

OZB Meeting 25.07.2013

- Jede Person (TBL:Person) soll Zugriff auf das OZB-System Pers. erhalten
=> Separate Tabelle entfernen, um fachliche & technische Attribute ausgetrennt zu halten

o - Tabelle Wirkung einführen



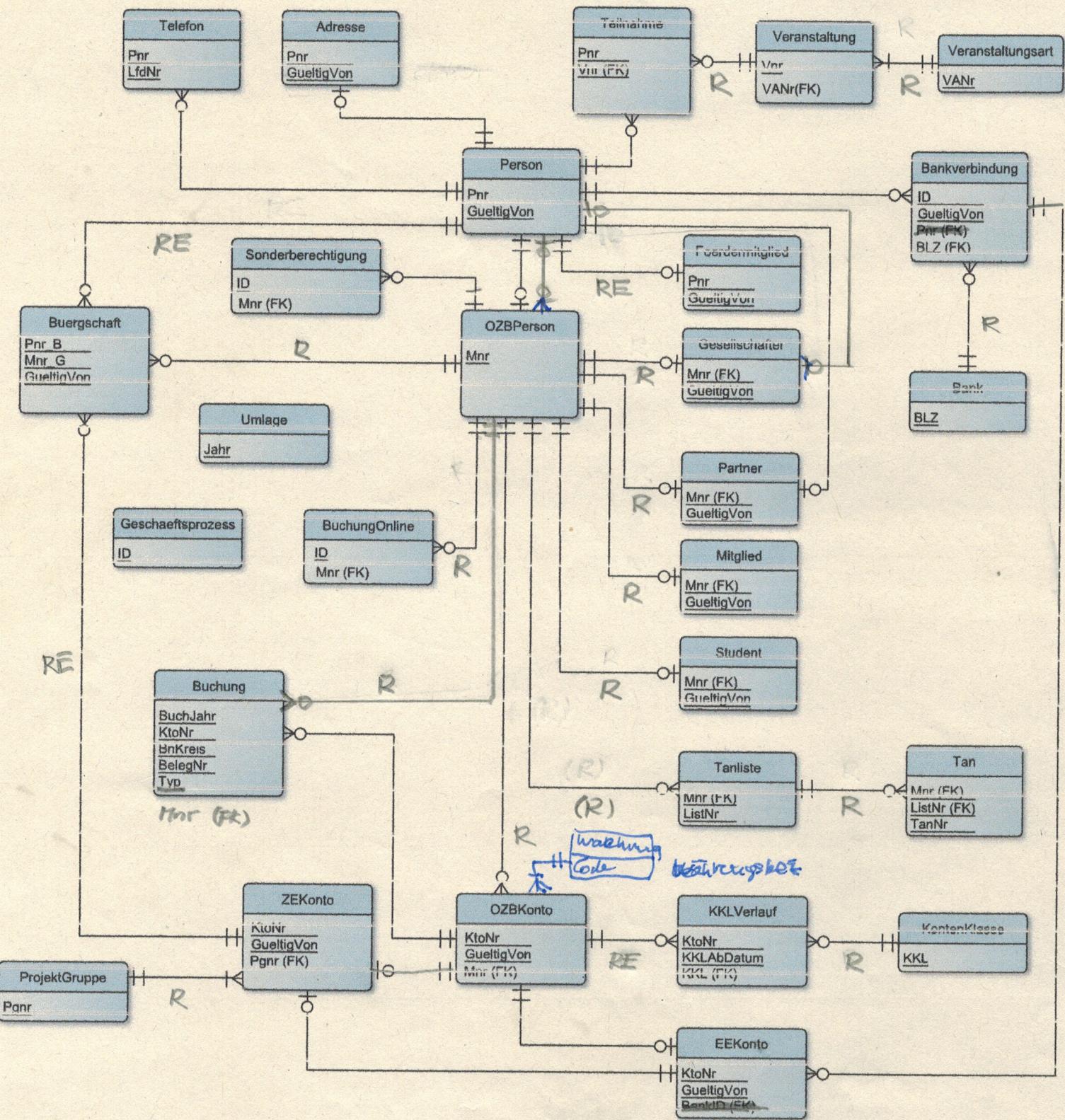
- Person lösen, wenn keine Bedingungen existieren, sonst nicht!

