

Datenbanken Vorlesungsbericht

Elizabeth Fobizi – Axel Schmidt – Andreas Hofmeier

Auftraggeber: Dipl. Ing. Heinrich Pasligh,
Hochschule Bremen
Vermittler: Prof. Dr.-Ing. Uwe Meyer
Abgabe am: 18.06.2004

Version: 0.12
Letzte Änderung: 18.06.2004 – 01:18 durch A. Hofmeier
Status: Ready, Last

Andreas Hofmeier _____

Axel Schmidt _____

Elizabeth Fobizi _____

Zusammenfassung

Dieser Bericht dokumentiert die Erstellung einer Datenbank durch Axel Schmidt, Elizabeth Fobizi und Andreas Hofmeier im Zuge der Datenbankvorlesung von Prof. Dr.-Ing. Meyer.

Es wurde ein Szenario erstellt, in dem das Einsatzziel festgehalten wurde. Anhand des Szenarios konnte ein ER-Diagramm entwickelt werden, welches die beschriebenen Zusammenhänge grafisch aufzeigt. Die Entwicklung des ER-Diagramms erfolgte hierbei in mehreren Schritten.

Mit Hilfe dieses Diagramms wurde ein Übergang in das relationale Datenmodell vorgenommen. Die entwickelten Relationen wurden zunächst tabellarisch aufgestellt und anschließend in SQL-Befehle umgewandelt. Um einen direkten Zugang auf die Datenbank zu vermeiden, wurden Sichten und Prozeduren definiert.

Das Szenario:

Für die Hochschule Bremen soll ein Datenbanksystem zur Verwaltung der Stellenausschreibungen und der daraufhin erfolgenden Bewerbungen erstellt werden.

Inhaltsverzeichnis

1	Das Szenario	3
2	Realisierung	5
2.1	ER-Diagramm	5
2.2	Konventionen	6
2.3	Tabellen/Entitys	6
2.3.1	personen, orte	6
2.3.2	institutionen	7
2.3.3	stellenaus – Stellenausschreibungen, verguetungsklassen .	8
2.3.4	bewerbungen, bmedien	9
2.3.5	qualifikationen, sstellenausqualifikationen, spersonenqualifikationen	10
2.3.6	abschluesse, abschlusstypen, sstellenausabschluesse, spersonenabschluesse	11
2.3.7	veroeffentlichungen, vmedien	12
2.3.8	erfahrungen, laender	13
2.4	Normalisierung	13
3	Funktionen und Views	13
3.1	get_abschluesse	14
3.2	add_abschluesse	15
3.3	rem_abschluesse	15
3.4	get_abschluesse_personen	15
3.5	add_abschluesse_personen	16

3.6	rem_abschluesse_personen	16
3.7	get_abschluesse_stellenaus	17
3.8	add_abschluesse_stellenaus	17
3.9	rem_abschluesse_stellenaus	18
3.10	get_abschluesstypen	18
3.11	add_abschluesstypen	18
3.12	rem_abschluesstypen	19
3.13	get_bewerbungen	19
3.14	add_bewerbungen	20
3.15	rem_bewerbungen	20
3.16	get_bmedien	20
3.17	get_erfahrungen	21
3.18	add_erfahrungen	21
3.19	rem_erfahrungen	22
3.20	get_institutionen	22
3.21	add_institutionen	22
3.22	rem_institutionen	23
3.23	get_laender	23
3.24	get_orte	23
3.25	get_personen	24
3.26	add_personen	24
3.27	rem_personen	25
3.28	get_qualifikationen	25
3.29	add_qualifikationen	25

3.30	rem_qualifikationen	26
3.31	get_qualifikationen_personen	26
3.32	add_qualifikationen_personen	26
3.33	rem_qualifikationen_personen	27
3.34	get_qualifikationen_stellenaus	27
3.35	add_qualifikationen_stellenaus	28
3.36	rem_qualifikationen_stellenaus	28
3.37	get_stellenausschreibungen	28
3.38	add_stellenausschreibungen	29
3.39	rem_stellenausschreibungen	30
3.40	get_verguetungsklassen	30
3.41	get_veroeffentlichungen	30
3.42	add_veroeffentlichungen	31
3.43	rem_veroeffentlichungen	31
3.44	get_vmedien	32
4	Konzepte in der Mehrbenutzerumgebung	32
4.1	Sicherheitskonzept	32
4.1.1	Benutzer	33
4.1.2	Verwalter	34
4.1.3	Administrator	34
4.2	Transaktionen	34
5	Eingabemasken – Zugriff auf die Datenbank	34
5.1	Einstiegsmaske	34

5.2	Stellenausschreibungen auflisten	35
5.3	Stellenausschreibungen erstellen, bearbeiten	35
5.4	Qualifikationen einer Stellenausschreibung hinzufügen	36
5.5	Qualifikationen einer Person hinzufügen	37
5.6	Qualifikationen hinzufügen, bearbeiten	37
5.7	Abschlüsse einer Stellenausschreibung hinzufügen	37
5.8	Abschlüsse einer Person hinzufügen	37
5.9	Abschlüsse hinzufügen, bearbeiten	38
5.10	Institutionen hinzufügen, bearbeiten	38
5.11	Alle Bewerber zu einer Stellenausschreibung auflisten	39
5.12	Übersicht Drucken	39
5.13	Drucken/Verschicken Absage	40
5.14	Drucken/Verschicken Einladung	40
5.15	Bewerbungen erstellen, bearbeiten	40
5.16	Erfahrungen einer Person hinzufügen, bearbeiten	42
5.17	Veröffentlichungen einer Person hinzufügen, bearbeiten	43
6	Beispielimplementierung in PostgreSQL	44

1 Das Szenario

Die Datenbank soll die Verwaltung und Verarbeitung von Stellenausschreibungen und den zugehörigen Bewerbungen an der Hochschule Bremen vereinfachen.

Hierzu müssen die Daten der Ausschreibung, sowie die Daten der Bewerber in die Datenbank eingepflegt werden.

Die Einstellung neuer Mitarbeiter gliedert sich in die folgenden Punkte auf:

1. Stellenausschreibung

Nachdem eine Stelle bewilligt ist, wird diese ausgeschrieben, das heißt zum Beispiel in einer Zeitung veröffentlicht oder dem Arbeitsamt gemeldet. Die Stellenausschreibung wird in der Datenbank erfasst, um eine spätere Zuordnung der Bewerber zu ermöglichen. Die Stellenausschreibung umfasst folgende Informationen:

- Stellenbezeichnung (z. B. Fachbereichssekretärin)
- Gewünschter Einstellungstermin
- Stellenbewertung / Vergütungsgruppe (z. B. Vc/Vb BAT)
- Aufgabenbereich(e)
- Musskriterien – Geforderte Voraussetzungen
- Sollkriterien – Gewünschte Voraussetzungen

2. Bewerbungen entgegennehmen

Nachdem die Stellenausschreibung veröffentlicht wurde, gehen Bewerbungen ein. Diese müssen in die Datenbank eingepflegt werden.

Aus den Bewerbungen werden folgende Daten extrahiert und in die Datenbank übernommen:

- Name
- Vorname
- Wohnort
- Strasse
- Geburtsdatum
- Familienstand
- Schulabschluss (Art, Abschlussdatum)
- Ausbildung (Art, Abschlussdatum)
- Studium (nicht für alle Stellen; u.U. mehrere Abschlüsse)

3. Vorauswahl treffen

Die eingehenden Bewerbungen werden nach folgenden Kriterien vorsortiert:

- Musskriterien, die in der Ausschreibung schon vorgegeben sind.
- Sollkriterien, die z.T. in der Ausschreibung vorgegeben oder nachträglich definiert werden.

Die Bewerber müssen über den Stand ihrer Bewerbung informiert werden.

