummer dient lediglich der Kontrolle. Alle anderen Felder zeigen die Adressdaten des Absolventen. Die Felder PLZ, Ort und Strasse sind wie gewohnt zu handhaben. Die Felder Bundesland und Land sind jedoch nur für Auslandsadressen zu verwenden. Im Feld Bemerkung kann der Anwender Kommentare wie z.B. „verzogen“ eingeben. In diesem Fall wird der betroffene Absolvent beim Drucken von Etiketten oder Rundschreiben ausgelassen.

Das Feld Postleitzahl ist änderbar. Wenn eine PLZ eingetragen wird, kontrolliert das Programm, ob diese bereits in der Tabelle Ort vorhanden ist. Wenn ja, wird automatisch der in der Tabelle Ort abgespeicherte Wortlaut der Ortsbezeichnung übernommen. Wenn die PLZ noch nicht in der Tabelle Ort angelegt ist, so wird sie hinzugefügt. Die Ortsbezeichnung, die in das Feld Ort eingetragen wird, wird der neuen PLZ angefügt.

```
Ort.MoveFirst
Do While Not Ort.EOF

    Ort.FindFirst "[PLZ] Like '*' & strPLZ & '"
    If Ort!PLZ = Forms!F_Hauptformular!PLZ Then

        aktadr.FindFirst "[ErlagscheinNr] Like '*' & strLfdNr & '"
        If aktadr!ErlagscheinNr = True Then
            aktadr.AddNew
            aktadr!PLZ = Forms!F_Hauptformular!PLZ
            aktadr.Update
        Else
            End If
    Else:
        Ort.AddNew
        Ort!PLZ = Forms!F_Hauptformular!PLZ
        Ort.Update
    End If

    Ort.MoveNext

Loop
```

```
Ort.MoveFirst
    strPLZ = Forms!F_Hauptformular!PLZ

    Ort.FindFirst "[PLZ] Like '*' & strPLZ & '' and isnull (Ort)"
    If IsNull(Ort!Ort) Then
        Ort.Edit
        Ort!Ort = Forms!F_Hauptformular!Ort
        Ort.Update
    Else: MsgBox "BLA"
    End If

Ort.Close
```

## 15.6 Ursprungsadresse

### Ursprungsadresse



ErlagscheinNr	111900001198
PLZ	3331
Ort	Kematen
Strasse Ursprung	Sonntagbergweg 7
Bemerkung	



### **15.6.1. ALLGEMEIN:**

Das Formular Ursprungsadresse ist ebenfalls ein Anzeigeformular. Es besitzt keine Suchfunktion. Der Anwender kann die Ursprungsadresse anzeigen lassen oder diese verändern. Zum gewünschten Datensatz gelangt man, indem man im Hauptformular die entsprechende laufende Nummer angibt und anschließend über das Formular Adressen in das Formular Ursprungsadresse geht.

### **15.6.2. FELDER:**

Die Erlagscheinnummer dient lediglich der Kontrolle. Alle anderen Felder zeigen die Adressdaten des Absolventen. Die Felder PLZ, Ort und Strasse Ursprung sind wie gewohnt zu handhaben. Im Feld Bemerkung kann der Anwender Kommentare eingeben. Da der Etiketten- und Rundschreibendruck nur für die aktuellen Adressen erfolgt, muss bei der Angabe von Bemerkungen auf nichts Rücksicht genommen werden.

Das Feld Postleitzahl ist ebenfalls änderbar. Wenn eine PLZ eingetragen wird, kontrolliert das Programm, ob diese bereits in der Tabelle Ort vorhanden ist. Wenn ja, wird automatisch der in der Tabelle Ort abgespeicherte Wortlaut der Ortsbezeichnung übernommen. Wenn die PLZ noch nicht in der Tabelle Ort angelegt ist, so wird sie hinzugefügt. Die Ortsbezeichnung, die in das Feld Ort eingetragen wird, wird der neuen PLZ angefügt.

## 15.7. e-mail

e - m a i l

**ErlagscheinNr**

21958000993

**neue e-mail-Adresse**

**alte e-mail-Adresse**

ing.gruber@gmx.at

### **15.7.1. ALLGEMEIN:**

Das Formular e-mail ist kein Suchformular. Es zeigt den Datensatz auf, der im Hauptformular gesucht wurde. Es dient zur Anzeige und zum Hinzufügen von neuen e-mail Adressen.

### **15.7.2. FELDER:**

Die angezeigte Erlagscheinnummer dient wieder der Kontrolle. Das Listenfeld alte e-mail-Adresse zeigt an, ob der gesuchte Absolvent eine e-mail Adresse angegeben hat. Es gibt die Möglichkeit für einen Absolventen mehrere e-mail Adressen zu speichern. Deswegen gibt es in der Tabelle e-mail das Feld laufende Nummer. Diese ist eine fortlaufend vergebene e-mail-Nummer, stammt also nicht aus der Erlagschiennummer. Wenn Sie eine neue e-mail-Adresse eintragen wollen, so müssen Sie diese im Feld neue e-mail-Adresse eingeben. Es wird automatisch die nächste fortlaufende e-mail-Nummer vergeben und ein neuer Datensatz mit der entsprechenden Erlagscheinnummer angelegt.

## 15.8. Abschlussdaten

### Abschlussdaten

x

ErlagscheinNr	111964000656
Abschlussjahr Hoehere	1968
Abschlussklasse Hoehere	M
Abschlussjahr Fachschule	1964
Abschlussklasse Fachschule	MF

### 15.8.1. ALLGEMEIN:

Das Formular Abschlussdaten ist ein Anzeigeformular. Es werde die Abschlussdaten nicht nur angezeigt, sie sind hier auch veränderbar.

### 15.8.2. FELDER:

Die Erlagscheinnummer dient als Kontrollfeld. Die folgenden vier Felder zeigen die Abschlussdaten. Das Jahr steht jeweils für das Abschlussjahr, die Klasse für die absolvierte Fachrichtung. Im Feld Klasse steht der Wortlaut der Fachrichtung, der zum Zeitpunkt des Abschlusses gültig war. Wenn es mehrere Züge in diesem Jahrgang gegeben hat, ist es auch möglich, dass am Ende der Abteilungsbezeichnung, dieser durch die Ergänzung a oder b gekennzeichnet ist.

Es werden die Fachschul- und die HTL-Absolventen getrennt geführt, da es nach der Fachschule möglich ist, die HTL zu absolvieren, und so zwei Abschlüsse an der HTL Waidhofen/Ybbs zu machen.

```
If strErlNr = True Then

    If IsNull(Forms!F_Abschlussdaten!ErlagscheinNr) Then
        MsgBox "Erlagscheinnummer nicht vorhanden!"
    End If
    Forms!F_Abschlussdaten!ErlagscheinNr = abschl!ErlagscheinNr
    If IsNull(abschl!AbsJahr_Hoehere) Then
        Forms!F_Abschlussdaten!AbsJahr_Hoehere = " "
    End If
    Forms!F_Abschlussdaten!AbsJahr_Hoehere = abschl!AbsJahr_Hoehere
    If IsNull(abschl!AbsKlasse_Hoehere) Then
        Forms!F_Abschlussdaten!AbsKlasse_Hoehere = " "
    End If
    Forms!F_Abschlussdaten!AbsKlasse_Hoehere = abschl!AbsKlasse_Hoehere
    If IsNull(abschl!AbsJahr_Fachschule) Then
        Forms!F_Abschlussdaten!AbsJahr_Fachschule = " "
    End If
    Forms!F_Abschlussdaten!AbsJahr_Fachschule = abschl!AbsJahr_Fachschule
    If IsNull(abschl!AbsKlasse_Fachschule) Then
        Forms!F_Abschlussdaten!AbsKlasse_Fachschule = " "
    End If
    Forms!F_Abschlussdaten!AbsKlasse_Fachschule =
abschl!AbsKlasse_Fachschule

Else: MsgBox "Fehler bei Schleife aktadr!"
End If
```

## 15.9. Aktionen

### Aktionen

**Absolventen hinzufügen**

**Rundschreiben  
Tabellenansicht**

**Serienbrief öffnen**

**Etikettendruck**

**Versickt abhaken**

**Klassenliste Bericht**

**Klassenliste exportieren**

**AbsolventInnen der HTL**

### 15.9.1. ALLGEMEIN:

Das Formular Aktionen ist, wie alle anderen, über das Hauptformular zu erreichen. Es beinhaltet alle ausführbaren Funktionen, die der Anwender benötigt, um die Arbeit für Absolventenverein zu minimieren. Die verschiedenen Funktionen sind thematisch zusammengehörig, wenn sie grau umrandet sind. Ansonsten ist die Funktion mit nur einem Knopfdruck auszuführen. Eine genaue Beschreibung der Funktionen und wie damit umzugehen ist, finden Sie in den folgenden Punkten.

### 15.9.2. SCHALTFLÄCHE ABSOLVENTEN HINZUFÜGEN:

Mit der Schaltfläche Absolventen hinzufügen, kann der Anwender die neuen Absolventen mit nur einem Knopfdruck in die Datenbank aufnehmen. Da Herr Schneider die Absolventen immer ein Jahr nach ihrem Schulaustritt in die Datenbank aufnahm, ist das Programm auch dafür ausgelegt. Es werden alle benötigten Daten aus dem Excelldokument, das der Absolventenverein jährlich von Schulsekretariat erhält, gelesen oder automatisch ergänzt. Anschließend werden die neuen Datensätze den Tabellen zugewiesen.

#### 15.9.2.1. Ausgangsdokument des Sekretariats:

Frau Spanring aus dem Sekretariat der HTL Waidhofen/Ybbs sendet dem Absolventenvereinsbetreuer jeden Herbst eine Liste in Form eines Excelldokuments. Da dies aus einer Datenbank der Schule stammt, besitzt es gleich bleibend folgende Formatierung.

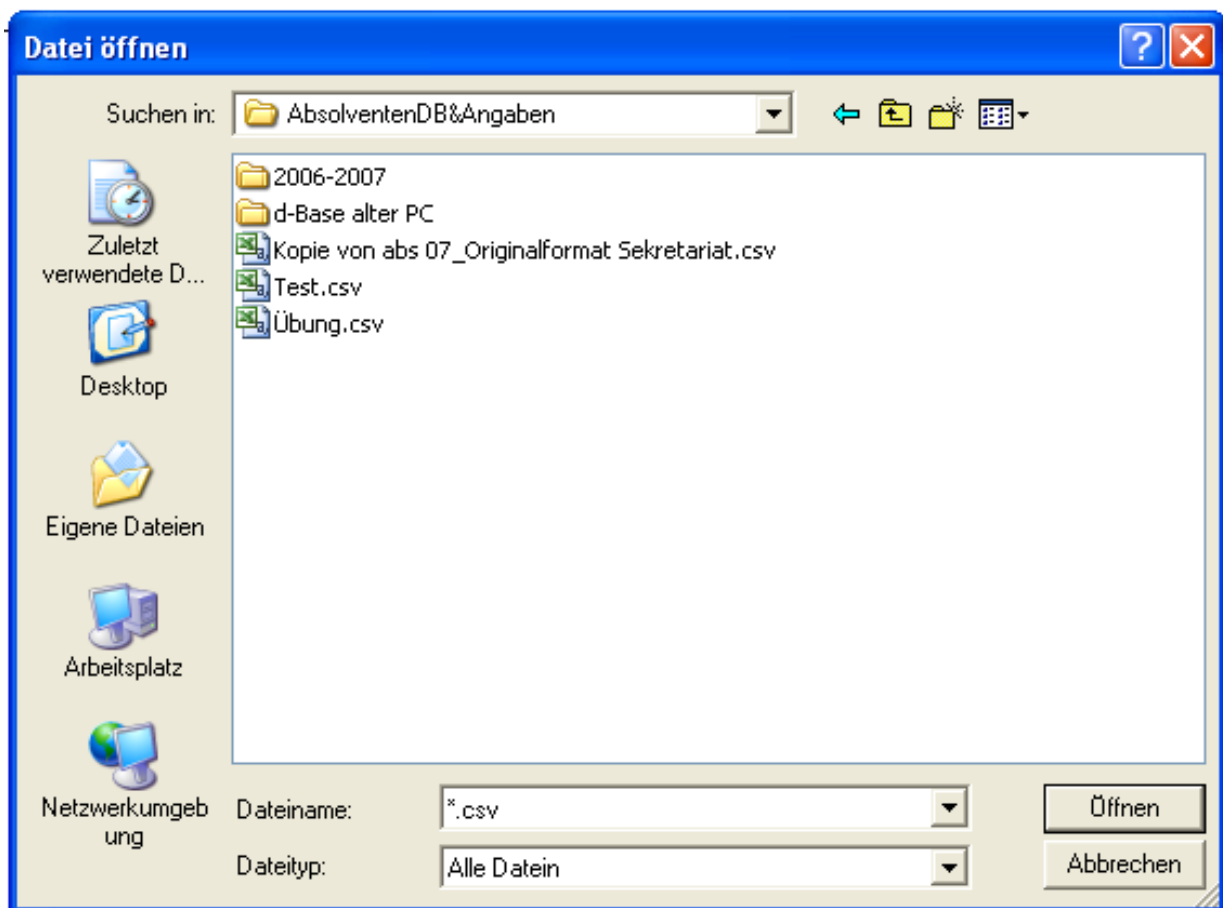
	A	B	C	D	E	F
1	KLASSE	VORNAME	ZUNAME	HEIM_PLZ	HEIM_ORT	HEIM_STR
2	4AFMBF	Peter	AFFENGRUBER	3312	Oed	Raiffeisenstraße 15
3	5AHETR	Markus	AIGLINGER	3321	Ardagger Markt	Markt 67
4	5AHETR	Thomas	AIGNER	3335	Weyer	Rapoldeck 95
5	4AFET	Manuel	AIGNER	3331	Kematen	Heide 3. Str. 3
6	4AFET	Herbert	AIGNER	3373	Kemmelbach	Wolfsbergstraße 9
7	5AHWIM	Matthias	ALBRECHT	3332	Rosenau	Siedlung Baichberg 6
8	5AHETI	Markus	ALTENHOFER	3340	Waidhofen an der Ybbs	Konradsheim 9
9	5AHWII	Florian	ALTWEGER	4360	Grein	Coburgerstraße 13
10	5AHETR	Bernhard	ARTMÜLLER	3251	Purgstall	Franz Eder-Straße 3
11	5AHMIA	Gregor	ATZENHOFER	3335	Weyer	Waidhofnerstraße 86
12	5AHWIM	Daniel	BACHER	3300	Amstetten	Anton Hanl-Straße 2
13	4AFMBF	Patrycjusz	BARANOWSKI	3362	Oehling	Urlweg 30/9
14	5BHMIA	Christoph	BAUER	3264	Gresten	Werkgasse 1
15	5AHMIA	Markus	BECKSTEINER	3340	Waidhofen an der Ybbs	Am Scheibenhang 6

Es ist sehr wichtig, dass an der vorliegenden Formatierung nichts verändert ist, da sich der Code für das Einspielen der Daten an ihr orientiert.