- Datenmodelle:

Noten PNR + Gesellschaft
Person Gesellschaft
1:n

Person OzbPerson über PNR

1 : n



1. Liste der Attribute und deren Beschreibung/Definition

Person

Create Table Anweisung		Keys	Fachliche Beschreibung	Wertevorrat	Beispiel
'Pnr` int(10) unsigned NOT NULL,		PK1	Nummer der Person	1 bis 9999	13
'GuelrigVon` datetime NOT NULL,		PK2	Fachliche Gültigkeit von		2013-01-01 :...
'GuelrigBis` datetime NOT NULL,			Fachliche Gültigkeit bis		9999-12-31 :...
'Rolle` enum('G', 'M', 'P', 'S', 'F') DEFAULT NULL,			Rolle der Person	G=Gesellschafter M=Mitglied P=Partner S=Student F=Fördermitglied	
'Name` varchar(20) NOT NULL,			Name der Person	Länge 30	Dresdner Bank
'Vorname` varchar(15) NOT NULL DEFAULT '' ,			Vorname der Person	Länge 20	Maria-Therese
'Geburtsdatum` date DEFAULT NULL,			Geburtsdatum	Gültiges Kalenderdatum	13.06.1983
'EMail` varchar(255) DEFAULT NULL,			Email-Adresse	50 Zeichen	IreneVoss@web.de
'SperrKZ` tinyint(2) unsigned NOT NULL DEFAULT '0' ,			0 = Logon freigegeben 1 = Logon gesperrt (nach 3 Tan-Falscheingaben)	0 oder 1	0
'SachPnr` int(10) unsigned DEFAULT NULL,			Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 9999	13

Adresse

Create Table Anweisung		Keys	Fachliche Beschreibung	Wertevorrat	Beispiel
'Pnr` int(10) unsigned NOT NULL,		PK1	Nummer der Person	1 bis 9999	13
'GuelrigVon` datetime NOT NULL,		PK2	Fachliche Gültigkeit von		2013-01-01 :...
'GuelrigBis` datetime NOT NULL,			Fachliche Gültigkeit bis		9999-12-31 :...
'Strasse` varchar(50) NOT NULL,			Name der Strasse		Rosenstrasse
'Nr` varchar(10) DEFAULT NULL,			Hausnummer		134A
'PLZ` int(5) unsigned DEFAULT NULL,			Postleitzahl	→ ändern char(10)	6M4X3A
'Ort` varchar(50) NOT NULL,			Name des Postortes	char(20) 20 genügt	Etobicoke
'Vermerk` varchar(100) DEFAULT NULL,			Zustellvermerk	char(30) 30 genügt	bei Müller
'SachPnr` int(10) unsigned DEFAULT NULL,			Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 9999	13

Gesellschafter

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Mnr` int(10) unsigned NOT NULL,	PK1, FK*)	Mitgliedsnummer (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01 : ...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31 : ...
`FAlfdNr` char(20),		Laufende Nr. unter dem der Gesellschafter beim FA geführt wird	1 bis 100	5
`FASteuerNr` char(15),		Steuernummer des Gesellschafters		0041/1829/00433
`FAIdNr` char(15),		Finanzamt-Identnummer	15-stellig	59 087 635 428
`Wohnsitzfinanzamt` varchar(50),		zuständiges Finanzamt		Schwäbisch-Gmünd
`NotarPnr` int(10) unsigned, DEFAULT NULL,		Pnr des Notariats, bei dem der Gesellschafter beurkundet wurde	1 bis 99999	888
`BeurkDatum` date, DEFAULT NULL,		Datum der Beurkundung	Kalenderdatum > Aufnahmedatum bzw. Wechsel von M → G	
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

*) FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr)

Student

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Mnr` int(10) unsigned NOT NULL,	PK1, FK*)	Mitgliedsnummer (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01 : ...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31 : ...
`AusbildBez` varchar(30),		Bezeichnung der Ausbildung/Studiengang		Erzieherin
`InstitutName` varchar(30),		Name des Ausbildungsinstituts		Seminar für Waldorfpädagogik
`Studienort` varchar(30),		Studienort		Stuttgart
`Studienbeginn` date,		Beginn des Studiums / Ausbildung	Kalenderdatum > Aufnahmedatum bzw.	
`Studienende` date,		voraussichtliches Ende des Studiums/Ausbildung	Kalenderdatum > Studienbeginn	

NULL,	`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13
-------	--	--	--	-------------	----

*	FOREIGN KEY (VANR)	REFERENCES Veranstaltungsort (VANR)
---	--------------------	-------------------------------------