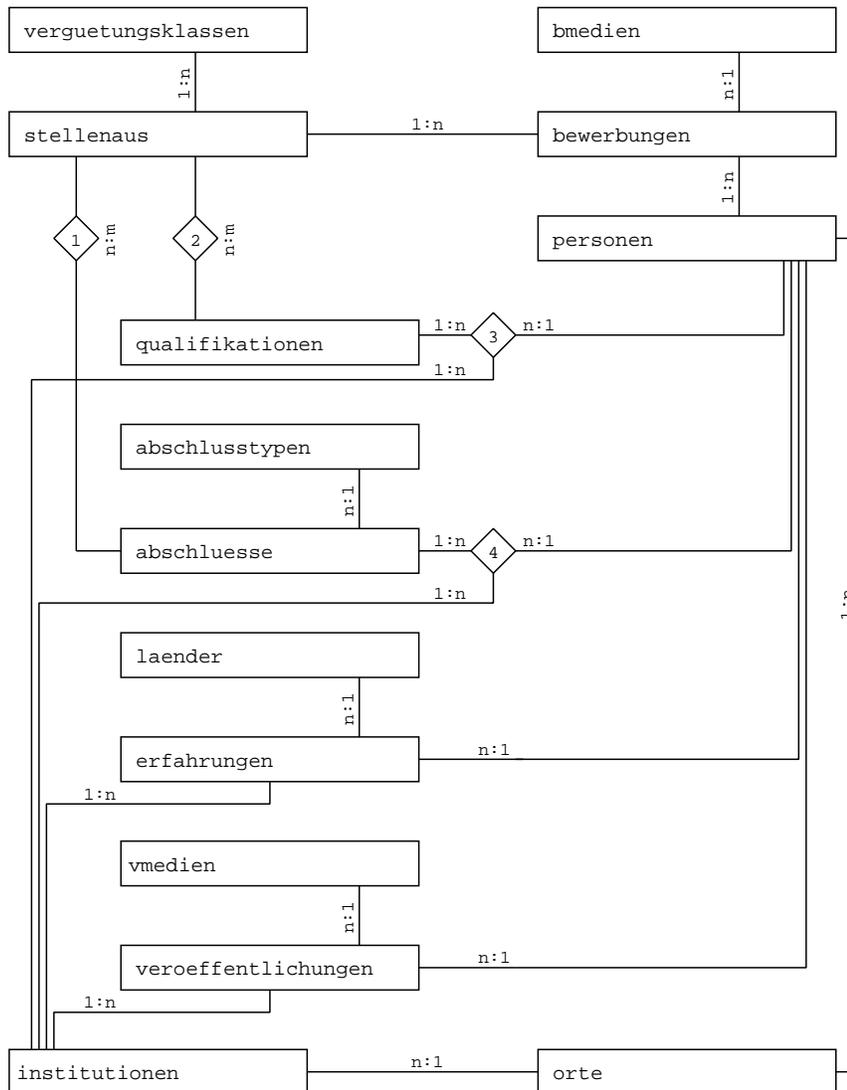
- Bestätigung über den Eingang der Bewerbung
- Absage
- Einladung zum Vorstellungsgespräch
- Zusage (Einstellung)

4. Vorstellungsgespräch

Bei den Vorstellungsgesprächen sollten die bisher erfassten Daten des Bewerbers übersichtlich vorliegen. Diese Daten werden durch zusätzliche Bewertungen (z. B. soziale Kompetenzen) ergänzt, die sich aus dem Vorstellungsgespräch ergeben.

2 Realisierung

2.1 ER-Diagramm



- 1: sstellenausabschluesse
- 2: sstellenausqualifikationen
- 3: spersonenqualifikationen
- 4: spersonenabschluesse

2.2 Konventionen

In den folgenden Tabellen werden folgende Abkürzungen und Zeichen verwendet:

NW	Nullwert
D.	Default
Con.	Constraint
→	Feld verweist auf vorhandenen Eintrag in Tabelle
←	... wird von Tabelle verwiesen
PK	Primary Key
FK	Foreign Key
STRING	Steht für CHARACTER VARYING in der Datenbank
NUM	Steht für NUMERIC in der Datenbank
SERIAL	Automatisches wählen einer neuen Nummer (letzte plus eins) beim Einfügen eines neuen Datensatzes.

2.3 Tabellen/Entitys

2.3.1 personen, orte

In diesen Tabellen werden die Stammdaten über die Bewerber abgelegt.

personen:

← bewerbungen	← veroeffentlichungen
← spersonenqualifikationen	← erfahrungen
← spersonenabschluesse	

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
titel	STRING	Ja	-	-	-
vorname	STRING	Nein	-	-	-
name	STRING	Nein	-	-	-
strasse	STRING	Nein	-	-	mit Hausnummer
plz	INT	Nein	-	-	-
orts_id	INT	Ja	-	FK	→ orte
teleniumpriv	STRING	Ja	-	-	-
teleniumfirm	STRING	Ja	-	-	-
teleniummobile	STRING	Ja	-	-	-
email	STRING	Ja	-	-	-
gebdatum	DATUM	Ja	-	-	-
familienstand	INT	Ja	-	-	0: ledig, 1: verheiratet, 2: geschieden

orte:

← personen

← institutionen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-

2.3.2 institutionen

← spersonenqualifikationen

← veroeffentlichungen

← spersonenabschluesse

← erfahrungen

An dieser Stelle werden Daten von Firmen, Unternehmen, Schulen, Hochschulen, etc, abgelegt.

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-
strasse	STRING	Ja	-	-	mit Hausnummer
plz	INT	Ja	-	-	-
orts_id	INT	Ja	-	FK	→ orte
ansprechpartner	STRING	Ja	-	-	Name des ...
email	STRING	Ja	-	-	-
telenium	STRING	Ja	-	-	-
bemerkung	TEXT	Ja	-	-	-

2.3.3 stellenaus – Stellenausschreibungen, verguetungsklassen

An dieser Stelle werden die Stellenausschreibungen abgelegt. Einer Stellenausschreibung können beliebig viele Bewerber zugeordnet sein.

stellenaus:

← vqualifikationensoll
 ← vqualifikationenmuss

← bewerbungen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
stellenbezeichnung	STRING	Nein	-	-	-
stellenbeschreibung	TEXT	Nein	-	-	-
veroeffentlicht	STRING	Ja	-	-	Kommentar
veroeffentlichtam	DATUM	Ja	-	-	letzte Ver.
aufgabenfeld	TEXT	Nein	-	-	Voraussichtliche Tätigkeiten
geweinstellungsdatum	DATUM	Ja	-	-	Wann soll die Stelle besetzt werden?
bewerbungsschluss	DATUM	Nein	-	-	-
verguetungsklasse	INT	Ja	-	FK	→ verguetungsklassen
fachbereich	STRING	Ja	-	-	-
verantwortlicher	STRING	Nein	-	-	Name
veranttele	STRING	Nein	-	-	Telefonnummer des Verantwortlichen
verantemail	STRING	Nein	-	-	eMail-Adresse des Verantwortlichen
probezeit	INT	Nein	-	-	in Monaten
befristung	INT	Nein	-	-	in Monaten
intern	BOOL	Nein	F	-	interne Ausschreibung

verguetungsklassen:

← stellenaus

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-
beschreibung	TEXT	Ja	-	-	-
einkommen	NUM	Ja	-	-	Euro pro Jahr

2.3.4 bewerbungen, bmedien

In dieser Tabelle werden die eigentlichen Bewerbungen abgelegt. Die Bewerbungen werden mittels des Attributs **bewerber**, welches auf eine Eintrag in der Tabelle **personen** verweist, einer Person zugeordnet. Mittels auf einer Stellenausschreibung. Mit Hilfe der Datums-Attribute kann der aktuelle Stand der Bewerbung ermittelt werden: Sind zum Beispiel die Felder **geladen** und **abgesagt** belegt, so ist daraus zu schließen, dass der Bewerber zwar zum Vorstellungsgespräch eingeladen wurde, schließlich jedoch nicht eingestellt wurde.

bewerbungen:

← personen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
ingang	DATUM	Nein	-	-	-
fruhstereinstellungstermin	DATUM	Ja	-	-	-
auf	INT	Nein	-	FK	→ stellenaus
bewerber	INT	Nein	-	FK	→ personen
abgesagt	DATUM	Ja	-	-	-
geladen	DATUM	Ja	-	-	zum Vorstellungsgespräch
geladen_zeit	STRING	Ja	-	-	Uhrzeit
zugesagt	DATUM	Ja	-	-	-
eingestellt	DATUM	Ja	-	-	-
intern	BOOL	Nein	F	-	interner Bewerber
vorrang	BOOL	Nein	F	-	bei Auswahl
medium	INT	Nein	-	FK	→ bmedien
kommentar	TEXT	Ja	-	-	Erwähnenswertes

bmedien:

In dieser Tabellen sollen die Namen aller Medien abgelegt werden, auf welchen eine Bewerbung eingeht. Zum Beispiel: per Post, eMail, persönlich abgegeben oder intern.

← bewerbungen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-

2.3.5 qualifikationen, sstellenausqualifikationen, spersonenqualifikationen

Hier werden Qualifikationen gespeichert, welche dann den Einzelnen Personen (Bewerbern) über `spersonenqualifikationen` zugeordnet werden. Einer Stellenausschreibung werden Soll- und Mussqualifikationen über `sstellenausqualifikationen` zugeordnet.

In den Zuordnungstabellen wird zusätzlich das Grad der jeweiligen Qualifikation abgelegt: 0 entspricht hier Grundkenntnissen, 4 kennzeichnet werden Experten.

qualifikationen:

← sstellenausqualifikationen

← spersonenqualifikationen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-
beschreibung	TEXT	Ja	-	-	-

Soll- und Mussqualifikationen:

sstellenausqualifikationen: `stellenaus` $\overset{n:m}{-\diamond-}$ `qualifikationen`

Feldname	Datentyp	NW	D.	Con.	Bemerkung
qualifikationen_id	INT	Nein	-	FK	→ qualifikationen
stellenaus_id	INT	Nein	-	FK	→ stellenaus
grad	INT	Nein	-	-	0=Grundlagen, ... 4=Experte
mussoll	INT	Nein	-	-	0=muss, 1=soll

Die beiden FKs und `grad` werden in dieser Tabelle zum PK zusammengefasst.