**WICHTIG:** Um diese Informationen problemlos in die Datenbank übernehmen zu können, muss jedoch das Format des Dokuments auf die Extension **.csv** geändert werden. Damit der Anwender nicht darauf vergisst, erscheint diese Meldung auch sofort nach dem Klicken auf den Button Absolventen hinzufügen.

### 15.9.2.2. Einlesen in die Datenbank:

Die Auswahl des Quelldokuments erfolgt über ein Auswahlfenster in dem das Speicherformat .csv bereits vorgegeben ist.



Der Import der Daten erfolgt über den Befehl:

```
DoCmd.TransferText acImportDelim, "absnew_importspez", "T_AbsNew", datnam
```



Nachdem die Datei gewählt wurde, werden die Informationen in die T\_AbsNew importiert. Es werden nicht nur die Daten sondern auch die Spaltenbezeichnungen übernommen. Für die nachfolgenden Prozesse wird die Tabelle automatisch um die Spalte ErlagscheinNr erweitert.

ID	KLASSE	VORNAME	ZUNAME	HEIM_PLZ	HEIM_ORT	HEIM_STR	ErlagscheinNr
12399	4AFMBF	Peter	AFFENGRUBER	3312	Oed	Raiffeisenstraße 15	222006104688
12400	5AHETR	Markus	AIGLINGER	3321	Ardagger Markt	Markt 67	272006104689
12401	5AHETR	Thomas	AIGNER	3335	Weyer	Rapoldeck 95	272006104690
12402	4AFET	Manuel	AIGNER	3331	Kematen	Heide 3. Str. 3	212006104691
12403	4AFET	Herbert	AIGNER	3373	Kemmelbach	Wolfsbergstraße 9	212006104692
12404	5AHWIM	Matthias	ALBRECHT	3332	Rosenau	Siedlung Baichberg 6	292006104693
12405	5AHETI	Markus	ALTENHOFER	3340	Waidhofen an der Ybbs	Konradsheim 9	262006104694
12406	5AHWII	Florian	ALTWEGER	4360	Grein	Coburgerstraße 13	282006104695
12407	5AHETR	Bernhard	ARTMÜLLER	3251	Purgstall	Franz Eder-Straße 3	272006104696
12408	5AHMIA	Gregor	ATZENHOFER	3335	Weyer	Waidhofnerstraße 86	232006104697
12409	5AHWIM	Daniel	BACHER	3300	Amstetten	Anton Hanl-Straße 2	292006104698
12410	4AFMBF	Patrycjusz	BARANOWSKI	3362	Oehling	Urlweg 30/9	222006104699
12411	5BHMA	Christoph	BAUER	3264	Gresten	Werkgasse 1	232006204700
12412	5AHMIA	Markus	BECKSTEINER	3340	Waidhofen an der Ybbs	Am Scheibenhang 6	232006104701
12413	5AHETI	Stefan	BENISCHEK	3300	Amstetten	Oberfeldstraße 3	262006104702

## 15.9.2.3. Generieren der neuen Erlagscheinnummern:

Um die neuen Datensätze in der Absolventendatenbank zuordnen zu können, wird in der temporären Tabelle T\_AbsNew die zwölfstellige Erlagscheinnummer generiert. Diese besteht (wie unter Punkt 14 beschrieben) aus folgenden Teilen:

11	1984	1	12345
Abteilung	Jahrgang	Klasse	lfd. Nummer

Die ersten beiden Ziffern kennzeichnen die **Abteilung** (Fachrichtung), die der Absolvent besuchte. Diese wird aus der Klassenbezeichnung gelesen.

```

KlBezeich = Mid(absnew!Klasse, 3, 4)
If KlBezeich = "HMIA" Then
    Abteilung = 23
ElseIf KlBezeich = "HETI" Then
    Abteilung = 26
ElseIf KlBezeich = "HETR" Then
    Abteilung = 27
ElseIf KlBezeich = "HWII" Then
    Abteilung = 28
ElseIf KlBezeich = "HWIM" Then
    Abteilung = 29
ElseIf KlBezeich = "FET" Then
    Abteilung = 21
ElseIf KlBezeich = "FMBF" Then
    Abteilung = 22

```

Es ist jede Fachrichtung eindeutig durch eine Zahl gekennzeichnet.

**ACHTUNG:** Ändern sich die Klassenbezeichnungen, muss auch der Code für das Einlesen der Abteilung verändert werden.

Der **Jahrgang** wird aus dem aktuellen Jahr des Betriebssystems gelesen. Da Herr Schneider es pflegt, die neuen Absolventen ein Jahr nach ihrem Schulaustritt in die Datenbank aufzunehmen, ist auch die Programmierung diesem Verhalten angepasst. Das aktuelle Jahr wird also um eins verringert.

```
Jahrgang = (year(Date)) - 1
```

Die **Klasse** steht ebenfalls in der Klassenbezeichnung. Sie ist ihr erste Buchstabe. Dieser wird ausgelesen und anschließend für die Klasse A eine 1, für B eine 2, ... vergeben.

```
KlBezeich = Mid(absnew!Klasse, 2, 1)
If KlBezeich = "A" Then
    Klasse = 1
ElseIf KlBezeich = "B" Then
    Klasse = 2
ElseIf KlBezeich = "C" Then
    Klasse = 3
ElseIf KlBezeich = "D" Then
    Klasse = 4
Else
    Klasse = 5
End If
```

Die letzten fünf Ziffern der Erlagscheinnummer ist die **fortlaufende Nummer**. Um eine neue Laufende Nummer vergeben zu können, muss die höchste bereits bestehende ermittelt werden. Dies erfolgt mit Hilfe der temporären Tabelle T\_lfdNr und der Abfrage A\_lfdNr. Es werden von jeder bestehenden Erlagscheinnummer die letzten fünf Zahlen ermittelt und anschließend in die Tabelle T\_lfdNr geschrieben. Die Abfrage A\_lfdNr sortiert die T\_lfdNr absteigend. So steht die höchst fortlaufende Nummer an der ersten Stelle. Diese wird ausgelesen und anschließend jeweils nach dem Schreiben eines Datensatzes um eins erhöht.

```
absnew.Update
absnew.MoveNext

newlfdNr = newlfdNr + 1
```

	lfdNr
►	04652
	04651
	04650
	04649
	04648
	04647
	04646
	04645
	04644
	04643
	04642
	04641
	04640
	04639
	04638

#### 15.9.2.4. Schreiben der Tabellen:

Es werden nacheinander alle Tabellen beschrieben.

```
Call Abs_stamm  
Call Abs_abschlussdaten  
Call Abs_ursprungsadr  
Call Abs_aktadr  
Call Abs_ort
```

Den Anfang macht die Haupttabelle **T\_stamm**. In ihr werden alle Absolventen beim Geschlecht auf „Herr“ gesetzt, da es in der HTL leider nur wenige weibliche Absolventen gibt. Die Damen müssen im Anschluss händisch auf „Frau“ umgebessert werden. Daran erinnert zum Schluss des Einlesevorgangs eine MessageBox. Weiters wird bei allen Personen, die die HTL absolviert haben der akademische Titel auf „Ing.“ gesetzt.

In der Tabelle **T\_Abs\_Daten** stehen der Abschlussjahrgang, die Fachrichtung und die Klasse. Da es möglich im Anschluss an die Fachschule die HTL zu besuchen, sind diese Positionen jeweils für die Fachschule und die HTL vorhanden. Diese werden je nach Klassenbezeichnung eingetragen.

```
If Abteilung = "HMIA" Then  
    absch!AbsJahr_Hoehere = Jahrgang  
    absch!AbsKlasse_Hoehere = Abteilung & Klasse  
ElseIf Abteilung = "HETI" Then  
    absch!AbsJahr_Hoehere = Jahrgang  
    absch!AbsKlasse_Hoehere = Abteilung & Klasse
```

Da in der Datenbank die Ursprungsadresse für die zum Zeitpunkt des Abschlusses aktuelle Adresse steht, werden die Tabellen **T\_Ursprungliche\_Adresse** und **T\_Aktuelle\_Adresse** mit den Adressdaten der T\_AbsNew beschrieben. In ihnen stehen jedoch nur die PLZ und die Straßenbezeichnung um Redundanzen zu vermeiden. Die Ortsbezeichnung zur jeweiligen PLZ befindet sich in der Tabelle **T\_Ort**. Beim Schreiben der Adressdaten wird abgefragt, ob die jeweilige PLZ bereits in der Tabelle Ort vorhanden ist. Ist dies nicht der Fall, so werden die PLZ und die Ortsbezeichnung an die Tabelle T\_Ort angefügt.

```
Ort.MoveFirst  
absnew.MoveFirst  
Do While Not absnew.EOF  
    strPLZ = absnew!HEIM_PLZ
```

```
Ort.FindFirst "[PLZ] Like '*' & strPLZ & '"
If Ort!PLZ = absnew!HEIM_PLZ Then
Else:
    Ort.AddNew
    Ort!PLZ = absnew!HEIM_PLZ
    Ort.Update
End If

Ort.FindFirst "[PLZ] Like '*' & strPLZ & '" and Ort Is Null"
If IsNull(Ort!Ort) Then
    Ort.Edit
    Ort!Ort = absnew!HEIM_ORT
    Ort.Update
Else
End If

End If

absnew.MoveNext
Loop
```

Abschließend werden die Tabelleninhalte der T\_Absnew und T\_lfdNr gelöscht.

```
CurrentDb().Execute "DELETE FROM T_AbsNew", dbFailOnError
CurrentDb().Execute "DELETE FROM T_lfdNr", dbFailOnError
```

## 15.9.3. RUNDSCHREIBEN

### 15. 9.3.1. ALLGEMEINES:

Jährlich wird an alle Absolventen ein Briefumschlag gesendet. Diese Sendung beinhaltet neben einem Mitteilungsblatt des Absolventenvereins und anderen Informationen auch einen Erlagschein.

Der Erlagschein dient zur Einzahlung des Mitgliedbeitrages, einer Spende und des Geldes für den Jahresbericht. Wenn ein Jahresbericht gewünscht wird, muss dies im Feld Verwendungszweck des Erlagscheines angegeben werden.

Jeder Erlagschein enthält die Adresse des Absolventen für die Zusendung. Damit ein eingezahlter Erlagschein dem jeweiligen Absolventen zugewiesen werden kann, befindet sich die Erlagscheinnummer aus der Absolventendatenbank als Identifikation auf dem Erlagschein.

Das heißt, dass für jeden Absolventen ein eigener Erlagschein bedruckt werden muss.

Angesichts der großen Menge an Daten erfolgt dies automatisch über einen Serienbrief beziehungsweise Seriendruck in Microsoft Word.

Der Serienbrief soll natürlich nur an jene Absolventen gesendet werden, von denen eine aktuelle Adresse in der Datenbank vorhanden ist. Da dies auch bei verstorbenen oder bei Absolventen, die umgezogen sind, der Fall sein kann, müssen auch diese ausgeschlossen werden.

Dies passiert über die Abfrage „A\_Rundschreiben“. Die Abfrage benötigt folgende Tabellen und Felder:

Tabelle	Feld	Verwendung
T_Stammdaten:	ErlagscheinNr	ID für Erlagschein
	Geschlecht	für Zusendung
	Akademischer_Titel	für Zusendung
	Vorname	für Zusendung
	Nachname	für Zusendung
	Bemerkung	Ausschluss v. verstorbenen Abs.
T_Aktuelle_Adresse:	Strasse	für Zusendung
	PLZ	für Zusendung
	Bundesland	für Zusendung
	Land	für Zusendung
	Bemerkung	Ausschluss von verzogenen Abs.
T_Ort:	Ort	Für Zusendung

**ABBILDUNG:  
SERIENBRIEF MIT DEN BENÖTIGTEN FELDERN** (gelb untermalt)

**Werter Absolvent !**  
**!!! Bitte kreuzen Sie im Verwendungszweck unbedingt das Gewünschte an !!!**  
Nur so können wir sicherstellen, dass alles ordnungsgemäß erledigt wird und Sie, wenn gewünscht einen Jahresbericht erhalten.

**Adressänderung !**  
Wenn Sie Ihre Adresse geändert haben, verwenden Sie zur Überweisung bitte den Original-Erlagschein und schreiben Ihre neue Adresse ins Feld Auftraggeber, senden uns eine E-mail oder besuchen unsere Homepage

**WICHTIG für Überweisungen mit Electronic Banking !!**  
Bitte geben Sie diese Nummer **ErlagscheinNr** im Feld Kundendaten ein.  
Nur so ist eine effiziente Bearbeitung der Zahlungseingänge möglich!  
Unter Verwendungszweck geben Sie bitte MB (für Mitgliedsbeitrag), **JB (für Jahresbericht)** oder SP (für eine eventuelle Spende) ein.