Teilnahme

Create Table Anweisung

Datenbestand		Wertebereich/Vorrat	Bespiel
Name	Typ		
'Pnr' int(10) unsigned NOT NULL ,	PK1	Personennummer 1 bis 9999	13
'Vnr' int(11) unsigned NOT NULL ,	PK2 FK*)	Nummer der Veranstaltung 1 bis 999	50
'TeilnArt' enum ('a','e','u','l','m'),	Art/Status der Teilnahme	1 = eingeladen e = entschuldigt m = ??? a = anwesend u = unentschuldigt	
'SachPnr' int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999
			13

*) FOREIGN KEY (Vnr) REFERENCES Veranstaltung(Vnr)

Partner

Create Table Anwendung

Beispiel		Wertebereich/Vorrat	
Rechtschreibung	Beschreibung	Wertebereich	Vorrat
`Mnr` int(10) unsigned NOT NULL ,	PK1 FK*) Mitgliedsnummer der OZB-Person mit Vollmitgliedschaft (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von	2013-01-01:...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis	9999-12-31:...
`Pnr_P` int(10) unsigned NOT NULL ,		Personennummer des Partners ohne eigenes Konto Anm: Partner muß nicht o/ZB-Mitglied sein	1 bis 99999 13
`Berechtigung` char(1) NOT NULL DEFAULT '1' ,		Berechtigung des Partners	1 = leseberechtigt v = vol...]

<code>'VerwendetAm'</code> date DEFAULT NULL ,		Datum, an dem die Tan verwendet wurde	Kalenderdatum > TanListDatum
<code>'Status'</code> enum ('o', 'x') ,		Verwendungsstatus einer Tan	o = noch verfügbar x = bereits verwendet

*) FOREIGN KEY (Mnr, ListNr) REFERENCES Tanliste (Mnr, ListNr)

Bank

Create Table Anweisung

Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
PK	Bankleitzahl	8-stellig	12030000
	Bank International Code	11-stellig	GENODES1RMA
	Name des Kreditinstituts	50-stellig	DKB Bank

Bankverbindung

Create Table Anweisung

Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
PK1	Identifikationsnummer	Wird generiert	20
PK2	Fachliche Gültigkeit von Fachliche Gültigkeit bis	2013-01-01 : ... 9999-12-31 : ...	
	Personennummer	1 bis 9999	13
	Kontonummer des o/ZB-Mitglieds bei der Bank	1011507447	
	International Bank Account Number	DE83 6006 0000 5610 02	
FK*)	Bankleitzahl	8-stellig	12030000
	Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999 13	

*) FOREIGN KEY (BLZ) REFERENCES Bank (BLZ)

OZBKonto

Create Table Anweisung

Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
PK1	o/ZB-Kontonummer des Mitglieds	10001 bis 79999 und 90001 bis 99999	90038
PK2	Fachliche Gültigkeit von	2013-01-01 : ... 9999-12-31 : ...	
FK1 *	Fachliche Gültigkeit bis Mitgliedsnummer (gleicher	1 bis 9999 13	

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Panr` tinyint(2) unsigned,	PK	Nummer der Projektgruppe/-kategorie	1 bis 20	2
`ProjGruppenBez` varchar(50),		Bezeichnung der Projektgruppe	char (50) wäre ausreichend	Umschuldung

ZEKonto

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`KtoNr` int (5) unsigned NOT NULL ,	PK1	o/ZB-Kontonummer des Mitglieds für Zusatzentnahmen	10001 bis 79999 und 90001 bis 99999	20038
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von	2013-01-01 : ...	
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis	9999-12-31 : ...	
`Pgnr` tinyint (2) unsigned NOT NULL,	FK	Nummer der Projektgruppe/-kategorie	1 bis 20	2
`EEKtoNr` int (5) unsigned NOT NULL,		Der ZE zugeordnetes EEKonto für die Abrechnung der ZE (Kredit)	10001 bis 79999 und 90001 bis 99999	90038
`ZENR` char (10) ,		Eindeutige Nummer des ZE-Vertrags		D130115, bzw. ZE130115 in Stgt, bzw. Bonn
`ZEAbDatum` date DEFAULT NULL,		Ab-Datum der ZE gem. ZE-Vertrag	Kalenderdatum >= KtoEinrDatum (i.d.R. Datum der ZENR)	15.01.2013
`ZEEndDatum` date DEFAULT NULL,		Endedatum der ZE gem. ZE-Vertrag	Kalenderdatum > ZEAbDatum	01.02.2018
`ZEBetrag` decimal (10,2) DEFAULT NULL,		Betrag der ZE		12000,00
`Laufzeit` tinyint (4) unsigned NOT NULL,		Laufzeit in Jahren	2 bis 20	5
`ZahlModus` char (1) DEFAULT 'M' ,		Zahlungsmodus/frequenz	m = monatlich q = quartalsweise j = jährlich	m
`TilgRate` decimal (10,2) NOT NULL DEFAULT '0.00' ,		vertraglich vereinbarte Tilgungsrate		55,50
`NachsparRate` decimal (10,2) NOT NULL DEFAULT '0.00' ,		vertraglich vereinbarte Nachsparrate		60,00
`KDURate` decimal (10,2) NOT NULL DEFAULT '0.00' ,		vertraglich vereinbarte Kostendeckungsumlage		0,00
`RDURate` decimal (10,2) NOT NULL DEFAULT '0.00' ,		vertraglich vereinbarte Risikodeckungsumlage		23,78
`ZESTatus` char (1) NOT NULL DEFAULT 'A' ,		Status der ZE	a = aktiv e = beendet u = unterbrochen	a
`Kalk Leihpunkte` int(11) DEFAULT NULL,		voraussichtlich für die ZE	ganzahlig	210423