Qualifikationen einer Person:

spersonenqualifikationen: `personen` $\overset{n:m}{-\diamond-}$ `qualifikationen`

Über das Attribut `institut_id` kann der Name und Anschrift der Institution abgelegt werden, bei welcher die Qualifikation erworben wurde.

Feldname	Datentyp	NW	D.	Con.	Bemerkung
qualifikationen_id	INT	Nein	-	FK	→ qualifikationen
personen_id	INT	Nein	-	FK	→ personen
institut_id	INT	Ja	-	FK	→ institutionen
grad	INT	Nein	-	-	0=Grundlagen, ... 4=Experte

Die beiden obersten FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.6 abschluesse, abschlusstypen, sstellenausabschluesse, spersonenabschluesse

In dieser Tabelle werden Abschlüsse gesammelt und mittels `spersonenabschluesse` einem Bewerber zugewiesen. Ob dieser Abschluss für eine Stelle notwendig oder wünschenswert ist, wird in der Tabelle `sstellenausabschluesse` festgelegt. Der Typ des Abschlusses kommt durch einen Verweis auf einen Eintrag der Tabelle `abschlusstypen` zum Ausdruck. In dieser Tabelle stehen zum Beispiel: Ausbildung, Studium, Schule.

abschluesse:

← sstellenausabschluesse

← spersonenabschluesse

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-
beschreibung	TEXT	Ja	-	-	-
regelzeit	INT	Ja	-	-	in Monaten
typ	INT	Ja	-	FK	→ abschlusstypen

abschlusstypen:

← abschluesse

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-
beschreibung	TEXT	Ja	-	-	-

Soll- und Mussabschlüsse für Stellenausschreibungen:

sstellenausabschluesse: stellenaus ^{n:m} -◇- abschluesse

Feldname	Datentyp	NW	D.	Con.	Bemerkung
abschluesse_id	INT	Nein	-	FK	→ abschluesse
stellenaus_id	INT	Nein	-	FK	→ stellenaus
musssoll	INT	Nein	-	-	0=muss, 1=soll

Die beiden FKs werden in dieser Tabelle zum PK zusammengefasst.

Abschlüsse von Personen:

personenabschluesse: personen $\overset{n:m}{-\diamond-}$ abschluesse

Feldname	Datentyp	NW	D.	Con.	Bemerkung
abschluesse_id	INT	Nein	-	FK	→ abschluesse
personen_id	INT	Nein	-	FK	→ personen
institut_id	INT	Ja	-	FK	→ institutionen
erlangt	DATUM	Ja	-	-	-
durchschnittsnote	NUM	Ja	-	-	-
dauer	INT	Ja	-	-	in Monaten

Die beiden obersten FKs werden in dieser Tabelle zum PK zusammengefasst.

2.3.7 veroeffentlichungen, vmedien

Hier werden mögliche Veröffentlichungen (z.B. Bücher, Artikel) eines Bewerbers abgelegt. Der Bewerber wird mit Hilfe eines FKs verwiesen.

veroeffentlichungen:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
bewerber	INT	Nein	-	FK	→ personen, Autor
titel	STRING	Nein	-	-	-
veroeffentlicht	DATUM	Ja	-	-	... am
beschreibung	TEXT	Ja	-	-	-
medium	INT	Ja	-	FK	→ vmedien
namezeitschrift	STRING	Ja	-	-	-
verweis	STRING	Ja	-	-	ISBN, Internetadr.
auflage	NUM	Ja	-	-	Anzahl Verkauft
verlag	INT	Ja	-	FK	→ institutionen

vmedien:

In dieser Tabellen sollen die Namen aller Medien abgelegt werden, durch welche eine Veröffentlichung stattfinden. Zum Beispiel: Fachzeitschrift, Zeitung, Buch, Internet.

← veroeffentlichungen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-

2.3.8 erfahrungen, laender

In dieser Tabelle werden Erfahrungen abgelegt. Erfahrungen können in diesem Zusammenhang sein: Anstellungen bei Unternehmen – Arbeitserfahrung, Auslandserfahrungen (z.B. Auslandssemester) oder Praktika.

erfahrungen:

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
bewerber	INT	Nein	-	FK	→ personen
name	STRING	Nein	-	-	-
beschreibung	TEXT	Ja	-	-	-
ausland	INT	Ja	-	FK	→ laender
institut_id	INT	Ja	-	FK	→ institutionen
dauer	INT	Ja	-	-	in Monaten
beendetam	DATUM	Ja	-	-	-

laender:

← erfahrungen

Feldname	Datentyp	NW	D.	Con.	Bemerkung
id	INT	Nein	-	PK	SERIAL
name	STRING	Nein	-	-	-

2.4 Normalisierung

Die Normalisierung wurde während der Erstellung des ER-Diagramms bereits intuitiv Berücksichtigt. Jeder Entität wurde ein Schlüsselattribut (Primärschlüssel oder zusammengesetzte Fremdattribute) zugeordnet. Aufzählungen wurden durch Auslagerung der Attribute in eigenständige Entitäten vermieden.

Allerdings ist die Normalisierung nicht in der ganzen Datenbank vollständig durchgeführt worden. So hätten zum Beispiel `namezeitschrift` aus der Tabelle `veroeffentlichungen` oder `straße` aus der Tabelle `personen` ausgelagert werden müssen.

3 Funktionen und Views

Die Schnittstelle zwischen der Benutzeroberfläche und der Datenbank stellen Funktionen und Views da.

Views sind im allgemeinen zum Auslesen von Daten aus der Datenbank gedacht. Sie Joinen verschiedene Tabellen zu einer sinnvollen Ausgabetabelle zusammen. Diese Ausgabetabelle kann bei der Abfrage, von einem Javaprogramm zum Beispiel, mittels where weiter eingeschränkt werden.

Um Daten in die Datenbank einzufügen, zu verändern oder zu löschen sind Funktionen vorgesehen.

Im weiteren wird davon ausgegangen, dass verknüpfte Datensätze mittels “ON DELETE CASCADE” aus der Datenbank entfernt werden können.

Konventionen:

Mit “Haupttabelle(n)” ist die Tabelle gemeint, welche primär ausgelesen oder verändert werden soll. Unter “verknüpfte Tabelle(n)” laufen alle Tabellen, welche von der jeweiligen Funktion mit verwendet werden.

Alle Funktionen beginnen mit “add_” oder mit “rem_”, wobei add_ das Hinzufügen oder Ändern von Datensätzen meint und rem_ sich auf das Entfernen von Datensätzen bezieht.

Views beginnen mit “get_”.

3.1 get_abschluesse

Haupttabelle(n): anschluesse
 verknüpfte Tabelle(n): abschlusstypen

Dieses View liefert eine Liste mit allen in der Datenbank abgelegten Abschlüssen.

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	ID des Abschlusses
name	STRING	Nein	des Abschlusses
beschreibung	TEXT	Ja	des Abschlusses
regelzeit	INT	Ja	in Monaten
typ	INT	Ja	ID des Abschlusstyps
typname	STRING	Ja	Name des Abschlusstyps

3.2 add_abschluesse

Haupttabelle(n): abschluesse
verknüpfte Tabelle(n): -

Die Übergabeparameter entsprechen den Feldern der Rückgabetabelle des Views `get_abschluesse` bis auf das Feld `typename`, welches weggelassen werden kann.

Ist das Feld `id` leer, so wird der Abschluss hinzugefügt. Enthält das Feld die ID eines vorhandenen Datensatzes, so wird dieser überschrieben.

Rückgabetabelle (eine Zeile):

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	des neu hinzugefügten oder geänderten Datensatzes

3.3 rem_abschluesse

Haupttabelle(n): abschluesse
verknüpfte Tabelle(n): -

Diese Funktion löscht den Abschluss mit der ID `id` – dies ist der einzige Übergabeparameter.

3.4 get_abschluesse_personen

Haupttabelle(n): spersonenabschluesse
verknüpfte Tabelle(n): abschluesse,
 personen,
 abschlusstypen,
 institutionen

Das View `get_abschluesse_personen` liefert eine Liste mit allen Abschlüssen und den zugeordneten Personen, bzw deren IDs (`pid`). Diese Tabelle kann mittels `where` von Anwenderprogramm auf eine bestimmte Person eingeschränkt werden.