**Für Überweisungen aus dem Ausland** BIC: SPSPAT21  
IBAN: AT782025603400001198

Österreichische Post AG Info.Mail Entgelt bezahlt

Geschlecht  
Akademischer Titel Vorname Nachname  
Strasse  
PLZ Ort Bundesland  
Land

SPARKASSE ST. POELTEN  
TM Waidhofen an der Ybbs

3400-001198 20256  
ABSOLVENTENVEREIN DER HTL  
3340 Waidhofen/YBBS

€ 4,50 Mitgliedsbeitrag 2006 ☐  
€ 7,50 Jahresbericht 2006 ☐  
€ ..... Spende ☐  
Unbedingt ankreuzen ! ↑  
Electronic Banking  
Kundendaten:  
**ErlagscheinNr**

84+ ErlagscheinNr 03400001198+ 00020256>

< 40+

### 15.9. 3.2. SQL-Abfrage „A\_Rundschreiben“:

#### 15. 9. 3.2.1. CODE

```

SELECT    T_Stammdaten.ErlagscheinNr, T_Stammdaten.Geschlecht,
            T_Stammdaten.Akademischer_Titel, T_Stammdaten.Vorname,
            T_Stammdaten.Nachname, T_Aktuelle_Adresse.Strasse,
            T_Aktuelle_Adresse.PLZ, T_Ort.Ort,
            T_Aktuelle_Adresse.Bundesland, T_Aktuelle_Adresse.Land

FROM      T_Ort INNER JOIN (T_Stammdaten INNER JOIN
            T_Aktuelle_Adresse ON T_Stammdaten.ErlagscheinNr =
            T_Aktuelle_Adresse.ErlagscheinNr) ON T_Ort.PLZ =
            T_Aktuelle_Adresse.PLZ

WHERE      (((T_Aktuelle_Adresse.Strasse) Is Not Null) AND
            ((T_Stammdaten.Bemerkung) Is Null) AND
            ((T_Aktuelle_Adresse.Bemerkung) Is Null)) OR
            (((T_Aktuelle_Adresse.Strasse) Like "") AND
            ((T_Stammdaten.Bemerkung) Not Like "*storb*") AND
            ((T_Aktuelle_Adresse.Bemerkung) Not Like "*zog*"))

ORDER BY T_Aktuelle_Adresse.Land, T_Aktuelle_Adresse.PLZ;

```

#### 15.9.3.2.2. AUSGABE-Felder

Die Abfrage „A\_Rundschreiben“ ist eine Auswahlabfrage, bei der folgende Felder durch die SELECT - Anweisung aus den Tabellen ausgegeben werden:

Tabelle	Feld
T_Stammdaten:	ErlagscheinNr
	Geschlecht
	Akademischer_Titel
	Vorname
	Nachname
T_Aktuelle_Adresse:	Strasse
	PLZ
	Bundesland
	Land
T_Ort:	Ort

### 15.9.3.2.3. Tabellen und Beziehungen

Die benötigten Tabellen und deren Verknüpfungen werden durch die FROM-Anweisung deklariert.

Die Tabellen „T\_Aktuelle\_Adresse“ und „T\_Stammdaten“ sind durch deren Primärschlüssel „ErlagscheinNr“ verknüpft.  
Die Tabellen „T\_Aktuelle\_Adresse“ und „T\_Ort“ werden über das Feld Postleitzahl verbunden.

Graphische Darstellung der Verknüpfungen:

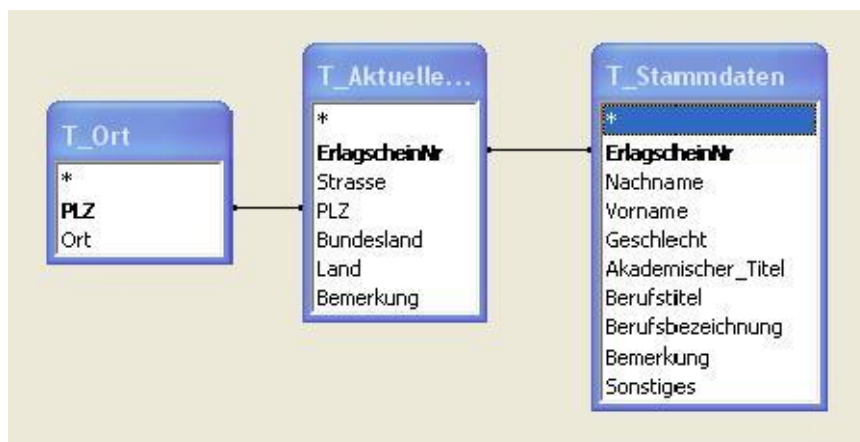


Abbildung: Beziehungen zwischen den Tabellen bei der Abfrage „Rundschreiben“

### 15.9.3.2.4. Auswahlbedingungen

In der WHERE-Klausel befinden sich folgende drei Bedingungen, wobei alle drei erfüllt sein müssen.

1. **(T\_Aktuelle\_Adresse.Strasse Is Not Null) OR  
(T\_Aktuelle\_Adresse.Strasse Like "\*")**

Die Abfrage wählt alle Absolventen aus, die Datensätze in der Tabelle „T\_Aktuelle\_Adresse“ besitzen, es sei denn, das Feld Strasse besitzt keinen Wert -> „Strasse Is Not Null“ (Ist Nicht Null).

Das würde bedeuten, dass keine Adresse vorhanden ist, zu der man senden könnte.

Der Befehl „Strasse Like “\*” “ bedeutet, dass in dem Feld Strasse Zeichen vorhanden sein müssen.



**2. (T\_Stammdaten.Bemerkung Is Null) OR  
(T\_Stammdaten.Bemerkung Not Like "\*storb\*")**

Es werden auch jene Datensätze nicht ausgewählt, die im Feld „Bemerkung“ in der Tabelle „T\_Stammdaten“ den Wortlaut „\*storb\*“ beinhalten. Dies schließt Datensätze aus mit der Bemerkung „verstorben“ oder „gestorben“. Mit dieser Bedingung wird ausgeschlossen, dass Sendungen an bereits verstorbene Absolventen, die aber noch immer eine aktuelle Adresse in der Datenbank besitzen, gehen.

**3. (T\_Aktuelle\_Adresse.Bemerkung Is Null) OR  
(T\_Aktuelle\_Adresse.Bemerkung Not Like "\*zog\*")**

Ebenso werden Datensätze ausgeschlossen die den Wortlaut „\*zog\*“ im Bemerkungsfeld der Tabelle „T\_Aktuelle\_Adresse“ enthalten. Dieser Wortlaut bezieht sich auf „verzogen“ oder „umgezogen“. Auch an Absolventen mit diesem Eintrag im Feld „Bemerkung“ in der Tabelle „T\_Aktuelle\_Adresse“ soll nicht gesendet werden.

### **15.9.3.2.5. Sortierung**

Sortiert werden die ausgewählten Datensätze nach dieser Anweisung:

ORDER BY T\_Aktuelle\_Adresse.Land, T\_Aktuelle\_Adresse.PLZ

Alle Absolventen, die die obigen Kriterien erfüllt haben, werden ausgewählt und anhand der Felder „PLZ“ und „Land“ aufsteigend sortiert.

Die Sortierung nach dem Inhalt des Feldes „Land“ erfolgt, um die Datensätze mit Adressen im Ausland herauszufiltern. Dies ist wichtig da diese Briefe bei der Post getrennt von den anderen behandelt werden. Außerdem dauert das Senden der Briefe ins Ausland länger, weshalb diese als erstes versendet werden. Inländische (österreichische) Adressen besitzen keinen Eintrag im Feld „Land“.

Die restlichen (österreichischen) Adressen werden nach der Postleitzahl sortiert. Dies ist nötig um diese große Menge an Briefen ordnungsgemäß aufgeben zu können.

### **15.9.3.3. Button „Rundschreiben Tabellenansicht“:**

Der Button „Rundschreiben – Tabellenansicht“ im Formular „F\_Aktionen“ führt die Abfrage „A\_Rundschreiben“ aus und zeigt die Daten dieser Abfrage in der Tabellenansicht. Diese Funktion dient zur übersichtlichen Ansicht der ausgewählten Daten und zur Überprüfung.

### **15.9.3.4. Schaltfläche „Serienbrief“:**

Der Button Serienbrief im Formular „F\_Aktionen“ öffnet die Datei „Rundschreiben.doc“, die sich wie die Datenbank auf dem Absolventenvereinlaufwerk befindet.

Beim Öffnen des Serienbriefes wird die Abfrage „A\_Rundschreiben“ ausgeführt und die Daten der Abfrage in den formatierten Brief eingefügt.

Das Drucken der Erlagscheine erfolgt im Menü über „Datei“ -> „Drucken“, oder durch gleichzeitiges Drücken der Tasten „STRG“ und „P“.

## 15.9.5. ETIKETTENDRUCK

### 15.9.4.1. ALLGEMEINES

Der Jahresbericht wird an alle Absolventen versendet, die auf ihrem eingezahlten Erlagschein, das Feld Jahresbericht angehakt haben und das Geld für den Jahresbericht überwiesen haben.

Die Adressen der Absolventen, die den Jahresbericht zugesendet bekommen, werden zum Versenden des Jahresberichts auf selbstklebende Etiketten gedruckt.

Dies erfolgt über einen Bericht. Der Bericht ist dem Etikettenpapier entsprechend formatiert. Die Adresse muss auf ein Feld von 7,0 mm Breite x 3,6 mm Höhe gedruckt werden. Dies ist die Größe einer Etikette. Auf einem A4-Blatt Etikettenpapier befinden sich in drei Spalten und acht Zeilen 24 einzelne Etiketten.

Für den Versand des Jahresberichts kommen nur jene Absolventen in Frage, bei denen in der Tabelle „T\_Beitraege“ das Feld „Jahresbericht“ angehakt, das heißt auf „True“ gesetzt ist, und das Feld „Verschickt“ nicht angehakt, das heißt auf „False“ gesetzt ist.

Das betrifft alle Absolventen, die einen Jahresbericht bestellt, aber noch nicht erhalten haben. Diese Auswahl erledigt die Abfrage „A\_Etikettendruck“.

Die Abfrage benötigt folgende Tabellen und Felder zur Auswahl:

<b>Tabelle</b>	<b>Feld</b>	<b>Verwendungszweck</b>
T_Stammdaten:	Geschlecht	Adressenausgabe
	Akademischer_Titel	Adressenausgabe
	Vorname	Adressenausgabe
	Nachname	Adressenausgabe
T_Aktuelle_Adresse:	Strasse	Adressenausgabe
	PLZ	Adressenausgabe
	Bundesland	Adressenausgabe
	Land	Adressenausgabe
T_Ort:	Ort	Adressenausgabe
T_Beitraege:	Jahrbuch	Auswahlkriterium
	Verschickt	Auswahlkriterium

### 15. 9.4.2. SQL-Abfrage „A\_Etikettendruck“

#### 15. 9. 4.2.1. Code

```

SELECT  T_Stammdaten.Geschlecht, T_Stammdaten.Akademischer_Titel,
        T_Stammdaten.Vorname, T_Stammdaten.Nachname,
        T_Aktuelle_Adresse.Strasse, T_Aktuelle_Adresse.PLZ, T_Ort.Ort,
        T_Aktuelle_Adresse.Bundesland, T_Aktuelle_Adresse.Land

FROM    T_Ort INNER JOIN ((T_Stammdaten INNER JOIN
        T_Aktuelle_Adresse ON T_Stammdaten.ErlagscheinNr =
        T_Aktuelle_Adresse.ErlagscheinNr) INNER JOIN T_Beitraege ON
        T_Stammdaten.ErlagscheinNr = T_Beitraege.ErlagscheinNr) ON
        T_Ort.PLZ = T_Aktuelle_Adresse.PLZ

WHERE   (((T_Beitraege.Jahrbuch)=True) AND
        ((T_Beitraege.Verschickt)=False))

ORDER BY T_Aktuelle_Adresse.Land, T_Aktuelle_Adresse.PLZ;

```

#### 15. 9. 4.2.2. Ausgabefelder

Die SELECT – Anweisung gibt die folgenden Felder bei der Auswahlabfrage „A\_Etikettendruck“, die zum Adressdruck benötigt werden aus:

Tabelle	Feld
T_Stammdaten:	Geschlecht
	Akademischer_Titel
	Vorname
	Nachname
T_Aktuelle_Adresse:	Strasse
	PLZ
	Bundesland
	Land
T_Ort:	Ort

### 15. 9. 4.2.3. Tabellen und Beziehungen

Die Tabellen „T\_Ort“ und „T\_Aktuelle\_Adresse“ werden über das Feld PLZ verknüpft. Die restlichen Tabellen „T\_Stammdaten“, „T\_Aktuelle\_Adresse“ und „T\_Beitraege“ werden über deren Primärschlüssel „ErlagscheinNr“ verbunden.

Graphische Darstellung der Beziehungen:

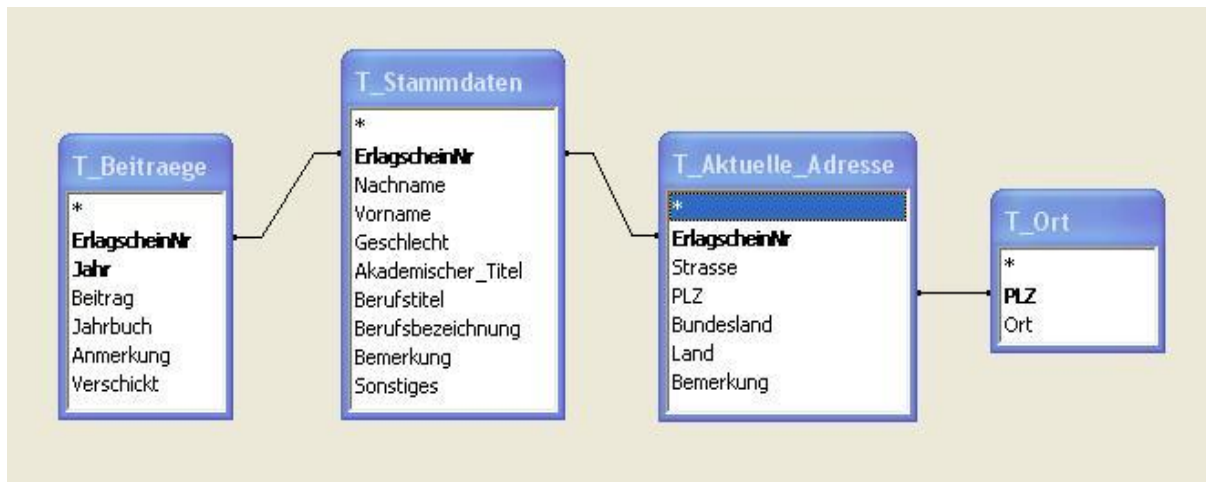


Abbildung: Beziehungen zwischen den Tabellen bei der Abfrage „Rundschreiben“

### 15. 9. 4.2.4. Auswahlbedingungen

Die Felder „Jahrbuch“ und „Verschickt“ in der Tabelle „T\_Beitraege“ sind binäre Felder. Sie können nur die Werte „Wahr“ oder „Falsch“ darstellen.

Das Kriterium zur Auswahl eines Datensatzes ist:

WHERE (((T\_Beitraege.Jahrbuch)=True) AND  
((T\_Beitraege.Verschickt)=False))

Es werden nur jene Datensätze ausgewählt, bei denen in der Tabelle „T\_Beitraege“ das Feld „Jahrbuch“ auf „Wahr“ und das Feld „Verschickt“ auf „Falsch“ gesetzt sind.

### 15. 9. 4.2.5. Sortierung

Die ausgewählten Datensätze werden aufgrund der Inhalte der Felder „PLZ“ und „Land“.

**ORDER BY** T\_Aktuelle\_Adresse.Land, T\_Aktuelle\_Adresse.PLZ;  
Somit können die Kuverts sortiert aufgegeben werden.

### 15. 9. 4.3. SCHALTFLÄCHE „Etikettendruck“

Der Button „Etikettendruck“ im Formular „F\_Aktionen“ öffnet den Bericht „B\_Etiketten“, der automatisch die Abfrage „A\_Etikettendruck“ ausführt.