Rückgabetable (get_abschluesse_personen):

Feldname	Datentyp	NW	Bemerkung
pid	INT	Nein	ID der Person
name	STRING	Nein	des Abschlusses
beschreibung	TEXT	Ja	des Abschlusses
typ	INT	Ja	ID des Abschlusstyps
typname	STRING	Ja	Name des Abschlusstyps
institut_id	INT	Ja	ID des Instituts
institut_name	INT	Ja	Name des Instituts
erlangt	DATUM	Ja	-
durchschnittsnote	NUM	Ja	-
dauer	INT	Ja	in Monaten

3.5 add_abschluesse_personen

Haupttabelle(n): spersonenabschluesse
 verknüpfte Tabelle(n): -

Diese Funktion fügt der Person mit der ID pid einen neuen Abschluss mit der ID aid hinzu. Der Abschluss wurde am Institut, welches unter der ID iid abgelegt ist, erreicht

Parameter:

Parametername	Datentyp	NW	Bemerkung
pid	INT	Nein	ID der Person
aid	INT	Nein	ID des Abschlusses
iid	INT	Ja	ID der Institution
erlangt	DATUM	Ja	-
durchschnittsnote	NUM	Ja	-
dauer	INT	Ja	in Monaten

3.6 rem_abschluesse_personen

Haupttabelle(n): spersonenabschluesse
 verknüpfte Tabelle(n): -

Diese Funktion entfernt den Abschluss mit der ID aid von der Person mit der ID pid.

Parameter (`rem_abschluesse_personen`):

Parametername	Datentyp	NW	Bemerkung
pid	INT	Nein	ID der Person
aid	INT	Nein	ID des Abschlusses

3.7 `get_abschluesse_stellenaus`

Haupttabelle(n): `sstellenausabschluesse`
 verknüpfte Tabelle(n): `abschluesse`,
`stellenaus`,
`abschlusstypen`

Das View `get_abschluesse_stellenaus` liefert eine Liste mit allen Abschluesse und den zugeordneten Stellen bzw. Stellenausschreibungen und deren IDs (`sid`). Diese Tabelle kann mittels `where` von Anwenderprogramm auf eine bestimmte Stellenausschreibungen eingeschränkt werden.

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
sid	INT	Nein	ID der Stellenausschreibung
name	STRING	Nein	des Abschlusses
beschreibung	TEXT	Ja	des Abschlusses
typ	INT	Ja	ID des Abschlusstyps
typename	STRING	Ja	Name des Abschlusstyps
musssoll	INT	Nein	0=muss, 1=soll Kriterium

3.8 `add_abschluesse_stellenaus`

Haupttabelle(n): `sstellenausabschluesse`
 verknüpfte Tabelle(n): -

Diese Funktion fügt der Stellenausschreibung mit der ID `sid` einen neuen Muss- oder Sollabschluss mit der ID `aid` hinzu.

Parameter:

Parametername	Datentyp	NW	Bemerkung
sid	INT	Ja	ID der Stellenausschreibung
aid	INT	Nein	ID des Abschlusses
musssoll	INT	Nein	0=muss, 1=soll Kriterium

3.9 rem_abschluesse_stellenaus

Haupttabelle(n): stellenausabschluesse
verknüpfte Tabelle(n): -

Diese Funktion löscht einen Muss- oder Sollabschluss mit der ID `aid` aus der Stellenausschreibung mit der ID `sid`.

Parameter:

Parametername	Datentyp	NW	Bemerkung
sid	INT	Nein	ID der Stellenausschreibung
aid	INT	Nein	ID des Abschlusses

3.10 get_abschluesstypen

Haupttabelle(n): abschlusstypen
verknüpfte Tabelle(n): -

Dieses View gibt eine Tabelle mit allen Abschlusstypen aus. Diese Tabelle kann mittels where eingeschränkt von Anwenderprogramm werden.

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	ID des Abschlusstyps
name	STRING	Nein	-
beschreibung	TEXT	Ja	-

3.11 add_abschluesstypen

Haupttabelle(n): abschlusstypen
verknüpfte Tabelle(n): -

Diese Funktion fügt einen neuen Abschlusstyp ein, wenn der Parameter `id` leer ist oder überschreibt einen Abschlusstyp, falls eine gültige ID angegeben ist.

Parameter:

Parametername	Datentyp	NW	Bemerkung
id	INT	Ja	ID des Abschlusstyps
name	STRING	Nein	-
beschreibung	TEXT	Ja	-

3.12 rem_abschlusstypen

Haupttabelle(n): abschlusstypen
verknüpfte Tabelle(n): -

Dieses Funktion entfernt den Abschlusstyp mit der ID `id` aus der Tabelle `abschlusstypen`.

Parameter:

Parametername	Datentyp	NW	Bemerkung
<code>id</code>	INT	Nein	ID des Abschlusstyps

3.13 get_bewerbungen

Dieses View liefert eine Liste, welche alle in der Datenbank gespeicherten Bewerbungen und den zugehörigen Personen und Stellenausschreibungen beinhaltet. Diese Liste kann mittels einer `where`-Klausel schon bei der Anfrage an die Datenbank auf eine Person beschränkt werden.

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
<code>id</code>	INT	Nein	der Bewerbung
<code>ingang</code>	DATUM	Nein	-
<code>fruhstereinstellungstermin</code>	DATUM	Ja	-
<code>auf</code>	INT	Nein	ID Stellenausschreibung
<code>bewerber</code>	INT	Nein	ID Person
<code>abgesagt</code>	DATUM	Ja	-
<code>geladen</code>	DATUM	Ja	zum Vorstellungsgespräch
<code>geladen_zeit</code>	STRING	Ja	Uhrzeit
<code>zugesagt</code>	DATUM	Ja	-
<code>eingestellt</code>	DATUM	Ja	-
<code>intern</code>	BOOL	Nein	interner Bewerber
<code>vorrang</code>	BOOL	Nein	bei Auswahl
<code>medium</code>	STRING	Nein	-
<code>kommentar</code>	TEXT	Ja	Erwähnenswertes

3.14 add_bewerbungen

Haupttabelle(n): bewerbungen
verknüpfte Tabelle(n): bmedien

Diese Funktion legt einen neuen Bewerbungen-Rekord in der Datenbank an oder ändert einen bestehenden ab. Die Übergabeparameter entsprechen den Spalten der Rückgabetabelle des Views `get_bewerbungen` mit diesen Unterschieden:

Das ID-Feld muss leergelassen werden, wenn eine neue Bewerbung hinzugefügt werden soll. Ist dieses Feld mit einer Nummer besetzt, so wird der durch diese Nummer gekennzeichnete Datensatz in der Datenbank (in Tabelle `bewerbungen`) mit den übergebenden Daten überschrieben.

Ist das Medium (über das die Bewerbung eingegangen ist) mit dem Namen `bmedien` noch nicht in der Datenbank (in Tabelle `bmedien`) eingepflegt, so wird dies nun getan.

3.15 rem_bewerbungen

Haupttabelle(n): bewerbungen
verknüpfte Tabelle(n): -

Diese Funktion entfernt eine Bewerbung aus der Datenbank (aus der Tabelle `bewerbungen`). Der zu löschende Datensatz wird durch den einzigen Parameter – ID – beschrieben.

3.16 get_bmedien

Haupttabelle(n): bmedien
verknüpfte Tabelle(n): -

Dieses View liefert eine Liste mit allen in der Tabelle `bmedien` verzeichneten Medien über die eine Bewerbung eingehen kann:

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
name	STRING	Nein	-

3.17 get_erfahrungen

Haupttabelle(n): erfahrungen
verknüpfte Tabelle(n): laender

Dieses View liefert eine Liste, welche alle in der Datenbank gespeicherten Erfahrungen und den zugehörigen Personen beinhaltet. Diese Liste kann mittels einer where-Klausel schon bei der Anfrage an die Datenbank auf eine Person beschränkt werden.

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	der Erfahrung
bewerber	INT	Nein	ID der Person
name	STRING	Nein	-
beschreibung	TEXT	Ja	-
ausland	STRING	Ja	Wenn ja, Name.
institut_id	INT	Ja	-
dauer	INT	Ja	in Monaten
beendetam	DATUM	Ja	-

3.18 add_erfahrungen

Haupttabelle(n): erfahrungen
verknüpfte Tabelle(n): laender

Diese Funktion legt einen neuen Erfahrungen-Rekord in der Datenbank an oder ändert einen bestehenden ab. Die Übergabeparameter entsprechen den Spalten der Rückgabetabelle des Views `get_erfahrungen` mit diesen Unterschieden:

Das ID-Feld muss leergelassen werden, wenn eine neue Erfahrung hinzugefügt und einer Person zugewiesen werden soll. Ist dieses Feld mit einer Nummer besetzt, so wird der durch diese Nummer gekennzeichnete Datensatz in der Datenbank (in Tabelle `erfahrungen`) mit den übergebenden Daten überschrieben.

Ist das Land mit dem Namen `ausland` noch nicht in der Datenbank (in Tabelle `laender`) eingepflegt, so wird dies nun getan.

3.19 rem_erfahrungen

Haupttabelle(n): erfahrungen
verknüpfte Tabelle(n): -

Diese Funktion entfernt eine Erfahrung aus der Datenbank (aus der Tabelle **erfahrungen**). Der zu löschende Datensatz wird durch den einzigen Parameter – ID – beschrieben.

3.20 get_institutionen

Haupttabelle(n): institutionen
verknüpfte Tabelle(n): orte

Dieses View liefert eine Liste, welche alle in der Datenbank gespeicherten Institutionen beinhaltet. Diese Liste kann mittels einer where-Klausel schon bei der Anfrage an die Datenbank beschränkt werden. So kann man zum Beispiel eine einzige zu eine ID gehörenden Institution finden.

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	-
name	STRING	Nein	-
strasse	STRING	Ja	mit Hausnummer
plz	INT	Ja	-
ort	STRING	Ja	-
ansprechpartner	STRING	Ja	Name des ...
email	STRING	Ja	-
telenium	STRING	Ja	-
bemerkung	TEXT	Ja	-

3.21 add_institutionen

Haupttabelle(n): institutionen
verknüpfte Tabelle(n): orte

Diese Funktion legt einen neuen Institutionen-Rekord in der Datenbank an oder ändert einen bestehenden. Die Übergabeparameter entsprechen den Spalten der Rückgabetable des Views **get_institutionen** mit folgendem Unterschieden:

Das ID-Feld muss leergelassen werden, wenn eine neue Institution hinzugefügt werden soll. Ist dieses Feld mit einer Nummer besetzt, so wird der durch diese

Nummer gekennzeichnete Datensatz in der Datenbank (in Tabelle `institutionen`) mit den übergebenden Daten überschrieben.

Ist der Ort mit dem Namen `ort` noch nicht in der Datenbank (Tabelle `orte`) eingepflegt, so wird dies nun getan.

3.22 `rem_institutionen`

Haupttabelle(n): `institutionen`
verknüpfte Tabelle(n): -

Diese Funktion entfernt eine Institution aus der Datenbank (aus der Tabelle `institutionen`). Der zu löschende Datensatz wird durch den einzigen Parameter `ID` beschrieben.

3.23 `get_laender`

Haupttabelle(n): `laender`
verknüpfte Tabelle(n): -

Dieses View liefert eine Liste mit allen in der Tabelle `laender` verzeichneten Ländern:

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
<code>name</code>	<code>STRING</code>	Nein	des Landes

3.24 `get_orte`

Haupttabelle(n): `orte`
verknüpfte Tabelle(n): -

Dieses View liefert eine Liste mit allen in der Tabelle `orte` verzeichneten Orten:

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
<code>name</code>	<code>STRING</code>	Nein	des Ortes

3.25 get_personen

Dieses View liefert eine Liste, welche alle Personen enthält. Diese Liste kann mittels einer where-Klausel schon bei der Anfrage an die Datenbank eingeschränkt werden. So ist es zum Beispiel möglich nur eine Person mit einer bestimmten ID pro Abfrage zu erhalten.

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	-
titel	STRING	Ja	-
vorname	STRING	Nein	-
name	STRING	Nein	-
strasse	STRING	Nein	mit Hausnummer
plz	INT	Nein	-
ort	STRING	Ja	-
teleniumpriv	STRING	Ja	-
teleniumfirm	STRING	Ja	-
teleniummobile	STRING	Ja	-
email	STRING	Ja	-
gebdatum	DATUM	Ja	-
familienstand	INT	Ja	0: ledig, 1: verheiratet, 2: geschieden

3.26 add_personen

Haupttabelle(n): personen
verknüpfte Tabelle(n): orte

Diese Funktion legt einen neuen Personen-Rekord in der Datenbank an oder ändert einen bestehenden. Die Übergabeparameter entsprechen den Spalten der Rückgabetabelle des Views `get_personen` mit folgendem Unterschieden:

Das ID-Feld muss leergelassen werden, wenn eine neue Person hinzugefügt werden soll. Ist dieses Feld mit einer Nummer besetzt, so wird der durch diese Nummer gekennzeichnete Datensatz in der Datenbank (in Tabelle `personen`) mit den übergebenden Daten überschrieben.

Ist der Ort mit dem Namen `ort` noch nicht in der Datenbank (Tabelle `orte`) eingepflegt, so wird dies nun getan.

3.27 rem_personen

Haupttabelle(n): personen
verknüpfte Tabelle(n): -

Diese Funktion entfernt eine Person aus der Datenbank (aus der Tabelle **personen**). Der zu löschende Datensatz wird durch den einzigen Parameter **ID** beschrieben.

3.28 get_qualifikationen

Haupttabelle(n): qualifikationen
verknüpfte Tabelle(n): -

Das View **get_qualifikationen** liefert eine Liste mit allen Qualifikationen.

Rückgabetabelle:

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	ID Qualifikation
name	STRING	Nein	der Qualifikation
beschreibung	TEXT	Ja	der Qualifikation

3.29 add_qualifikationen

Haupttabelle(n): qualifikationen
verknüpfte Tabelle(n): -

Die Übergabeparameter entsprechen den Feldern der Rückgabetabelle des Views **get_qualifikationen**.

Ist das Feld **id** leer, so wird die Qualifikation hinzugefügt. Enthält das Feld die ID eines vorhandenen Datensatzes, so wird dieser überschrieben.

Rückgabe: id: ID des neu hinzugefügten oder geänderten Datensatzes.

3.33 rem_qualifikationen_personen

Haupttabelle(n): spersonenqualifikationen
verknüpfte Tabelle(n): -

Diese Funktion entfernt die Qualifikation mit der ID `qid` von der Person mit der ID `pid`.

Parameter:

Parametername	Datentyp	NW	Bemerkung
<code>pid</code>	INT	Nein	ID der Person
<code>qid</code>	INT	Nein	ID der Qualifikation

3.34 get_qualifikationen_stellenaus

Haupttabelle(n): stellenausqualifikationen
verknüpfte Tabelle(n): qualifikationen
stellenaus

Das View `get_qualifikationen_stellenaus` liefert eine Liste mit allen Qualifikationen und den zugeordneten Stellen bzw. Stellenausschreibungen und deren IDs (`sid`). Diese Tabelle kann mittels `where` von Anwenderprogramm auf eine bestimmte Stellenausschreibungen eingeschränkt werden.

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
<code>sid</code>	INT	Nein	ID der Stellenausschreibung
<code>name</code>	STRING	Nein	der Qualifikation
<code>beschreibung</code>	TEXT	Ja	der Qualifikation
<code>musssoll</code>	INT	Nein	0=muss, 1=soll Kriterium
<code>grad</code>	INT	Nein	0=Grundlagen, ... 4=Experte

3.35 add_qualifikationen_stellenaus

Haupttabelle(n): stellenausqualifikationen
verknüpfte Tabelle(n): -

Diese Funktion fügt der Stellenausschreibung mit der ID `sid` eine neue Muss- oder Sollqualifikation mit der ID `qid` hinzu.

Parameter:

Parametername	Datentyp	NW	Bemerkung
sid	INT	Nein	ID der Stellenausschreibung
qid	INT	Nein	ID der Qualifikation
mussoll	INT	Nein	0=muss, 1=soll Kriterium
grad	INT	Nein	0=Grundlagen, ... 4=Experte

3.36 rem_qualifikationen_stellenaus

Haupttabelle(n): stellenausqualifikationen
verknüpfte Tabelle(n): -

Diese Funktion löscht ein Muss- oder Sollqualifikation mit der ID `qid` aus der Stellenausschreibung mit der ID `sid`.

Parameter:

Parametername	Datentyp	NW	Bemerkung
sid	INT	Nein	ID der Stellenausschreibung
qid	INT	Nein	ID der Qualifikation
grad	INT	Nein	0=Grundlagen, ... 4=Experte

3.37 get_stellenausschreibungen

Haupttabelle(n): stellenaus
verknüpfte Tabelle(n): verguetungsklassen

Das View `get_stellenausschreibungen` liefert eine Liste mit allen Stellenausschreibungen. Diese Tabelle kann mittels `where` von Anwenderprogramm noch weiter eingeschränkt werden. Das View vereinigt die Tabellen `stellenaus` und `verguetungsklassen` miteinander.