Die Adressdaten, der durch die Abfrage ausgewählten Absolventen werden in den Bericht eingefügt. Der Bericht wird in der Seitenansicht geöffnet.

Die Adressdaten im Bericht sind folgendermaßen formatiert:

Geschlecht									
=[Akademischer_Titel] & " " & [Vorname] & " " & [Nachname]									
Strasse									
=[PLZ] & " " & [Ort] & [Bundesland]									
Land									

Abbildung: Entwurfsansicht des Berichts „B\_Etikettendruck“

Wobei das Berichtsfeld eine Höhe von 3,6 mm und eine Breite von 7,0 mm haben muss. Auf einem A4-Blatt Etikettenpapier müssen 24 Datensätze platz finden. Dazu müssen die Ränder des bedruckbaren Bereiches richtig eingestellt sein.

Unter „Datei“ -> „Seite einrichten“ müssen die Ränder links und rechts auf 0,0 cm gesetzt werden. Der obere und untere Abstand des bedruckbaren Bereiches müssen 0,43 cm betragen. Nur bei diesen Einstellungen wird das Etikettenpapier richtig bedruckt.

Zum Drucken ist der Menüpunkt „Datei“ -> „Drucken“ zu wählen, oder sind die Tasten „STRG“ und „P“ gleichzeitig zu drücken.

#### **15. 9. 4.4. SCHALTFLÄCHE „Verschickt Abhaken“**

Nachdem die Etiketten für den Versand des Jahrbuches gedruckt wurden, ist es notwendig das Feld „Verschickt“ in der Tabelle „T\_Beitraege“ abzuhaken.

Da das Verschicken der Jahresbücher etappenweise vollzogen wird und sonst diese Datensätze beim nächsten Mal wieder ausgewählt und das Jahrbuch ein zweites Mal verschickt werden würde.

Bei Klicken der Schaltfläche „Verschickt abhaken“ wird die SQL-Abfrage „A\_Verschickt“ ausgeführt.

Somit ist gewährleistet, dass die Etiketten für jeden gewünschten Jahresbericht nur einmal ausgedruckt werden.

Der Button darf allerdings wirklich nur dann angewählt werden, wenn die Etiketten schon gedruckt wurden.

Es erfolgt automatisch ein Warnmeldung von MS Access, dass dieser Button eine Aktualisierungsabfrage durchführt. Diese Abfrage verändert Daten in den Tabellen und dieser Befehl kann nicht mehr rückgängig gemacht werden.

##### **15. 9. 4.4.1. SQL-Abfrage „A\_Verschickt“**

Die Abfrage A\_Verschickt ist eine Aktualisierungsabfrage, dass heißt die Abfrage manipuliert die Datensätze in der Tabelle.

Code der Abfrage „A\_Verschickt“:

```
UPDATE    T_Beitraege

SET       T_Beitraege.Verschickt = True

WHERE     T_Beitraege.Jahrbuch=True And
            T_Beitraege.Verschickt=False;
```

Mit der Anweisung UPDATE wird die Tabelle ausgewählt, die aktualisiert werden soll.

Mit der SET-Anweisung wird angegeben, dass das Feld „Verschickt“ auf „Wahr“ gesetzt werden soll.

Durch die WHERE-Bedingung wird bei den gleichen Datensätzen, die bei der Abfrage „A\_Etikettendruck“ ausgewählt wurden, das Feld „Verschickt“ auf „Wahr“ gesetzt. Das bedeutet jene Datensätze, bei denen das Feld „Jahrbuch“ auf „Wahr“ gesetzt ist und das Feld „Verschickt“ auf „Falsch“.

## 15. 9.5. KLASSENLISTE

### 15. 9.5.1. ALLGEMEINES

Diese Abfrage ermöglicht die Ausgabe von Namen und Adressdaten von einer beliebigen Abschlussklasse. Die Ausgabefelder können auf Wunsch jederzeit erweitert werden. Die Abfrage eignet sich zum Beispiel für Adresslisten für Klassentreffen.

Folgende Felder sind für die Abfrage „A\_Klassenliste“ erforderlich

Tabelle	Feld	Verwendungszweck
T_Stammdaten:	Vorname	Ausgabefeld
	Nachname	Ausgabefeld
T_Aktuelle_Adresse:	Strasse	Ausgabefeld
	PLZ	Ausgabefeld
	Bundesland	Ausgabefeld
	Land	Ausgabefeld
T_Ort:	Ort	Ausgabefeld
T_email	E-Mail	Ausgabefeld
T_Abs_Daten	AbsJahr_Hoehere	Auswahlbedingung
	AbsKlasse_Hoehere	Auswahlbedingung
	AbsJahr_Fachschule	Auswahlbedingung
	AbsKlasse_Fachschule	Auswahlbedingung



## 15. 9. 5.2. SQL-Abfrage „A\_Klassenliste“

### 15. 9. 5.2.1. CODE

```
SELECT    T_Stammdaten.Vorname, T_Stammdaten.Nachname,
           T_Aktuelle_Adresse.Strasse, T_Aktuelle_Adresse.PLZ, T_Ort.Ort,
           T_Aktuelle_Adresse.Bundesland, T_Aktuelle_Adresse.Land,
           T_email.E_Mail

FROM      (T_email RIGHT JOIN (T_Aktuelle_Adresse RIGHT JOIN
           (T_Abs_Daten LEFT JOIN T_Stammdaten ON
           T_Abs_Daten.ErlagscheinNr = T_Stammdaten.ErlagscheinNr) ON
           T_Aktuelle_Adresse.ErlagscheinNr =
           T_Abs_Daten.ErlagscheinNr) ON T_email.ErlagscheinNr =
           T_Abs_Daten.ErlagscheinNr) LEFT JOIN T_Ort ON
           T_Aktuelle_Adresse.PLZ = T_Ort.PLZ

WHERE      (((T_Abs_Daten.AbsJahr_Hoehere)=[Klassenliste für welches
           Abschlussjahr?]) AND
           ((T_Abs_Daten.AbsKlasse_Hoehere)=[Klassenliste für welche
           Klasse?])) OR
           (((T_Abs_Daten.AbsJahr_Fachschule)=[Klassenliste für welches
           Abschlussjahr?]) AND
           ((T_Abs_Daten.AbsKlasse_Fachschule)=[Klassenliste für welche
           Klasse?]))

ORDER BY T_Stammdaten.Nachname;
```

### 15. 9. 5.2.2. Ausgabefelder

Durch die Anweisung SELECT werden folgende Felder werden aus den Tabellen ausgegeben:

Tabelle	Feld
T_Stammdaten:	Vorname
	Nachname
T_Aktuelle_Adresse:	Strasse
	PLZ
	Bundesland
	Land
T_Ort:	Ort
T_email	E_Mail

### 15. 9. 5.2.3. Tabellen und Beziehungen

Alle Tabellen werden über deren Primärschlüssel „ErlagscheinNr“ verbunden. Ausnahme: Die Tabellen „T\_Aktuelle\_Adresse“ und „T\_Ort“ werden über das Feld „PLZ“ verknüpft.

Die Tabellen sind auf folgende Weise miteinander verknüpft:

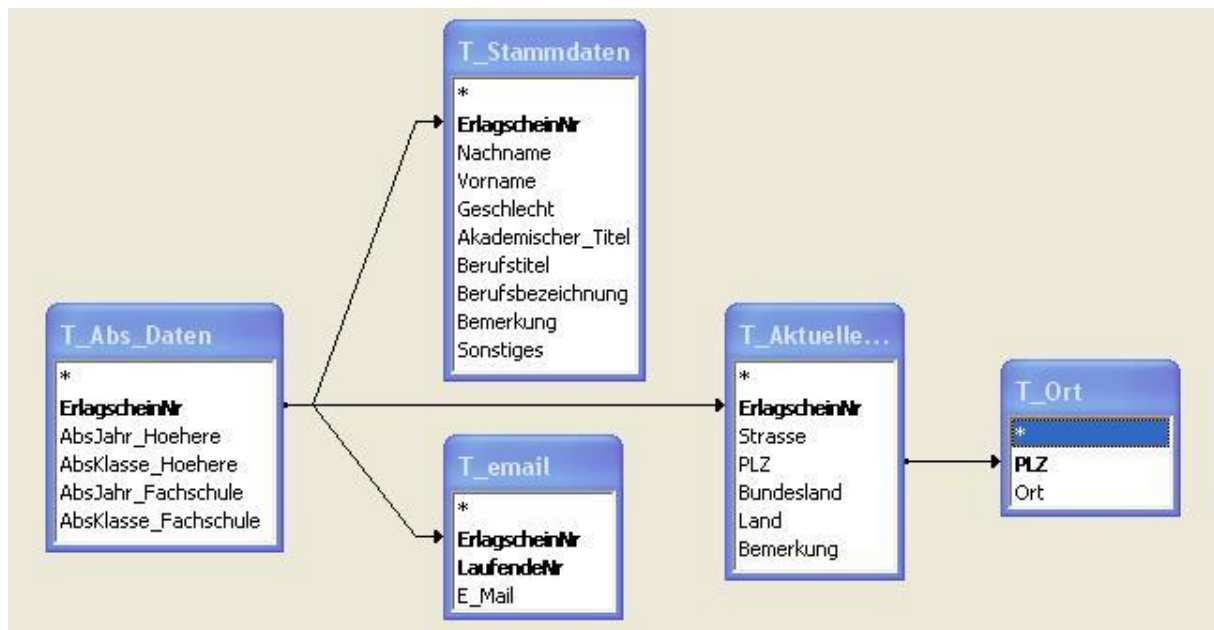


Abbildung: Beziehungen zwischen den Tabellen bei der Abfrage „Klassenliste“

### 15. 9. 5.2.4. Auswahlbedingungen

Damit diese Abfrage ausgeführt werden kann, müssen Parameter eingegeben werden. Das Parameterwerteingabefeld öffnet sich automatisch, sobald die Abfrage ausgeführt wird. Die Vergleichsoperatoren sind „Jahr“ und „Klassenbezeichnung“.

Zuerst wird das Abschlussjahr abgefragt („Abschlussjahr?“). Die Eingabe erfordert das Abschlussjahr der gewünschten Klasse mit vier Ziffern (z.B.: „1997“). Danach muss die Klassenbezeichnung eingegeben werden (z.B.: „ef“). Auf Groß- beziehungsweise Kleinschreibung wird nicht geachtet.

Ausgegeben werden jene Datensätze (Absolventen), bei denen entweder beide Eingabewerte mit den Feldern „AbsJahr\_Hoehere“ und „AbsKlasse\_Hoehere“, oder mit „AbsJahr\_Fachschule“ und „AbsKlasse\_Fachschule“ aus der Tabelle „T\_Abs\_Daten“ übereinstimmen.

#### **15. 9. 5.2.5. Sortierung**

Die ausgewählten Datensätze werden mit der Anweisung ORDER BY dem Nachnamen nach aufsteigend sortiert.

#### **15. 9. 5.3. Schaltfläche „Klassenliste – Bericht“**

Durch das Anwählen des Buttons „Klassenliste – Bericht“ im Formular „F\_Aktionen“ wird die Abfrage „A\_Klassenliste“ ausgeführt und die Daten im Bericht „B\_Klassenliste“ tabellarisch aufgelistet. Diese Funktion eignet sich zum Drucken von Klassenlisten.

#### **15. 9. 5.4. Schaltfläche „Klassenliste exportieren“**

Klassenlisten werden des Öfteren auch als E-Mail in Form von Microsoft Excel Dateien versandt. Der Button „Klassenliste exportieren“ im Formular „F\_Aktionen“ führt die Abfrage nach der Abschlussklasse „A\_Klassenliste“ aus und führt nach der Parametereingabe automatisch den Befehl exportieren aus. Dadurch öffnet sich das Fenster „Exportieren“, in welchem der Pfad und der gewünschte Dateityp zum Exportieren der Klassenliste eingegeben werden können.

## 15. 9.6. ABSOLVENTINNEN

### 15. 9.6.1. Schaltfläche „Absolventinnen der HTL“

Bei anwählen dieser Schaltfläche wird eine Auswahlabfrage ausgeführt, die alle Mädchen angibt, die die HTBLuVA Waidhofen/Ybbs absolviert haben.

### 15. 9. 6.2. SQL-Abfrage „A\_Absolventinnen“

#### 15. 9. 6.2.1. Code

```
SELECT    T_Stammdaten.Vorname,
            T_Stammdaten.Nachname,
            T_Abs_Daten.AbsJahr_Hoehere,
            T_Abs_Daten.AbsKlasse_Hoehere,
            T_Abs_Daten.AbsJahr_Fachschule,
            T_Abs_Daten.AbsKlasse_Fachschule

FROM      T_Stammdaten INNER JOIN T_Abs_Daten ON
            T_Stammdaten.ErlagscheinNr = T_Abs_Daten.ErlagscheinNr

WHERE      (((T_Stammdaten.Geschlecht)="Frau"))

ORDER BY  T_Abs_Daten.AbsJahr_Hoehere,
            T_Abs_Daten.AbsJahr_Fachschule;
```

#### 15. 9. 6.2.2. Ausgabefelder

Durch die SELECT - Anweisung werden folgende Felder ausgegeben:

<b>Tabelle</b>	<b>Feld</b>
T_Stammdaten:	Vorname
	Nachname
T_Abs_Daten:	AbsJahr_Hoehere
	AbsKlasse_Hoehere
	AbsJahr_Fachschule
	AbsKlasse_Fachchule

### 15. 9. 6.2.3. Tabellen und Beziehungen

Die beiden Tabellen, die zur Ausgabe dieser Daten notwendig sind, sind verknüpft über ihr Primärschlüsselfeld „ErlagscheinNr“.

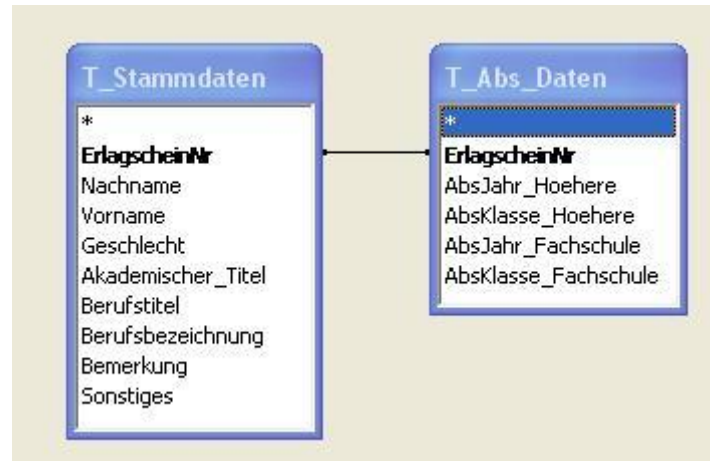


Abbildung: Beziehung zwischen den Tabellen in der Abfrage A\_Absolventinnen

### 15.9. 6.2.4 Auswahlbedingung und Sortierung

Ausgegeben werden, bedingt durch die WHERE - Klausel, Datensätze, bei denen das Feld „Geschlecht“ in der Tabelle „T\_Stammbaum“ den Wert „Frau“ aufweist.