Rückgabetabelle (`get_stellenausschreibungen`):

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	Identifiziert Datensatz
stellenbezeichnung	STRING	Nein	-
stellenbeschreibung	TEXT	Nein	-
veroeffentlicht	STRING	Ja	Kommentar
veroeffentlichtam	DATUM	Ja	letzte Ver.
aufgabenfeld	TEXT	Nein	Voraussichtliche Tätigkeiten
geweinstellungsdatum	DATUM	Ja	Wann soll die Stelle besetzt werden?
bewerbungsschluss	DATUM	Nein	-
vkname	STRING	Ja	Name der Vergütungsklasse
vkbesch	STRING	Ja	Beschreibung der Vergütungsklasse
vkeink	STRING	Ja	Einkommen der Vergütungsklasse
fachbereich	STRING	Ja	-
verantwortlicher	STRING	Nein	Name
veranttele	STRING	Nein	Telefonnummer des Verantwortlichen
verantemail	STRING	Nein	eMail-Adresse des Verantwortlichen
probezeit	INT	Nein	in Monaten
befristung	INT	Nein	in Monaten
intern	BOOL	Nein	interne Ausschreibung

3.38 add_stellenausschreibungen

Haupttabelle(n): `stellenaus`
 verknüpfte Tabelle(n): `verguetungsklassen`

Die Funktion `add_stellenausschreibungen` legt eine neue Stellenausschreibung in der Datenbank ab oder ändert eine bestehende. Die Übergabeparameter entsprechen den Spalten der Rückgabetabelle des Views `get_stellenausschreibungen` mit folgendem Unterschieden:

Das ID-Feld muss leergelassen werden, wenn eine neue Stellenausschreibung hinzugefügt werden soll. Ist dieses Feld mit einer Nummer besetzt, so wird der durch diese Nummer gekennzeichnete Datensatz in der Datenbank (in Tabelle `stellenaus`) mit den übergebenden Daten überschrieben.

Die Felder `vkbesch` und `vkeink` dürfen freigelassen werden. Ist die Vergütungsklasse mit dem Namen `vkname` noch nicht in der Datenbank eingepflegt, so wird dies nun getan. Sind die Felder `vkbesch` und `vkeink` gegeben wird die

Datenbank (die Tabelle `verguetungsklassen`) mit diesen Werten aktualisiert.

3.39 `rem_stellenausschreibungen`

Haupttabelle(n): `stellenaus`
verknüpfte Tabelle(n): `verguetungsklassen`

Die Funktion `rem_stellenausschreibungen` entfernt eine Stellenausschreibung aus der Datenbank (aus der Tabelle `stellenaus`). Der zu löschende Datensatz wird durch den einzigen Parameter `ID` beschrieben.

3.40 `get_verguetungsklassen`

Haupttabelle(n): `verguetungsklassen`
verknüpfte Tabelle(n): -

Dieses View liefert eine Liste mit allen in der Tabelle `verguetungsklassen` verzeichneten Vergütungsklassen:

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
<code>id</code>	<code>INT</code>	Nein	der Vergütungsklasse
<code>name</code>	<code>STRING</code>	Nein	der Vergütungsklasse
<code>beschreibung</code>	<code>TEXT</code>	Ja	-
<code>einkommen</code>	<code>NUM</code>	Ja	Euro pro Jahr

3.41 `get_veroeffentlichungen`

Haupttabelle(n): `veroeffentlichungen`
verknüpfte Tabelle(n): `vmedien`,
`institutionen`

Dieses View liefert eine Liste, welche alle in der Datenbank gespeicherten Veröffentlichungen und den zugehörigen Personen beinhaltet. Diese Liste kann mittels einer `where`-Klausel schon bei der Anfrage an die Datenbank auf die Veröffentlichungen eine Person beschränkt werden.

Rückgabetable (get_veroeffentlichungen):

Feldname	Datentyp	NW	Bemerkung
id	INT	Nein	der Veröffentlichung
bewerber	INT	Nein	ID des Autors(Person)
titel	STRING	Nein	-
veroeffentlicht	DATUM	Ja	... am
beschreibung	TEXT	Ja	-
medium	STRING	Ja	-
namezeitschrift	STRING	Ja	-
verweis	STRING	Ja	ISBN, Internetadresse
auflage	NUM	Ja	Anzahl Verkauft
verlag	INT	Ja	ID des Verlages(Institution)
verlagname	STRING	Ja	Name des Verlages

3.42 add_veroeffentlichungen

Haupttabelle(n): veroeffentlichungen
 verknüpfte Tabelle(n): vmedien

Diese Funktion legt einen neuen Veroeffentlichungen-Rekord in der Datenbank an oder ändert einen bestehenden ab. Die Übergabeparameter entsprechen den Spalten der Rückgabetable des Views `get_veroeffentlichungen` mit diesen Unterschieden:

Das Feld `verlagname` entfällt.

Das ID-Feld muss leergelassen werden, wenn eine neue Erfahrung hinzugefügt und einer Person zugewiesen werden soll. Ist dieses Feld mit einer Nummer besetzt, so wird der durch diese Nummer gekennzeichnete Datensatz in der Datenbank (in Tabelle `veroeffentlichungen`) mit den übergebenden Daten überschrieben.

Ist das Medium (über das veröffentlicht wurde) mit dem Namen `vmedien` noch nicht in der Datenbank (in Tabelle `vmedien`) eingepflegt, so wird dies nun getan.

3.43 rem_veroeffentlichungen

Haupttabelle(n): veroeffentlichungen
 verknüpfte Tabelle(n): -

Diese Funktion entfernt eine Veröffentlichung aus der Datenbank (aus der Tabelle `veroeffentlichungen`). Der zu löschende Datensatz wird durch den einzigen

Parameter – ID – beschrieben.

3.44 get_vmedien

Haupttabelle(n): vmedien
verknüpfte Tabelle(n): -

Dieses View liefert eine Liste mit allen in der Tabelle vmedien verzeichneten Veröffentlichungsmedien:

Rückgabetable:

Feldname	Datentyp	NW	Bemerkung
name	STRING	Nein	-

4 Konzepte in der Mehrbenutzerumgebung

Das erarbeitete Szenario bezieht sich auf eine Umgebung, in der mehrere Benutzer auf einen Datenbestand zugreifen.

Während “normale” Benutzer (diejenigen, die eine Stelle ausschreiben) nur lesend auf die eingetragenen Informationen zugreifen, kann derjenige, der die Stellenausschreibung und die Daten der Bewerber in die Datenbank einpflegt auch schreibrechte besitzen. Zusätzlich muss der Administrator sämtliche Rechte auf die Datenbank besitzen.

Daher sollten für die unterschiedlichen Benutzer, unterschiedliche Rechte festgelegt werden.

Ferner kann ggf. über den Einsatz von Transaktionen ein Verfälschen der Daten verhindert werden, wenn gleichzeitige Zugriffe auf dieselben Daten gemacht werden.

4.1 Sicherheitskonzept

Es sind drei Benutzergruppen zu identifizieren: “normale” Benutzer, Verwalter und Administratoren. Die Benutzergruppen werden als “Gruppen” auf dem SQL-Server realisiert. Hinterher werden die jeweiligen Rechte auf die Gruppen vergeben. So muss ein entsprechender Benutzer nur in der jeweiligen Gruppe auf der Datenbank eingetragen sein. Die Zuweisung der Rechte direkt auf einem Benutzer wäre zu unflexibel und zu unsicher.

Das folgende Beispiel bezieht sich exemplarisch auf eine PostgreSQL Datenbank.

Mit den folgenden SQL-Statements werden die Gruppen auf der Datenbank angelegt:

```
CREATE GROUP administratoren;  
CREATE GROUP benutzer;  
CREATE GROUP Verwalter;
```

Mit den folgenden SQL-Statements werden die User auf der Datenbank angelegt:

```
CREATE USER dab_admin PASSWORD '****' CREATEUSER CREATEDB IN GROUP  
administratoren;  
CREATE USER dab_benutzer CREATEUSER CREATEDB IN GROUP benutzer;  
CREATE USER dab_verwalter CREATEUSER CREATEDB IN GROUP verwalter;
```

4.1.1 Benutzer

Benutzer haben grundsätzlich nur Leserechte auf alle Sichten. Sie dürfen nicht direkt neue Datensätze in eine Tabelle einfügen.

Beispiel (exemplarisch):

```
GRANT SELECT ON get_personen TO GROUP benutzer;
```

Diese Funktion würde eine Übersicht über bestimmte Personen, die in der Datenbank eingepflegt wurden, anzeigen.

Desweiteren ist darüber nachzudenken, ob alle Benutzer alles lesen können sollen. Ein Vorschlag ist, den Benutzer nur jeweils die Informationen (z.B. Bewerber) lesen zu lassen, die zu einer seiner Ausschreibungen gehören. So kann vermieden werden, dass Fachbereich A von den Einstellungen von Fachbereich B erfährt. Dieses Wissen kann nur missbraucht werden, es gibt keinen Grund diese Information weiterzugeben.

Stellenausschreibungen gehen grundsätzlich über die Verwaltung, daher besteht kein Grund, dass der "normale" Benutzer schreibend auf die Daten zugreift.

4.1.2 Verwalter

Verwalter müssen eingeschränkte Schreibrechte sowie Leserechte besitzen, da sie die Daten (Stellenausschreibung und Bewerber) in die Datenbank einpflegen müssen.

Beispiel (exemplarisch):

```
GRANT SELECT ON get_personen TO GROUP verwalter;  
GRANT EXECUTE ON FUNCTION add_personen (Parameter) TO GROUP  
verwalter;
```

4.1.3 Administrator

Administratoren haben vollen Zugriff auf alle festgelegten Sichten und Funktionen. In Bereichen, in denen Konsistenzprobleme auftreten können ist der Administrator jedoch gezwungen, die dafür vorgesehenen Funktionen zu nutzen. Insofern er Datensätze in eine Tabelle einfügen kann, ohne dass er dafür eine Funktion benutzt, wird davon ausgegangen, dass der Administrator weiß, was er macht.