Sortiert werden die Datensätze mit der ORDER BY - Anweisung nach dem Abschlussjahr in der Höheren Abteilung beziehungsweise in der Fachschule.

## 16. Einspielen der Absolventendaten

### 16.1. Alte Datenbankstruktur

Die alte Datenbankstruktur war eine einzige große Tabelle, die alle Daten beinhaltete. Sie entsprach also keiner Normalform.

	Feldname	Felddatentyp
	ERLAGSCHEIN-NR	Text
?	LFD NR	Zahl
	BEITRAG 05-06	Währung
	JAHRB 05-06	Ja/Nein
	FAMNAME	Text
	VORNAME	Text
	AKADTITEL	Text
	ABSJAHR	Text
▶	ABSJAHR1	Text
	LAND-PLZ	Text
	LAND	Text
	BUNDESLAND	Text
	PLZ	Zahl
	ORT	Text
	STRASSE	Text
	HERRN_FRAU	Text
	KONTONUMMER	Text
	ANMERKUNG	Text
	BERUFTITEL	Text
	BERUFBEZ	Text
	GEBURTSDATUM	Text
	E-MAIL 1	Text
	E-MAIL 2	Text

	BEITR98_99	Zahl
	JAHRB 98-99	Ja/Nein
	BEITR 99-00	Zahl
	JAHRB 99-00	Ja/Nein
	BEITRAG 00-01	Zahl
	JAHRB 00-01	Ja/Nein
	BEITRAG 01-02	Währung
	JAHRB 01-02	Ja/Nein
	BEITRAG 02-03	Währung
	JAHRB 02-03	Ja/Nein
	BEITRAG 03-04	Währung
	JAHRB 03-04	Ja/Nein
	BEITRAG 04-05	Währung
	JAHRB 04-05	Ja/Nein
	ADRESSE ALT	Text
	Anmerkung JB	Text

Ebenso wurde mit einem einzigen, alles umfassenden Formular gearbeitet.

LFD NR	<input type="text" value="1"/>	<input type="text" value="1219770000001"/>		
Beitrag 05 / 06:	<input type="text"/>	<input type="checkbox"/> JB 05/06	Anmerkung JB:	<input type="text"/>
BERUFTITEL	<input type="text"/>	Berufsbezeichnung	<input type="text"/>	Klasse: <input type="text" value="M77"/>
AKADTITEL	<input type="text" value="Ing."/>	FAMNAME	<input type="text" value="Ahrer"/>	VORNAME <input type="text" value="Ernst"/>
STRASSE	<input type="text"/>			
PLZ	<input type="text"/>	ORT	<input type="text"/>	
BUNDESLAND	<input type="text"/>	LAND	<input type="text"/>	
Adresse alt:	<input type="text" value="1212 Wien, Am Bahnhof 4/4/22;"/>			
E-Mail 1:	<input type="text"/>	Anmerkung	<input type="text"/>	
E-Mail 2:	<input type="text"/>			

	Beitrag	JB
98 / 99	<input type="text"/>	<input type="checkbox"/>
99 / 00	<input type="text" value="100"/>	<input type="checkbox"/>
00 / 01	<input type="text"/>	<input type="checkbox"/>
01 / 02	<input type="text"/>	<input type="checkbox"/>
02 / 03	<input type="text"/>	<input type="checkbox"/>
03 / 04	<input type="text"/>	<input type="checkbox"/>
04 / 05	<input type="text"/>	<input type="checkbox"/>

## 16.2. Nötige Änderungen an der Originaldatenbank

Um die Einhaltung der Normalformen zu gewährleisten, schafften wir ein neue Datenbankstruktur (siehe Punkt 13). Weiters wiesen die Daten der Absolventendatenbank im Urzustand viele kleine Fehler auf. So standen z.B. vereinzelt die Orts- und Straßenbezeichnung im Feld der PLZ. Um also die Daten übernehmen zu können, musste diese in Handarbeit verbessert werden. Dabei waren folgende Punkte vorrangig.

Adressen durchsehen:

- Straße und Hausnummer in der Spalte Straße (wobei Straße vor HNr)
- Daten der Spalte Ort gehen verloren! D.h. dass sämtliche Zusatzbezeichnungen und Hausnummern in die Spalte Straße oder einer eigenen müssen.
- ACHTUNG: Für manche PLZ gibt es mehrere Ortsbezeichnungen!
- Anmerkung ist in der neuen DB in 2x Bemerkung (verstorben, verzogen) gegliedert. -> suchen nach Wortlaut „\*storb\*“. Andere Anmerkungen kommen in Sonstiges.
- Bei alter Adresse nur eine Adresse möglich (wird Ursprungsadresse), wobei Ort und Straße durch Beistrich getrennt sind.
- Für nicht besprochene Sonderfälle bitte eine Liste erstellen in der die Erlagscheinnummer und die Spaltenbezeichnung angeführt werden.

Während dem Schreiben der Programme ergaben sich noch folgende vorzunehmende Änderungen:

1. Fürs Einfügen der ADRESSE\_ALT muss folgende Formatierung der Adressen vorliegen:  
PLZ Ort, Strasse; (Bsp: 8656 Hilgertshausen, Parkstraße 4;)  
ACHTUNG: 3 od 4 ausländische Adressen!
2. Im Feld Bundesland stehen in der originalDB Postleitzahlen – diese händisch in die Tabelle Ort nachtragen
3. Tabelle Absverein auf db umbenennen
4. Spaltenüberschrift ERLAGSCHEIN-NR auf ERLAGSCHEIN\_NR ändern!
5. EMAIL1 auf email umbenennen – EMAIL2 wird nicht eingelesen!
6. 89 Einträge der Schüler, die erst die Fachschule und dann die Höhere absolviert. AbsHöhere händisch ergänzen.
7. Bei allen Beitrag- und Jahrfeldern die Bindestriche gegen Unterstriche tauschen und Abstand zwischen Wort und erster Zahl entfernen. Ab 00\_01 bei BEITRAG ergänzen.



8. Auch zwischen „Anmerkung“ und „JB“ ein Unterstrich.
9. Zwischen ADRESSE und ALT einen Unterstrich hinzufügen
10. Neue Spalten in der db: PLZ\_ALT (text), ORT\_ALT (text);  
STRASSE\_ALT (text)

### 16.3. Programme zum Einlesen der Daten

The image shows a vertical stack of buttons on a dark red background. The top button is labeled 'Stammdaten' and has a dashed border. Below it are six more buttons, each with a solid border and a light beige background. The labels on these buttons, from top to bottom, are: 'absdaten', 'email', 'beiträge', 'adtadr', 'ort', and 'uradr'.

Um das Einlesen der Daten schrittweise ablaufen zu lassen, erstellten wir das Formular Daten einlesen. Dieses Formular ist nicht mit dem Hauptformular verbunden, da es nicht kontinuierlich sondern nur einmal benötigt wurde. Deswegen wird es zum Abschluss der Diplomarbeit gemeinsam mit der alten Tabelle und dem alten Formular in einer eigene Datenbank abgespeichert.

Das Formular Daten einlesen beinhaltet je zu schreibender Tabelle einen Button.

Dadurch, dass jede Tabelle einzeln geschrieben wird, ist es möglich den Vorgang gut zu kontrollieren.

### 16.3.1. TABELLE STAMMDATEN:

Die Tabelle Stammdaten ist in der neuen Datenbank die Haupttabelle. Sie beinhaltet die personenbezogenen wichtigsten Daten. Bei der Übernahme der alten Daten gab es keinerlei Probleme, da diese alle im Format Text gespeichert sind.

### 16.3.2. TABELLE ABSOLVENTENDATEN:

Die Erstellung dieser Tabelle erwies sich als schwieriger, da es möglich ist, dass eine Person sowohl einen Fachschul- als auch einen HTL-Abschluss hat. Weiters mussten die Abteilung, die Klasse und der Jahrgang aus einer eigens entwickelten Zeichenfolge ausgelesen werden. Der Eintrag war z.B. folgendermaßen aufgebaut: M77a. M für die alte Abteilungsbezeichnung Maschinenbau, 77 für das Abschlussjahr 1977 und für die Klasse. Weiters beinhaltet der Datenstand der Absolventendatenbank alle je da gewesenen Abteilungsbezeichnungen.

#### *Finden der Fachschüler und Vergabe der entsprechenden Nummer*

```
KlBezeich = Mid(db!ERLAGSCHEIN_NR, 1, 2)
If KlBezeich = 11 Then
    Abteilung = "MF"
    absch!AbsKlasse_Fachschule = Abteilung & Klasse
    absch!AbsJahr_Fachschule = Mid(db!ERLAGSCHEIN_NR, 3, 4)
ElseIf KlBezeich = 14 Then
    Abteilung = "EF"
    absch!AbsKlasse_Fachschule = Abteilung & Klasse
    absch!AbsJahr_Fachschule = Mid(db!ERLAGSCHEIN_NR, 3, 4)
ElseIf KlBezeich = 21 Then
    Abteilung = "FET"
    absch!AbsKlasse_Fachschule = Abteilung & Klasse
    absch!AbsJahr_Fachschule = Mid(db!ERLAGSCHEIN_NR, 3, 4)
ElseIf KlBezeich = 22 Then
    Abteilung = "FMBF"
    absch!AbsKlasse_Fachschule = Abteilung & Klasse
    absch!AbsJahr_Fachschule = Mid(db!ERLAGSCHEIN_NR, 3, 4)
Else
```

#### *Finden und anschließendes Ergänzen des Abschlussjahrs*

```
abschl.MoveFirst
db.MoveFirst
Do While Not db.EOF
    strErlNr = db!ERLAGSCHEIN_NR
```

```
abschl.FindFirst "[ErlagscheinNr] Like '*' & strErlNr & '"
If Not (IsNull(db!ABSJAHR1)) Then
    Abteilung = Mid(db!ABSJAHR1, 1, 1)
    Klasse = Mid(db!ABSJAHR1, 4, 1)
    If IsNull(Klasse) Then
        abschl.Edit
        abschl!AbsKlasse_Hoehere = Abteilung & Klasse
        abschl.Update
    Else
        abschl.Edit
        abschl!AbsKlasse_Hoehere = Abteilung
        abschl.Update
    End If

    Jahrgang = Mid(db!ABSJAHR1, 2, 2)
    If Jahrgang <= 99 Then
        abschl.Edit
        Jahr = "19" & Jahrgang
        abschl!AbsJahr_Hoehere = Jahr
        abschl.Update
    Else
        abschl.Edit
        Jahr = "20" & Jahrgang
        abschl!AbsJahr_Hoehere = Jahr
        abschl.Update
    End If

Else
End If

abschl.MoveNext
db.MoveNext
Loop
```

### 16.3.3. TABELLE E-MAIL:

Beim Schreiben dieser Tabelle gab es keinerlei Probleme, da die gesamten Tabelleninhalte unverändert übernommen werden konnten.

```
db.MoveFirst
Do While Not db.EOF

    If db!email = 0 Then
    Else
        email.AddNew
        email!ErlagscheinNr = db!ERLAGSCHEIN_NR
        email!E_Mail = db!email
        email.Update

    End If

    db.MoveNext
Loop
```

### 16.3.4. TABELLE BEITRÄGE:

Früher wurden für jedes Jahr jeweils ein Beitragsfeld und ein Jahresfeld angelegt. Es standen also für jeden Absolventen pro Jahr einmal Beitrag und einmal Jahr, egal ob diese Felder benötigt wurden oder nicht. Dies hatte einen immensen Verbrauch an Speicherplatz zur Folge. Deswegen ist die Tabelle Beiträge der neuen Datenbankstruktur so aufgebaut, dass das Jahr als Fremdschlüssel dient. Es sind also nur jene Personen in der Tabelle enthalten, die im fraglichen Jahr einen Beitrag an den Absolventenverein entrichteten. Das heißt, dass beim Einspielen der Daten auch nur jene Personen in die Tabelle Beiträge übernommen werden, die mindestens einmal spendeten.

Weiters wird bei jedem Datensatz, egal ob ein Jahrbuch gefordert wurde oder nicht, das Feld verschickt auf true gesetzt, da die Funktion, die die Etiketten für den Jahrbuchversand abrufen, die Einträge sucht bei denen ein Jahrbuch gewünscht wird und das Feld Verschickt noch auf false steht.

```
If IsNull(db!BEITR98_99) Then
Else
    btjb.AddNew

    btjb!ErlagscheinNr = db!ERLAGSCHEIN_NR
    btjb!Jahr = 1998
    btjb!Beitrag = db!BEITR98_99
    btjb!Jahrbuch = db!JAHRB98_99
    btjb!Anmerkung = db!Anmerkung_JB
    btjb!Verschickt = True

    btjb.Update
End If
```

### 16.3.5. TABELLE AKTUELLE ADRESSE:

Die Adressfelder, die in der alten Datenbank den Feldern PLZ, ORT, STRASSE, BUNDESLAND und LAND entsprachen, stehen für die zu letzt bekannte (und damit aktuelle) Adresse der Absolventen. Sie werden also in die Tabelle Aktuelle Adresse aufgenommen. Wobei das Feld LAND-PLZ nicht mehr benötigt wird. Beim Schreiben der Tabelle wird abgefragt, ob die jeweilige PLZ bereits in der Tabelle Ort vorhanden ist. Wenn das nicht der Fall ist, wird sie in diese aufgenommen. Das Anfügen der dazu gehörigen Ortsbezeichnung erfolgt erst später. Ist das Feld leer, so wird es mit 0 ergänzt. Null steht für die Ortsbezeichnung Unbekannt.

```
db.MoveFirst
Do While Not db.EOF
    aktadr.AddNew

    aktadr!ErlagscheinNr = db!ERLAGSCHEIN_NR
    aktadr!Strasse = db!Strasse
    aktadr!PLZ = db!PLZ
    aktadr!Bundesland = db!Bundesland
    aktadr!Land = db!Land
    aktadr!Bemerkung = db!Anmerkung

    aktadr.Update

    aktadr.FindFirst "[ErlagscheinNr] Like ErlagscheinNr and PLZ Is Null"
    If IsNull(aktadr!PLZ) Then
        aktadr.Edit
        aktadr!PLZ = 0
        aktadr.Update
    Else
        End If

    db.MoveNext
Loop
```

### 16.3.6. TABELLE ORT:

Zuerst wird die Tabelle Ort nochmals durchlaufen. Wenn ein PLZ leer geblieben sein sollte, so wird es jetzt mit 0 ergänzt. Anschließend sucht das Programm Datensätze bei denen das Feld PLZ ausgefüllt und das Feld Ort leer ist. Wenn dies der Fall ist, sucht das Programm in der alten Tabelle nach dieser PLZ. Hat es sie gefunden, so wird das Feld Ort in der Tabelle Ort mit der Ortsbezeichnung aus der alten Tabelle gefüllt.

```
ort.FindFirst "[PLZ] Like '*' & strPLZ & '' and Ort Is Null"
If IsNull(ort!ort) Then
    ort.Edit
    ort!ort = db!ort
    ort.Update
Else
    End If
```

### 16.3.7. TABELLE URSPRUNGSADRESSE:

In der alten Datenbank ist die bekannte Adresse immer in den Adressfeldern gestanden. Wurde eine neue Adresse bekannt gegeben, so kam die vorherige in das Feld ADRESSE\_ALT. Diese war folgendermaßen formatiert: PLZ Ort, Strasse; (Bsp: 8656 Hilgertshausen, Parkstraße 4;)

Um also diese Adressdaten in die Tabelle ursprüngliche Adresse schreiben zu könne, muss mit Hilfe von Befehlen gezielt PLZ, Ort und Strasse werden. Um dies leichter erreichen zu können, werden für die eben angesprochenen Attribute jeweils Spalten in der alten Tabelle geschaffen und dort hin geschrieben.

```
ort.MoveFirst
db.MoveFirst
Do While Not db.EOF

    If db!PLZ_ALT = 0 Then
    Else
    db.Edit
        komma = InStr(db!ADRESSE_ALT, ",")
        laenge = komma - 6
        db!ORT_ALT = Mid(db!ADRESSE_ALT, 6, laenge)
    db.Update
    End If
    db.MoveNext
Loop
```

Aus diesen Spalten wird anschließend ausgelesen und in die Tabelle ursprüngliche Adressen geschrieben.

```
db.MoveFirst
Do While Not db.EOF
    strPLZ = db!PLZ_ALT

    If db!PLZ_ALT = 0 Then
    Else
        uradr.AddNew
        uradr!ErlagscheinNr = db!ERLAGSCHEIN_NR
        uradr!Strasse Ursprung = db!STRASSE_ALT
        uradr!PLZ = db!PLZ_ALT
        uradr!Bemerkung = db!Anmerkung
        uradr.Update
    End If
```

Anschließend wird noch die Tabelle Ort nach bestehenden PLZ mit fehlenden Ortsnamen durchlaufen und bei Bedarf ergänzt.

## **17. Sicherheit**

### **17.1. Login**

Da die Daten in der Absolventendatenbank dem Datenschutzgesetz unterliegen, ist notwendig zu gewährleisten, dass Personen, die nicht dazu berechtigt sind, auch nicht auf die Datenbank zugreifen können. Deshalb ist die Datenbank mit einem Passwort geschützt. Dieses Passwort sollte nur dem Verwalter der Datenbank bekannt sein und muss bei jedem Öffnen der Datenbank eingegeben werden.

Weiters wurden der Datenbank Benutzerkonten hinzugefügt. In die Datenbank kann man sich als Administrator und als User einloggen. Wobei beiden Benutzerkonten verschiedene Rechte vergaben wurden.

Der Datenbankverwalter soll auch als einziger imstande sein sich als Administrator einzuloggen. Der Administrator genießt alle Rechte. Er kann die Datenbank exklusiv öffnen, das heißt, er ist in der Lage das Passwort zu ändern. Weiters kann er alle Tabellen, Formulare, Makros, Abfragen und Berichte nicht nur öffnen und ausführen, sondern auch deren Entwürfe lesen und ändern und Daten aktualisieren, einfügen und löschen.

Der User hingegen, kann die Datenbank nicht exklusiv öffnen. Er hat auch nicht das Recht Entwürfe zu lesen oder zu ändern. Er darf auch keine Daten löschen. Dieses Benutzerkonto ist zu benutzen, wenn zum Beispiel jemand den Administrator unterstützt, indem er die Daten der eingezahlten Erlagscheine eingibt.

### **17.2. Autoexec**

Ein Autoexec wird beim Öffnen des Programms automatisch abgerufen. Es erstellt eine Sicherungsdatenbank. Das heißt es werden jedes mal beim Öffnen der Datenbank alle Tabellen kopiert und abgespeichert. Wenn es bei der Arbeit mit der Datenbank also einmal zu Fehlern kommen sollte, muss man alle Funktionen der Datenbank schließen. Anschließend werden die Tabellen der geöffneten Datenbank durch die der Sicherungsdatenbank ersetzt.

Weiters empfiehlt es sich die Datenbankstände regelmäßig zu kopieren und anschließend namentlich mit Datum zu versehen.

## **18. Backend & Frontend**

Es war bis April die Teilung der Datenbank in ein Backend und ein Frontend geplant. Ein Backend beinhalten die Tabellen und liegt auf einem Server, um für mehrere User zugänglich zu sein. Das Frontend umfasst alle Formulare, Abfragen, Berichte und sonstige Funktionen. Es am PC des Client installiert und greift auf das Backend zu.

Da wir im April von Herrn Schneider erfuhren, dass er die Verwaltung des Absolventenvereins abgibt, hat die Teilung in ein Back- und ein Frontend zur Zeit keinen Sinn, da noch kein Nachfolger bekannt ist und wir dementsprechend nicht wüsten auf welchen User wir das Frontend installieren sollen. So haben wir uns mit Herrn Dietrich dazu entschieden, die Datenbank nicht zu Teilen, sondern sie als gesamtes auf das Login des Absolventenvereins zu stellen. Eine nachfolgende Trennung der Programmteile ist weiterhin jederzeit möglich.

## **19. Resümee**

Die Diplomarbeit konnte trotz spätem Programmierbeginn erfolgreich durchgeführt werden. Auch wenn es zum Schluss der Arbeit Probleme gab, so konnten wir unsere Ideen und die Wünsche unserer Betreuungslehrer gut in die Tat umsetzen. Da uns unsere Betreuungslehrer gut auf diese Diplomarbeit vorbereitet haben und uns im Vorhinein klar ihre Wunsch äußerten, gelang es uns Struktur in unsere Aufgabenplanung zu bringen. Da wir keine gezielte Meilensteinplanung hatten, bekamen wir neben sehr vielen Prüfungen zum Schluss Zeitprobleme. Also mussten wir den Arbeitsdruck von uns selbst erhöhen und sehr viel Arbeit und Zeit investieren.

Letztendlich müssen wir sagen, dass uns das arbeiten immer leichter fiel, da wir sehr viel gelernt haben. Wir hoffen dieses erworbene Wissen auch in Zukunft mit Freude einsetzen zu können.



## 20. Arbeitslisten

**Bianca Ecker:**

Datum	Stunden	Arbeit	Quelle
14.09.2006	1	Besprechung mit Herrn Schneider	
19.09.2006	1,5	Einarbeitung in Projektthema und Findung von Lösungsideen	
20.09.2006	2	Genauere Projekterklärung – Aufgabenstellung – von Herrn Schneider	
25.09.2006	1	Besprechung mit Herrn Dietrich, Lösungs- und Ideenfindung, MS Data Engine	
25.09.2006	2	Ausarbeitung der Unterlagen für LSI	
27.09.2006	1,5	Gemeinsames Ergänzen für unsere Unterlagen für LSI	
27.10.2006	1	Projektbesprechung mit Herrn Pracherstorfer, Projekterklärung, nützliche Unterlagen für unsere Dokumentation	
16.11.2006	1,5	Einarbeiten in bestehende Tabellenstruktur	
22.11.2006	2,5	Entwurf einer neuen Formular- und Tabellenstruktur	Binf-Unterlagen 3.KI
24.11.2006	3,5	Absprechen der Tabellenstruktur (Normalformen) mit Michi, 1 Std. Besprechung mit Herrn Pracherstorfer, 1 Std. Besprechung mit Herrn Schneider	
27.11.2006	1,5	Zusammenschreiben der neuen Tabellenstruktur, Schreiben sämtlicher Besprechungsprotokolle	
01.12.2006	1	Durchgehen der Formularstruktur und Planung für weiteres Vorgehen mit Herrn Pracherstorfer, anschließend Durchgehen der geplanten Formularstruktur mit Herrn Schneider und Etwaiges ändern	
13.12.2006	2,5	Erstellen einer Musterdatenbank: Tabellen mit Mustereinträgen, mit Formularen angefangen	Binf-Unterlagen 3.KI

14.12.2006	2,5	Erstellen der Formulare (inkl. Layout), Makros, Sicherungsdatenbank	
		Planung der Projektausstellung am ToT, Überarbeitung der bestehenden Formulare und Tabellen. Nachdem einige Datenformate falsch waren und durch die bereits bestehende Verknüpfung nicht oder nur schwer änderbar waren, erstellte ich die Tabellenstruktur und die Formulare der Musterdatenbank mit neuem Layout wieder.	
23.01.2007	4,5	Vornahme kleiner Layoutänderungen an den Formularen, Erstellung der Makros. Absprache mit Herrn Schneider und Herrn Grossberger über die Inhalte und Materialien wegen der Projektausstellung am ToT und dem weiteren Projektverlauf.	
24.01.2007	4	Heraussuchen, Bearbeiten und Drucken der Ausstellungsinhalten, Besorgung der Karteikarten des Abs.-Vereins und des alten D-Base-Rechners – Umgang mit diesem.	Herr Schneider
25.01.2007	3	Projektbesprechung mit Herrn Pracherstorfer, Herrn Grossberger, zeitweilig auch Herrn Dietrich: Kontrolle der bestehenden Musterdatenbank (Tabellen, Beziehungen, Formulare, Funktionen)	
02.02.2007	1	Besprechung mit Herrn Schneider und Herrn Grossberger: Kurze Vorstellung der Musterdatenbank; Festlegung des nächsten Besprechungstermins.	
02-02-2007	0,5	Treffen mit Herrn Schneider und Herrn Grossberger: Genaues Durchbesprechen und Erklären der Musterdatenbank, Wiederholung der bereits ausgemachten Wünsche und Richtlinien an den Umgang mit der Datenbank und der neuen Datenverwaltungsideen. Festlegen der gewünschten Abfragen.	
06.02.2007	3	Zusammenfassen der handgeschriebenen Notizen und formulieren der Problemstellungen und Anforderungen an das Programm	
07.02.2007	2	Anlegen der neuen Datensätze 12345 bis 12348 um einfacher testen zu können. Das Unterformular Beiträge laut den Wünschen Herrn Schneiders sortiert. Das Formular e-mail neu gestaltet und Verknüpft, ein Unterformular e-mail geschaffen und in absteigende Reihenfolge gebracht.	
08.02.2007	2	Sichern der und Sortieren der unterschiedlichen Datenstände. Ändern des Datenformates der beiden Schlüssel Erlagscheinnummer und PLZ von Zahl auf Text, um VBA-Funktion zu ermöglichen bzw. auf das canadische PLZ-Format einzugehen. Anschließend Herstellen einer referenziellen Integrität und neuer Beziehungen sowie grobes Austesten der bestehenden Datenbankfunktionen. Habe versucht über die Methode Find bei Eingabe der der lfd Nummer die dazugehörige 12-stellige Erlagscheinnummer zu finden – leider gescheitert.	Hilfe VBA
20.02.2007	3		

22.02.2007	1	Besprechung mit Herrn Schneider über die zu ändernden Daten in der Originaldatenbank, um 3.NF und richtige Datenformate zu gewährleisten (siehe Änderungen original DB.doc)	
22.02.2007	2	Neues Formular Aktionen und Schaltfläche „Absolventen hinzufügen“ angelegt. Mit Filedialog zum Auffinden des vom Sekretariat abgefragten Excelldokuments.	
23.02.2007	3	Weiterentwickeln des Filedialoges für das automatische Hinzufügen der neuen Absolventen. Besprechung mit Herrn Dietrich. Lösungsfindung, um die Absolventendaten in den Formularen über die vierstellige laufende Nummer Finden zu können (findfirst, like und ungebundenes Feld).	<a href="http://www.office-loesung.de">http://www.office-loesung.de</a>
27.02.2007	3	Weiterentwickeln der Lösung für das Auffinden der Absolventen über die lfd. Nummer, d.h. richtiges Einfügen des strLfdnr – Strings. Anschließend Schaffung neuer Strings (für jedes Textfeld im Hauptformular) und Abrufen dieser in einer If-Schleife.	
02.03.2007	1	Besprechung mit Herrn Pracherstorfer über den aktuellen Stand der Diplomarbeit, die Anforderungen an die Dokumentation und die Vorabgabe- und Abgabetermine. Weiters sprachen wir noch über die Konsequenzen, wenn das Projekt nicht rechtzeitig fertig wird.	
08. 03.2007	5	Habe meine letzten Arbeiten und Besprechungen nachgetragen und mit Herrn Großberger über die nächsten Termine für das Fertigbekommen der Diplomarbeit gesprochen (telefonisch). Ich habe es endlich geschafft im Hauptformular die betreffenden Daten, eines durch die laufende Nummer abgefragten Datensatzes, in die jeweiligen Felder einzutragen und habe dies anschließend gründlich getestet. Habe mit dem Formular Beiträge angefangen. Dabei fielen mir einige Probleme auf: <ul style="list-style-type: none"> <li>- Das automatische Abrufen von Daten aus anderen Formularen ist mir nicht möglich. Ich wollte dies beim Öffnen des neuen Formulars über Ereignis - beim Klicken abrufen. Dass ist aber nicht möglich, weil sich dort bereits das Makro öffnen befindet.</li> <li>- Ich kann die Tabellendaten zwar ins Formular übernehmen, darf sie aber nicht verändern oder löschen!</li> <li>- Brauche eine Abfrage, die das aktuelle Jahr ermittelt – für die veränderlichen Beiträge usw. Jahr aus Betriebssystem! -&gt; vergangene Jahre können nicht mehr verändert werden.(SQL-Abfrage)</li> </ul>	
09.03.2007	6	Ich habe die 3 Felder PLZ, Ort und Strasse im Hauptformular erstellt und die Anweisungen zum automatischen Einfügen der lt. Lfd. Nr. ermittelten Daten geschrieben. Weiters habe ich eine Anweisung für das Unterformular Beiträge erstellt. Besprechung mit Herrn Dietrich über die über die oben erwähnten Probleme. Die Makros werden gelöscht und durch VBA-Code ersetzt. Das Sperren der Beiträge der vergangenen Jahre ist möglich indem ich in einer	

Schleife angebe, dass jedes Jahr, das kleiner als das aktuelle ist, nicht mehr verändert werden darf und sperren der Tabellen für die User bei der Rechtevergabe. (aktuelle Jahr über Befehl DATE aus Betriebssystem). Wir haben leider noch keine Lösung gefunden, um die Tabellendaten über die Eingabe in den Formularen zu erlauben.