4.2 Transaktionen

Eine Transaktion nutzt eine Sperre um Änderungen an den Daten durch andere Benutzer zu verhindern, die gerade durch die Transaktion bearbeitet werden. Diese Sperre wird durch das Datenbankmanagementsystem zur Verfügung gestellt.

5 Eingabemasken – Zugriff auf die Datenbank

Der Benutzer arbeitet mit Eingabemasken die, die eben definierten Funktionen aufrufen:

5.1 Einstiegsmaske

Die Einstiegsmaske könnte die Möglichkeit bieten die folgende Operationen auszuführen:

- → Stellenausschreibung erstellen, bearbeiten
- → Stellenausschreibung auflisten

5.2 Stellenausschreibungen auflisten

Eine Tabelle mit allen Stellenausschreibungen erscheint mit der Möglichkeit einzelne Ausschreibung zu löschen (`rem_stellenausschreibungen`) oder die zu bearbeiten → `Stellenausschreibung erstellen, bearbeiten`

Des weiteren können Bewerber zu der Stellenausschreibung hinzugefügt werden (→ `Bewerbungen erstellen, bearbeiten`) oder alle bisherigen Bewerber aufgelistet werden (→ `Alle Bewerber zu einer Stellenausschreibung auflisten`).

Die Stellenausschreibungen könnten nach Datum sortiert werden, damit immer die aktuellsten oben stehen (wichtig, falls Stellenausschreibungen über einen längeren Zeitraum ausbewahrt werden).

5.3 Stellenausschreibungen erstellen, bearbeiten

Diese Maske bietet die Möglichkeit eine Stellenausschreibung einzugeben. Falls von einer vorherigen Maske eine ID übergeben wurde, wird die dazugehörige Stellenausschreibung geladen. Es stehen folgende Eingabefelder zur Verfügung:

`get_stellenausschreibungen, add_stellenausschreibungen:`

- Stellenbezeichnung
- Stellenbeschreibung
- Veröffentlicht im
- Veröffentlicht am
- Aufgabenfeld
- Gewünschtes Einstellungsdatum
- Bewerbungsschluss
- Vergütungsklasse [Name, Beschreibung, Einkommen]: Die in der Datenbank vorhandenen Vergütungsklassen [deren Name] können in einer Pull-Down-Liste ausgewählt werden (`get_verguetungsklassen`). Ein Button bietet das Anlegen (`add_verguetungsklassen`) einer Vergütungsklasse an.

- Fachbereich
- Verantwortlicher
- Telefonnummer des Verantwortlichen
- eMail-Adresse der Verantwortlichen
- Probezeit
- Befristung
- Intern
- Listen mit Soll- und Mussqualifikationen (`get_qualifikationen_stellenaus`) mit Option zum Löschen (`rem_qualifikationen_stellenaus`) und hinzufügen (`→ Qualifikationen einer Stellenausschreibung hinzufügen`).
- Listen mit Soll- und Mussabschlüssen (`get_abschluesse_stellenaus`) mit Option zum Löschen (`rem_abschluesse_stellenaus`) und hinzufügen (`→ Abschluesse einer Stellenausschreibung hinzufügen`).

Aktionen:

- Löschen `→ rem_stellenausschreibungen`
- Stellenausschreibungen auflisten `→ Stellenausschreibung auflisten, löschen`
- Zugehörige Bewerber auflisten `→ Alle Bewerber zu einer Stellenausschreibung auflisten`

5.4 Qualifikationen einer Stellenausschreibung hinzufügen

Bietet die Möglichkeit eine Qualifikationen einer Stellenausschreibung hinzuzufügen, entweder als Muss- oder als Soll-Kriterium.

`add_qualifikationen_stellenaus:`

- Auswahlmöglichkeit der Qualifikationen (`get_qualifikationen`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (`→ Qualifikationen hinzufügen, bearbeiten`)
- Auswahl Muss- oder Sollkriterium
- Grad des Könnens (Grundlagen, ... Experte)

5.5 Qualifikationen einer Person hinzufügen

Bietet die Möglichkeit eine Qualifikationen einer Person hinzufügen.

`add_qualifikationen_personen:`

- Auswahlmöglichkeit der Qualifikationen (`get_qualifikationen`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (\rightarrow `Qualifikationen hinzufügen`, `bearbeiten`)
- Grad des Könnens (Grundlagen, ... Experte)

5.6 Qualifikationen hinzufügen, bearbeiten

(`get_qualifikationen`, `add_qualifikationen`)

- Name der Qualifikation
- Beschreibung der Qualifikation

5.7 Abschlüsse einer Stellenausschreibung hinzufügen

Bietet die Möglichkeit Abschlüsse einer Stellenausschreibung hinzufügen, entweder als Muss- oder als Soll-Kriterium.

`add_abschluesse_stellenaus:`

- Auswahlmöglichkeit der Abschlüsse (`get_abschluesse`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (\rightarrow `Abschlüsse hinzufügen`, `bearbeiten`)
- Auswahl Muss- oder Sollkriterium

5.8 Abschlüsse einer Person hinzufügen

Bietet die Möglichkeit einen Abschluss einer Person hinzufügen.

`add_abschluesse_personen:`

- Auswahlmöglichkeit der Abschlüsse (`get_abschluesse`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (\rightarrow Abschlüsse hinzufügen, bearbeiten)
- Auswahlmöglichkeit der Institute (`get_institutionen`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (\rightarrow Institutionen hinzufügen, bearbeiten)
- Erlangt am
- Durchschnittsnote
- Dauer in Monaten (Das dem Abschluss vorangehende Studium)

5.9 Abschlüsse hinzufügen, bearbeiten

(`get_abschluesse`, `add_abschluesse`)

- Name der Qualifikation
- Beschreibung der Qualifikation
- Regelzeit in Monaten
- Abschlusstyp – Die In der Datenbank vorhandenen Abschlusstypen [deren Name] können in einer Pull-Down-Liste ausgewählt werden (`get_abschlusstypen`). Ein Button bietet das Anlegen (`add_abschlusstypen`) eines Abschlusstypen an.

5.10 Institutionen hinzufügen, bearbeiten

(`get_institutionen`, `add_institutionen`)

- Name des Instituts
- Strasse mit Hausnummer
- PLZ
- Ort – hier wird eine Liste angeboten `get_orte`, es können auch neue Werte eingegeben werden.
- Ansprechpartner
- eMail
- Telefonnummer
- Bemerkungen

5.11 Alle Bewerber zu einer Stellenausschreibung auflisten

Liest die Liste mit allen Bewerbern aus der Datenbank auslesen (`get_bewerbungen`). Diese wird mittels einer where-Klausel auf eine Stellenausschreibung beschränkt. Die Namen und Adressen der Bewerber werden mittels `get_personen` geladen. Diese Tabelle wird nun angezeigt, dazu werden die Operationen Löschen (`rem_bewerbungen`), Bearbeiten und hinzufügen (`→ Bewerbungen erstellen, bearbeiten`) bereitgestellt.

Die erste Spalte der Tabelle könnte den Status der Bewerbung zeigen. Zum Beispiel:

- rote Box: Noch keine Benachrichtigung versandt. (Absagedatum, Einladungsdatum und Einstellungsdatum sind leer)
- orangene Box: Wurde zum Einstellungsgespräch geladen, aber noch nicht abgesagt oder zugesagt. (Einladungsdatum enthält Datum; Absagedatum und Einstellungsdatum sind leer)
- grüne Box: Wurde Eingestellt. (Einladungsdatum enthält Datum)
- blaue Box: Wurde Abgesagt. (Absagedatum enthält Datum)

Auf Grund dieser Staaten können die jeweils sinnvollen Funktionen für jede Zeile angeboten werden:

- Status rot: Drucken/Verschicken Absage, Drucken/Verschicken Einladung
- Status orange: Drucken/Verschicken Absage, Drucken/Verschicken Zusage (Einstellung)
- Status grün: keine
- Status blau: keine

5.12 Übersicht Drucken

Damit man beim Vorstellungsgespräch weiß, mit wem man es zu tun hat, wird hier eine Übersicht über den Bewerber erstellt und ausgedruckt. Dieser Ausdruck kann auch etwas Platz für Notizen enthalten, welche dann nach dem Vorstellungsgespräch in die Datenbank (Tabelle: `bewerbungen`; Feld: `kommentar`) eingepflegt werden können.

5.13 Drucken/Verschicken Absage

Ein Standard-Absage-Schreiben wird vorgeschlagen und kann gedruckt werden. Nach Druck wird das Attribut `abgesagt` auf das aktuelle Datum gesetzt.

5.14 Drucken/Verschicken Einladung

Nach Eingabe eines Termins für das Vorstellungsgespräch wird ein Standard-Einladungs-Schreiben vorgeschlagen und kann gedruckt werden. Nach Druck wird das Attribut `geladen` und `geladenum` auf das eingegebene Datum bzw. Zeit gesetzt.

5.15 Bewerbungen erstellen, bearbeiten

Diese Maske bietet die Möglichkeit eine Bewerbungen einzugeben. Falls von einer vorherigen Maske eine ID übergeben wurde, die die dazugehörige Bewerbungen in die Maske geladen und kann bearbeitet werden. Dazu sind folgende Eingabefelder vorhanden:

Um die Übersicht zu wahren und der Bewerber zuordnen zu können werden folgende Werte in der Maske angezeigt, können aber nicht verändert werden `get_stellenausschreibungen`:

- Stellenbezeichnung
- Stellenbeschreibung
- Aufgabenfeld
- Gewünschtes Einstellungsdatum
- Bewerbungsschluss
- Vergütungsklasse [Name]
- Fachbereich
- Verantwortlicher
- Telefonnummer des Verantwortlichen
- eMail-Adresse der Verantwortlichen
- Intern

(get_personen, add_personen)

- Titel
- Vorname
- Name
- Strasse
- PLZ
- Ort – hier wird eine Liste angeboten `get_orte`, es können auch neue Werte eingegeben werden.
- Telefonnummer Privat
- Telefonnummer Firma
- Telefonnummer Mobile
- eMail
- Gebdatum
- Familienstand

(get_bewerbungen, add_bewerbungen)

- Eingang
- Fruhstereinstellungstermin
- Abgesagt
- Geladen
- Geladen_am
- Zugesagt
- Eingestellt
- Intern
- Vorrang
- Medium – hier wird eine Liste angeboten `get_bmedien`, es können auch neue Werte eingegeben werden.
- Kommentar

- Listen mit Qualifikationen (`get_qualifikationen_personen`) mit Option zum Löschen (`rem_qualifikationen_personen`) und hinzufügen (`→ Qualifikationen einer Person hinzufügen`).
- Listen mit Abschlüssen (`get_abschluesse_personen`) mit Option zum Löschen (`rem_abschluesse_personen`) und hinzufügen (`→ Abschluesse einer Person hinzufügen`).
- Listen mit Erfahrungen (`get_erfahrungen`) mit Option zum Löschen (`rem_erfahrungen`), hinzufügen und bearbeiten (`→ Erfahrungen einer Person hinzufügen, bearbeiten`).
- Listen mit Veröffentlichungen (`get_veroeffentlichungen`) mit Option zum Löschen (`rem_veroeffentlichungen`), hinzufügen und bearbeiten (`→ Veröffentlichungen einer Person hinzufügen, bearbeiten`).

Nachdem die Bewerbung in die Datenbank übernommen wurde schlägt das Programm einen Standart-Brief vor, welcher den Sinn hat dem Bewerber mitzuteilen, dass seine Bewerbung angekommen ist und bearbeitet wird.

5.16 Erfahrungen einer Person hinzufügen, bearbeiten

Bietet die Möglichkeit Erfahrungen einer Person hinzufügen. Falls eine ID (von der aufrufenden Maske) einer Erfahrung übergeben wird, so wird diese in die Maske eingeladen und kann bearbeitet werden.

`get_erfahrungen, add_erfahrungen:`

- Name der Erfahrung
- Beschreibung
- Ausland – hier wird eine Liste angeboten `get_laender`, es können auch neue Werte eingegeben werden.
- Dauer
- Beendetam
- Auswahlmöglichkeit der Institute (`get_institutionen`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (`→ Institutionen hinzufügen, bearbeiten`)

5.17 Veröffentlichungen einer Person hinzufügen, bearbeiten

Bietet die Möglichkeit Veröffentlichungen einer Person hinzufügen. Falls eine ID (von der aufrufenden Maske) einer Veröffentlichung übergeben wird, so wird diese in die Maske eingeladen und kann bearbeitet werden.

`get_veroeffentlichungen`, `add_veroeffentlichungen`:

- Titel der Veröffentlichung
- Veröffentlicht am
- Beschreibung
- Medium – hier wird eine Liste angeboten `get_vmedien`, es können auch neue Werte eingegeben werden.
- Namezeitschrift
- Verweis z.B. ISBN, Internetadresse
- Auflage – Anzahl der Verkauften Exemplare
- Verlag (wird intern als Institutionen abgelegt) Auswahlmöglichkeit der Institute (`get_institutionen`) mit Möglichkeit eine Hinzuzufügen oder zu bearbeiten (→ `Institutionen hinzufügen`, `bearbeiten`)

6 Beispielimplementierung in PostgreSQL

```
--dieser View zeigt alle Bewerber die in Bremen wohnen
create view hbwohn as
  select
    person.name
    person.ort
  from person
  where ort = Bremen

--dieser View zeigt Ansprechpartner und Institute von alle Bewerber
create view institutansprech as
  select
    nr, name, ansprechpartner
  from institute
  group by nr

--dieser View zeigt alle Bewerber dessen Bewerbungen abgesagt wurde
create view bewerbungabsag as
  select
    nr , geladen
  from bewerbungen
  where geladen = ja

--dieser View zeigt Bewerber die ihre Ausbildung in der Regelzeit
--geschafft haben und die durchschnittsnote
create view regelzeit as
  select vschulabschlussepersonen.durchschnittsnote ,
    vausbildungpersonen.nr, vausbildungpersonen.name,
    vausbildungpersonen.regelausbildungszeitzeit
  from vschulabschlussepersonen , vausbildungpersonen
  where vschulabschlussepersonen.nr = vausbildungpersonen.nr
    and vschulabschlussepersonen.durchschnittsnote >.....
    and vausbildungpersonen.Regelausbildungszeitzeit = ja
  group by vausbildungpersonen.nr

--dieser View zeigt alle Bewerber die schonmal im Ausland fuer
--mindestens ueber 6 Monate gearbeitet haben
create view bewerberausland as
  select
    bewerber , beschreibung ,ausland , dauer
  from erfahrungen
  where ausland > 1
    and dauer > 6

--dieser View zeigt alle Stelle die nur im internet överffentlicht wurde
create view internetveroeff as
  select
    nr, title , veroeffentlicht , medium
  from vveroeffentlichungenpersonen
  where medium = internet

insert into orte (name) values ('Bremen');
insert into orte (name) values ('Delmenhorst');
insert into orte (name) values ('Oldenburg');
insert into orte (name) values ('Osnabrueck');

insert into personen (familienstand) values ('ledig');
insert into personen (familienstand) values ('verheiratet');
insert into personen (familienstand) values ('geschieden');

insert into stellenaus (fachbereich, verantwortlicher, probezeit)
  values ('Allgemeinwissenschaftliche äGrundlagenfcher', 'Prof. ABC', '6');
insert into stellenaus (fachbereich, verantwortlicher, probezeit)
  values ('Architektur', 'Prof. DEF', '6');
insert into stellenaus (fachbereich, verantwortlicher, probezeit)
  values ('Bauingenieurwesen', 'GHI', '6');
insert into stellenaus (fachbereich, verantwortlicher, probezeit)
  values ('Elektrotech&Informatik', 'JKL', '6');
```

```

insert into stellenaus (fachbereich, verantwortlicher, probezeit)
values ('Maschinenbau', 'Prof. MNO', '6');

insert into stellenaus (fachbereich, verantwortlicher, probezeit)
values ('Nautik und Internationale Wirtschaft', 'Prof. PQR', '6');

insert into stellenaus (fachbereich, verantwortlicher, probezeit)
values ('Schiffbau, Meerestechnik und Angewandte Naturwissenschaften', 'Prof. STU', '6');

insert into stellenaus (fachbereich, verantwortlicher, probezeit)
values ('Sozialwesen', 'Prof. VWX', '6');

insert into stellenaus (fachbereich, verantwortlicher, probezeit)
values ('Wirtschaft', 'Prof. XYZ', '6');

insert into bewerbungen (name) values ('post');
insert into bewerbungen (name) values ('e-mail');
insert into bewerbungen (name) values ('personliche Abgabe');

insert into abschluesse (name) values ('Diplom');
insert into abschluesse (name) values ('Bachelor');
insert into abschluesse (name) values ('Master');

insert into qualifikationen (name) values ('PhD');

insert into veroeffentlichungen (namezeitschrift) values ('Weser Kurrier');
insert into veroeffentlichungen (namezeitschrift) values ('Frankfurter Times');

insert into vmedien (name) values ('Fachzeitschrift');
insert into vmedien (name) values ('Internet');
insert into vmedien (name) values ('Zeitung');
insert into vmedien (name) values ('Buch');

insert into laender (name) values ('England');
insert into laender (name) values ('Frankreich');
insert into laender (name) values ('Kanada');
insert into laender (name) values ('Schweiz');

```