Ich habe die Makros gelöscht und stattdessen VBA-Funktionen erstellt. Weiters habe ich die Formulare umformatiert, die Reihenfolge der Ausgabe im Unterformular Beiträge absteigend sortiert und zuletzt die Reihenfolgenpositionen der Felder sortiert. Daraufhin funktionierte aber die Abfrage der Tabellendaten nicht mehr! Habe noch nicht herausgefunden warum? Ich habe versucht beim Weitergehen vom Haupt- in das Beitragsformular, die aktuellen Daten im Hauptformular abzufragen. Dies funktioniert leider noch nicht.

- |            |     |  |
|------------|-----|--|
| 12.03.2007 | 1   | Dokumentation der geleisteten Arbeit der letzten Tage.   |
| 13.03.2007 | 2   | <p>Arbeit mit Herrn Pracherstorfer: Haben nach der Ursache gesucht, warum sich die Tabellendaten in den Formularen nicht ändern oder löschen ließen. Weiters funktionierte meine Abfrage nicht mehr, da ich die Reihenfolgenpositionen in den Formularen veränderte. Die Ursache beider Probleme war, dass das Feld „Erlagscheinnummer“ gebunden war.</p> <p>Kann die Daten in den Formularfeldern jetzt zwar ändern, jedoch werden die Änderung nicht in die Tabelle übernommen.</p>  |
| 15.03.2007 | 4,5 | <p>1 Stunde Besprechung mit Herrn Dietrich und Herrn Pracherstorfer über den Fortgang der Diplomarbeit. Ich soll meinen Teil fertig machen, egal ob der zweite Teil geleistet wird oder nicht. Weiters sprachen wir über einen Termin (12.04), um uns die Vorabgabe der Diplomarbeitdokumentation nach den Osterferien gemeinsam durchzusehen, das Backend und wann wir es erstellen (nach Osterferien – überarbeitete DB-original von Sn)</p> <p>1 Stunde Lösungsfindung mit Herrn Dietrich. Wir suchten eine Funktion mit der sich die geänderten Datensätze aus den Formularen gleich nach ihrer Aktualisierung in die Tabellen (richtige Datensatzposition !!!) übernehmen lassen. Bisher ohne Erfolg.</p> <ul style="list-style-type: none"> <li>• LfdNr wird zwar ins Formular Beiträge übernommen, aber es wird nicht automatisch aktualisiert.</li> <li>• Abfragen erstellt, um DS ins Formular BTJB zu schreiben. Jedoch passen die Beiträge noch nicht. Es soll das aktuelle Jahr aus BS ausgelesen werden, in das Formular geschrieben werden (= neuer DS in Tabelle erstellen) und für das aktuelle Jahr sollen JB, Verschickt, usw angelegt werden.</li> <li>• Abfrage für das automatische Einlesen der Daten lt. Erlagscheinnummer im Hauptformular.</li> </ul> |
| 16.03.2007 | 4,5 | <p>Habe das Schreiben der DS in den Formularen Ursprungsadresse und e-mail geschrieben. Habe bemerkt, dass der anfangs dastehende DS beim Schließen des Formulars mit dem abgefragten DS überschrieben</p>   |

wird – und zwar bei allen programmierten Abfragen (d.h. ab F\_Beiträge). Habe den Fehler gesucht, aber leider noch nicht gefunden.  
30 min arbeiten mit Herrn Pracherstorfer. Suche des Fehlers, warum die DS überschrieben werden. Ausbessern und Ergänzen der überschriebenen oder fehlenden DS, Bindung aller Formularfelder lösen, Programmverbesserungen bei Tabelle aktuelle Adresse, Ursprungsadresse und bei e-mail. Fehler bei der Anzeige des Unterformulars e-mail noch nicht gefunden.

20.03.2007

- 6 Habe mit Unterstützung von Herrn Dietrich einige neue Funktionen erstellt:
- Nach dem Abfragen der Formulardaten wird über die Funktion Bookmark die Datensatzposition ermittelt
  - Das Rückschreiben der im Formular veränderten Daten erfolgt über das Ereignis nach Aktualisierung des jeweiligen Formularfeldes -> über Unterprogramm aufrufen. Muss noch für alle anderen Felder gemacht werden.
  - Das Ermitteln des aktuellen Jahres aus Betriebssystem und Einfügen ins Formular. Gehört noch in Tabelle geschrieben.

www.office-  
loesung.com

Besprechung mit Herrn Schneider über den aktuellen Stand meiner Arbeit. Habe ihm alle Funktionen vorgeführt – keine Verbesserungsvorschläge. Es ist o.k., dass beim Eingeben einer ErlNr im F\_BTJB ein neuer DS mit dem aktuellen Jahr angelegt wird. Es soll jedoch auch möglich sein, im nächsten Jahr beim aktuellen Jahr noch etwas zu verändern, da manchmal verspätet Beiträge eintreffen. Dies soll möglich werden indem der Administrator das Recht hat im Hauptformular\_Unterformular\_Beiträge die alten DS zu ändern.

Prozeduren für das automatische Einlesen der Abschlussdaten und Rausschreiben aller Daten sämtlicher Formulare über Unterprogramme (ohne Unterformulare). Austesten jeder einzelnen Funktion. Löschen, Anpassen und Testen sämtlicher Messageboxes. Umgestaltung der Feldbezeichnungen und Löschen des Geschlechts im HF.

Dokumentation der geleisteten Arbeit.

23.03.2007

- 1 Besprechung mit Herrn Pracherstorfer und Herrn Grossberger über Fortgang des Projekts und der Probleme bei mir und Herrn Grossberger beim Abfragen der E-Mail-Daten. Da mit der Tabelle E-Mail alles passt, müsste das Problem bei der Namensgebung der Tabelle E-Mail liegen.

26.03.2007

- 5 Habe mit der Programmierung für das automatische Einlesen der neuen Absolventen begonnen. Die gewünschte Exceldatei wird über Filedialog ausgewählt und soll anschließend (kurzzeitig) als T\_AbsNew in

VBA Hilfe

der DB abgespeichert werden. Da mir die Option zur Erstellung einer Importspezifikation fehlt, funktioniert das leider noch nicht automatisch. Über Mid-Funktion und Schleifen wird bereits eine neue Erlagscheinnummer generiert, bei der jedoch noch die laufende Nummer fehlt, da ich (auch in der Hilfe) keine Funktion finden konnte, die die größte Zahl sucht. Bisher wird die unvollständige Erlagscheinnummer noch in die T\_AbsNew geschrieben. (Diese soll nach dem Auslesen und Umschreiben der Daten in die jeweiligen Tabellen, wieder automatisch gelöscht werden.)  
Anschließend habe ich den Artikel für den Jahresbericht verfasst.

27.03.2007	<p>5 ✓ Kann die neuen Absolventen jetzt auf einen Knopfdruck importieren:</p> <ul style="list-style-type: none"> <li>• Habe die „absnew_importspez“ erstellt</li> <li>• Dies war möglich, weil ich das Dateiformat auf Text (*.csv) umgespeichert habe</li> <li>• <b>!!! WICHTIG:</b> Das Dateiformat muss immer csv sein. Die Datei wird im Filedialog auch nur gefunden, wenn das Suchformat auf *.csv + Enter eingestellt wird. <b>!!!</b> MsgBox gibt dies auch aus.</li> </ul> <p>✓ Für das Einlesen der neuen Absolventen fehlt noch die lfdnew. Diese wird ermittelt indem temporär eine T_lfdNr erstellt wird (bereits erledigt). In diese Tabelle werden sämtliche aus der T_stamm ausgelesenen (Right-Befehl) lfdNr geschrieben und anschließend absteigend sortiert. Die erste (= höchste) lfdNr wird ausgelesen und jeweils um 1 erweitert um die lfdnew zu schaffen. Daten werden in die T_lfdnr geschrieben, müssen noch sortiert werden!</p> <p>✓ Habe den Fehler bei dem Unterformular E-Mail noch immer nicht entdeckt.</p> <p>✓ Um beim Anlegen eines neuen Beitrags für 2007 alte Daten nicht zu überschreiben, müssen in die Schleife addnew sämtliche Daten mitgenommen werden, wie der Primärschlüssel Erlagscheinnr. Sie sollen aus dem Formular ausgelesen werden und als neuer Datensatz in die T_Beiträge geschrieben werden. Funktioniert noch nicht!</p> <p>✓ Habe den 300 Wörter langen Artikel für den Jahresbericht verfasst.</p>	<a href="http://www.access-im-unternehmen.de">http://www.access-im-unternehmen.de</a>
28.03.2007	<p>4</p> <ul style="list-style-type: none"> <li>• Habe den Dateityp (im Feld Dateiname) für das Importieren der Absolventendatei auf .csv umgestellt.</li> <li>• Die T_lfdnr wird nun auch automatisch absteigend sortiert, so ist die größte lfdNr immer an erster Stelle. Es gibt jedoch beim Auslesen noch einen Fehler, d.h. er schreibt nicht erste lfdNr + 1 sondern nur 1. -&gt; liest nicht oder falsch aus!!! Es gibt wahrscheinlich einen Fehler in der Schleife.</li> <li>• Habe mich mit dem Problem des Überschreibens durch den neun Beitragsdatensatz beschäftigt. Hab es leider nicht beheben können.</li> </ul>	
29.03.2007	<p>2 Habe weiter am Fehler bei den neun Beitragsdatensätzen gesucht, alte Messageboxes entfernt und nicht mehr aktuelle Erinnerungskommentare entfernt und neue hinzugefügt. Besprechung mit Herrn Schneider und Herrn Großberger wegen des Artikels für den Jahresbericht und anschließend Foto.</p>	
02.04.2007	<p>2 Habe allgemeine Informationen im Internet gesucht, um sie als Hilfestellung bei dem allgemeinen Teil der</p>	div. Seiten,

Dokumentation zu verwenden (DB, VBA, SQL, NF, MS DE)			hauptsächlich Wikipedia
04.04.2007	3	Habe die Arbeitsdokumentation und das Besprechungsprotokoll auf den aktuellen Stand gebracht und anschließend mit den ersten Kapiteln der Dokumentation begonnen.	
05.04.2007	3,5	Weiterführung der Dokumentation.	
10.04.2007	4,5	Schreiben der Kapitel 10 bis 13.	Binf-Unterlagen 3. Kl. und div. Internetseiten
16.04.2007	3	Weiterprogrammieren am automatischen Hinzufügen der Absolventen; lfd Nr fertig, nun schreiben der Daten in die jeweiligen Tabellen.	
19.04.2007	6	Weiter an der Programmierung zum Importieren der neuen Absolventen gearbeitet. Wieder Probleme mit Update! Dokumentation: Visio Grafik Arbeitsteilung verändert, die zwei temporären Tabellen in Dok. ergänzt (neue Beziehungen), die Formularstruktur und das Hauptformular beschrieben, eventuelle Fehler und Fehlermeldungen, Code verbessert, Kommentare, Rechtschreibkontrolle der Dokumentation.	
20.04.2007	7	Weiterdokumentieren, Fehlersuche beim Hinzufügen der Absolventen; Weiterschreiben beim Code für das Schreiben der Absolventen in die Tabellen – Probleme mit PLZ, Ziehen von Sicherheitskopien von allen Dateien der Dipl.Arbeit.	
21.04.2007	3,5	Programmierung für das Hinzufügen neuer Beiträge/JB	
22.04.2007	1,5	Dokumentation	
23.04.2007	7	Abschließen der Programmierung für das Hinzufügen der neuen Absolventen, Beiträge hinzufügen abgeschlossen, Anfang der Programmierung für das Einlesen der originalen Absolventendaten; Besprechung mit Herrn Schneider über die nötigen Änderungen in der originalen AbsDB., Dokumentation der Arbeit.	
24.04.2007	11	Schreiben der restlichen 5 Programme, und einer Liste, die angibt was in der originalDB durchgesehen und geändert gehört, um das Einlesen der originalen Daten zu gewährleisten. Fehlersuche in der zusammengeführten (Gross' und mein Programmteil) DB.	
25.04.2007	6	Fehlersuche in den Programmen zum einlesen der originalDB. Erweitern und Umschreiben der	

Programme, anschließend testen. Durchsehen der originalDB mit Herrn Schneider um eventuelle Fehler zu finden. Einlesen der DB und gleichzeitig korrigieren der letzten Fehler. Besprechung mit Herrn Dietrich über Backend -> Die DB wird aus wirtschaftlichen Gründen doch nicht zweigeteilt. Arbeitsaufteilung mit Herrn Grossberger.

27.04.2007	1,5	Fehlersuche beim Eintragen der neuen Beiträge.
30.04.2007	3	Korrektur des Fehlers beim eintragen der neuen Beiträge. Fehlersuche im Unterformular e-mail leider erfolglos.
01.05.2007	6,5	Dokumentation der Formulare; Ändern der Kommentare im Code, leichte Codeabänderungen
02.05.2007	7	Erstellung Autoexec, Unterformulare Beiträge und e-mail; Dokumentation; Fehlersuche und Behebung Code; Kommentare
03.05.2007	9	Dokumentation, aktuelle Adressen umprogrammieren, um neue PLZ aufnehmen zu können. Kommentare Code, Beenden Dokumentation
04.05.2007	3	Zusammenfügen der Dokumentationsteile, Layout, Einschicken
<u>Σ 212</u>		



## Michael Großberger:

Datum	Arbeit	Stunden
15.06.2006	Besprechung mit VI. Schneider, Lastenheft/Pflichtenheft	1
07.09.2006	Besprechung mit VI. Schneider, Lastenheft/Pflichtenheft	1
14.09.2006	Besprechung mit VI. Schneider und Bianca über Aufgabenstellung	1
19.09.2006	Erläuterung der Aufgabenstellung durch VI. Schneider, Einführung	2
22.09.2006	Besprechung über Vorgehensweise mit VI. Schneider	2
25.09.2006	Besprechung mit DI Dietrich – Betreuungslehrer, MSDE	1
27.09.2006	Unterlagen für LSI zusammengestellt	2
27.10.2006	Besprechung mit DI Pracherstorfer, Projektunterlagen	1
20.11.2006	Alter DB-Stand, Tabellen, Formulare durchgesehen	3
23.11.2006	DB-Struktur - Konzept	1
24.11.2006	DB-Struktur – Konzept mit Bianca besprochen	2
01.12.2006	Konzept mit DI Pracherstorfer und VI. Schneider besprochen	2
04.12.2006	Tabellen in Access erstellt	2
05.12.2006	Beziehungen durchdacht und erstellt	2
06.12.2006	Datensätze zum Testen eingegeben	2
15.12.2006	Änderung der Beziehungen	1
23.01.2007	Konzept für ToT entwickelt	1
24.01.2007	Planung und Erarbeitung von Anschau material für ToT	2
25.01.2007	Materialien für ToT zusammengestellt	4
26.01.2007	Vorstellung des Projekts am ToT	4
27.01.2007	Vorstellung des Projekts am ToT	4
02.02.2007	Besprechung mit VI. Schneider – aktueller Stand	1
06.02.2007	Besprechung mit VI. Schneider – neue Datenbank, Anforderungen	2
23.02.2007	Anforderungen an Abfragen dokumentiert	1
27.02.2007	Einarbeitung SQL, Unterlagen zusammengesucht	3
01.03.2007	Abfrage Klassenliste begonnen	1
04.03.2007	Abfrage Klassenliste – Probleme mit E-Mail Tabelle	3
07.03.2007	Abfrage Rundschreiben	3
08.03.2007	Änderungen in Tabelle Stammdaten – Feld Geschlecht, ...	2
12.03.2007	Änderungen bei Beziehungen vorgenommen	1
13.03.2007	Dokumentation, Unterlagen für Dokumentation	2
15.03.2007	Etikettenabfrage, Besprechung mit VI. Schneider über Druck	3
18.03.2007	Etikettendruckbericht erstellt, Abfrage Rundschreiben	3
21.03.2007	Abfrage Absolventinnen erstellt, Formularänderungen	2
23.03.2007	Besprechung mit DI Pracherstorfer – Problem mit E-Mail Tabelle	1
24.03.2007	Abfrage Klassenliste fertig, Problem E-Mail gelöst	2

25.03.2007	Klassenlisten Bericht erstellt, Schaltflächen	2
27.03.2007	Exportieren der Klassenliste in Excel, Makro erstellt	1
30.03.2007	Rundschreiben fertig gestellt	3
05.04.2007	Serienbrief erstellt und formatiert	3
07.04.2007	Formularanpassungen	2
11.04.2007	Serienbrief bearbeitet	2
15.04.2007	Summary für Dokumentation geschrieben	1
17.04.2007	MSDE - Einarbeitung, Unterlagen	2
24.04.2007	Zusammenfügen der Datenbanken - Fehlersuche	5
25.04.2007	Fehlersuche, und Ändern der Berichte	6
26.04.2007	Abfrage Versickt	2
27.04.2007	Eingefügte Datensätze überprüft, geändert – Fehler im Code	3
28.04.2007	Dokumentation, Gliederung...	3
29.04.2007	Abfragen überarbeitet	2
30.04.2007	Dokumentation	12
01.05.2007	Benutzerkonten und Passwort festgelegt	2
02.05.2007	Datenbank mit Vl. Schneider durch besprochen, Testen, Änderungen	7
03.05.2007	Dokumentation überarbeitet, abschließende Arbeiten	10
04.05.2007	Dokumentation fertig gestellt, zusammengefügt	4
		Σ 153

## 21. Besprechungsprotokolle

### Projektbesprechungen:

Datum: 14.09.2006

Anwesende: Schneider, Ecker

Thema: **Aufgabenstellung/Projekterklärung**

Grundlegende Erklärungen

- zum Ist-Zustand der Datenbank,
- ihrer Geschichte,
- der Probleme die es derzeit gibt, der bisherigen Funktionen,
- der benötigten Funktionen,
- der Arbeiten, die jährlich für den ABS-Verein verrichtet werden,
- usw.

---

Datum: 20.09.2006

Anwesende: Schneider, Ecker, Grossberger

Thema: **Fragen zur DB und erste Lösungsideen**

Punktuelle Aufarbeitung der von uns bereits überarbeiteten Anforderungsliste von Herrn Schneider (Anforderung Programm.doc). Dass heißt, dass wir bei Unklarheiten zur DB oder der Arbeit mit ihr noch mal nachfragen konnten und wir unsere bereits überlegten Lösungen Herrn Schneider erklärten. In weiterer Folge haben wir auch gemeinsam neue Lösungen erarbeitet.

---

Datum: 25.09.2006

Anwesende: Dietrich, Ecker

Thema: **Lösungs- und Ideenfindung**

Herr Dietrich, der die Projektstruktur bereits durch Herrn Schneider kennt, hat sich unsere Lösungsideen angesehen und mit mir durch besprochen. Weiters empfahl er das Projekt in ein Backend (Server/Administrator) und ein Frontend (Arbeitsoberfläche/Client) zu gliedern. Das Backend wird über MS Data Engine verwirklicht werden.

---

Datum: 27.10.2006

Anwesende: Pracherstorfer, Ecker, Grossberger

Thema: **Projekterklärung/Unterlagen**

Wir haben Herrn Pracherstorfer die Datenbank und die bisherige Arbeit mit ihr, den gewünschten Projektumfang und unsere geplanten Lösungen erklärt. Er gab uns nützliche Materialien wie einen Leitfaden für die Dokumentation und eine Beispieldiplomarbeit.

---

Datum: 24.11.2006

Anwesende: Pracherstorfer, Ecker, Grossberger

Thema: **Tabellenstruktur**

Besprechen und korrigieren der neuen Tabellenstruktur der DB (3. Normalform!).

---

Datum: 24.11.2006

Anwesende: Schneider, Ecker, Grossberger

Thema: **Tabellenstruktur/ToT**

Besprechen der geplanten Tabellenstruktur. Wir hatten auch noch Fragen bezüglich der Sinnhaftigkeit oder der Verwendung mancher Felder. Weiters zeigte uns Herr Schneider auch die früheren Arbeitsmethoden des Absolventenvereins (Karteikarten, D-BASE) -> können wir am ToT präsentieren.

---

Datum: 01.12.2006

Anwesende: Pracherstorfer, Ecker

Thema: **Formularstruktur**

Durchgehen der geplanten Formularstruktur und Meilensteine (ausgearbeitete Tabellen und Formulare, sowie Planung der Abfragen) bis 15. Dezember.

---

Datum: 01.12.2006

Anwesende: Schneider, Ecker

Thema: **Formularstruktur**

Durchgehen der Formularstruktur und Einbringen der praktisch bezogenen Einwände von Herrn Schneider -> dementsprechende Änderung.

---

Datum: 24.01.2007

Anwesende: Schneider, Ecker, Grossberger

Thema: **Projektvorstellung am ToT**

Absprache mit Herrn Schneider und Herrn Grossberger über die Inhalte und Materialien wegen der Projektausstellung am Tag der offenen Tür und dem weiteren Projektverlauf.

---

Datum: 02.02.2007

Anwesende: Pracherstorfer, zeitweise Dietrich, Ecker, Grossberger

Thema: **Kontrolle der neuen Musterdatenbank**

Kontrolle der bestehenden Musterdatenbank

- Tabellen (3. NF eingehalten),
  - Beziehungen (Korrekt und Überblick der Tabelleneinteilung),
  - Formulare (Handlichkeit, vorhandene und fehlende Funktionen),
  - Funktionen (welche)
- 

Datum: 02.02.2007

Anwesende: Schneider, Ecker, Grossberger

Thema: **Musterdatenbank**

Kurze Vorstellung der Musterdatenbank; Festlegung des nächsten Besprechungstermins.

---

Datum: 06.02.2007

Anwesende: Schneider, Ecker, Grossberger

Thema: **Musterdatenbank/Abfragen**

- Genaues Durchbesprechen und Erklären der Musterdatenbank,
  - Wiederholung der bereits ausgemachten Wünsche und Richtlinien an den Umgang mit der Datenbank und
  - der neuen Datenverwaltungsideen.
  - Festlegen der gewünschten Abfragen.
  - Hinweise Herrn Schneiders auf eventuelle Probleme mit dem Datenformat bei der Übernahme der Daten.
  - Erweiterung des Aufgabenfeldes um automatisches Importieren von e-mail-Adressen auf mailing-Programme
- 

Datum: 22.02.2007

Anwesende: Schneider, Ecker

Thema: **zu Ändernde Daten**

Besprechung mit Herrn Schneider über die zu ändernden Daten in der Originaldatenbank, um 3.NF und richtige Datenformate zu gewährleisten (siehe Änderungen original DB.doc)

Kurze Vorstellung der Musterdatenbank; Festlegung des nächsten Besprechungstermins.

---

Datum: 23.02.2007

Anwesende: Dietrich, Ecker

Thema: **Lösungsfindung laufende Nummer**

Besprechung mit Herrn Dietrich: Lösungsfindung, um die Absolventendaten in den Formularen über die vierstellige laufende Nummer finden zu können (findfirst, like und ungebundenes Feld).

---

Datum: 02.03.2007

Anwesende: Pracherstorfer, Ecker

Thema: **aktueller Arbeitsstand, Probleme**

Besprechung mit Herrn Pracherstorfer über den aktuellen Stand der Diplomarbeit, die Anforderungen an die Dokumentation und die Vorabgabe- und Abgabetermine. Weiters sprachen wir noch über die Konsequenzen, wenn das Projekt nicht rechtzeitig fertig wird.

---

Datum: 09.03.2007

Anwesende: Dietrich, Ecker

Thema: **Problembehandlung**

Besprechung mit Herrn Dietrich über die über die oben erwähnten Probleme. Die Makros werden gelöscht und durch VBA-Code ersetzt. Das sperren der Beiträge der Vergangenheit ist möglich indem ich in einer Schleife angebe, dass jedes Jahr, das kleiner als das aktuelle ist, nicht mehr verändert werden darf und durch das Sperren der Tabellen für die User bei der Rechtevergabe. (aktuelle Jahr über Befehl DATE aus Betriebssystem). Wir haben leider noch keine Lösung gefunden, um die Tabellendaten über die Eingabe in den Formularen zu erlauben.

---

Datum: 15.03.2007

Anwesende: Dietrich, Pracherstorfer, Ecker

Thema: **Diplomarbeitsfortgang, Termine**

1 Stunde Besprechung mit Herrn Dietrich und Herrn Pracherstorfer über den Fortgang der Diplomarbeit. Ich soll meinen Teil fertig machen, egal ob der zweite Teil geleistet wird oder nicht. Weiters sprachen wir über einen Termin (12.04), um uns die Vorabgabe der Diplomarbeitsdokumentation nach den Osterferien gemeinsam anzusehen, das Backend und wann wir es erstellen (nach Osterferien – überarbeitete DB-original von Sn)

---

Datum: 15.03.2007

Anwesende: Dietrich, Ecker  
Thema: **Problembehandlung**

1 Stunde Lösungsfindung mit Herrn Dietrich. Wir suchten eine Funktion mit der sich die geänderten Datensätze aus den Formularen gleich nach ihrer Aktualisierung in die Tabellen (richtige Datensatzposition!!!) übernehmen lassen. Bisher ohne Erfolg.

---

Datum: 20.03.2007  
Anwesende: Schneider, Ecker  
Thema: **Programmfunktionen**

Besprechung mit Herrn Schneider über den aktuellen Stand meiner Arbeit. Habe ihm alle Funktionen vorgeführt – keine Verbesserungsvorschläge. Es ist o.k., dass beim Eingeben einer ErlNr im F\_BTJB ein neuer DS mit dem aktuellen Jahr angelegt wird. Es soll jedoch auch möglich sein, im nächsten Jahr beim aktuellen Jahr noch etwas zu verändern, da manchmal verspätet Beiträge eintreffen. Dies soll möglich werden indem der Administrator das Recht hat im Hauptformular\_Unterformular\_Beiträge die alten DS zu ändern.

---

Datum: 23.03.2007  
Anwesende: Grossberger, Pracherstorfer, Ecker  
Thema: **Tabelle E-Mail**

Besprechung mit Herrn Pracherstorfer und Herrn Grossberger über Fortgang des Projekts und der Probleme bei mir und Herrn Grossberger beim Abfragen der E-Mail-Daten. Da mit der Tabelle E-Mail alles passt, müsste das Problem bei der Namensgebung der Tabelle E-Mail liegen.

---

Datum: 29.03.2007  
Anwesende: Grossberger, Schneider, Ecker  
Thema: **Bericht Jahrbuch**

Besprechung mit Herrn Schneider und Herrn Grossberger über den Artikel über die Diplomarbeit im Schuljahrbuch. Gemeinsames Durchgehen und Kürzen des Artikels und anschließendes Schießen von Fotos für den Bericht.

---

Datum: 20.04.2007  
Anwesende: Grossberger, Pracherstorfer, Ecker  
Thema: **Programmierung Absolventen hinzufügen, Diplomarbeitsabgabe**

Herr Pracherstorfer half uns bei der Fehlerfindung im Modul abs\_import. Weiters machter er uns noch darauf aufmerksam, dass die praktische Arbeit erst von Herrn Schneider getestet werden muss, bevor wir sie abgeben.

---

Datum: 23.04.2007

Anwesende: Schneider, Ecker

Thema: **Änderungen für das Einlesen der Absolventendaten**

Besprechung mit Herrn Schneider über die nötigen Änderungen in der originalen AbsDB und Terminisierung.

---

Datum: 25.04.2007

Anwesende: Schneider, Ecker, Grossberger

Thema: **Änderungen für das Einlesen der Absolventendaten**

Da Herr Schneider erst mit dem Durchsehen begonnen hatte, besprachen wir die Gesamtanforderungen und besondere Einzelfälle noch einmal durch.

---

Datum: 25.04.2007

Anwesende: Dietrich, Ecker, Grossberger

Thema: **Backend**

Herr Dietrich war erst der Meinung, dass wir anstatt des MS Desktop Engine eine zweite Access Datenbank erstellen, die die Tabellen beinhaltet und auf dem Server liegt, während sich alle Funktionen am Frontend auf dem PC von Herrn Schneider befinden. Nachdem uns Herr Schneider aber zwei Tage zuvor sagte er wolle die Verantwortlichkeit für den Absolventenverein aufgeben und es sei noch kein Nachfolger bekannt, hat die Zweiteilung in ein Front- und ein Backend keinen Sinn. Also wird das Projekt als ganzes auf dem Schulserver unter dem Login des Absolventenvereins zur Verfügung stehen.

---