Create Table Anweisung		Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`KKL`	char (1) NOT NULL,	PK	Kontenklasse	A bis Z	A
`KKEinrDatum`	date NOT NULL,		Datum an dem die Kontenklasse eingerichtet wurde		
`Prozent`	decimal (5, 2) unsigned NOT NULL,		Kontenklassen-Prozentsatz		100,00

KKLVerlauf

Create Table Anweisung		Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`KtoNr`	int (5) unsigned NOT NULL ,	PK1	O/ZB-Kontonummer, die einer Kontenklasse zugeordnet wird	10001 bis 79999 und 90001 bis 99999	20038
`KKLAbDatum`	date NOT NULL,	PK2	Beginn der Zuordnung	Kalenderdatum > Datum der letzten Zuordnung	
`KKL`	char (1) NOT NULL,	FK*)	Kontenklasse	A bis Z	A

*) FOREIGN KEY (KKL) REFERENCES kontenklasse (KKL)

Buchung

Create Table Anweisung		Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`BuchJahr`	int (4) unsigned NOT NULL,	PK1	Jahr der Buchung		2013
`KtoNr`	int (5) unsigned NOT NULL,	PK2	O/ZB-Konto auf dem die Buchung gelistet werden soll	10001 bis 79999 und 90001 bis 99999	20038
`BnKreis`	char (2) NOT NULL,	PK3	Belegnummernkreis (keine exclusive Aufzählung, kann in der FiBu jederzeit erweitert werden)	'B-' = Bank 'U-' = Umbuchung 'K-' = Kasse 'P-' = Punkte 'V-' = Saldovortrag	U-
`BelegNr`	int (10) unsigned NOT NULL,	PK4	Belegnummer aus der Finanzbuchhaltung	1 bis 100000	334
`Typ`	char (1) NOT NULL,	PK5	Typ der Buchung	w = Währungsbuchung p = Punktebuchung	W
`Mnr`	int (10) unsigned NOT NULL,	FK*)	Mitgliedsnummer (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`Belegdatum`	date NOT NULL,		Datum des Buchungsbelegs	Kalenderdatum	
`BuchDatum`	date NOT NULL,		Datum des Buchungslaufs	Kalenderdatum	

Sonderberechtigung

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
` <u>ID</u> ` int(11) unsigned NOT NULL AUTO_INCREMENT,	PK FK*	Identnummer Nummer des Mitglieds, das die Onlinebuchung ausgeführt hat	1 bis 1000 1 bis 9999	5 13
`Email` varchar(40) NOT NULL,		Emailadresse der/s Berechtigten		EvaMueller@web.de
`Berechtigung` enum('IT','MV','RW','ZE','OeA') NOT NULL,		Art der Berechtigung		IT = Informatiker MV = Mitgliederverw. RW = Rechnungswesen ZE = Zusatznahmen OeA= Öffentlichkeit
*				

*) FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr)

Geschäftsprozess

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
` <u>ID</u> ` int(11) unsigned NOT NULL AUTO_INCREMENT, `Beschreibung` varchar(200) NOT NULL,	PK	Identnummer Kurzbeschreibung des Geschäftsprozesses	1-1000	14
`IT` tinyint(1) unsigned NOT NULL,		GP für Informatiker	0 oder 1	
`MV` tinyint(1) unsigned NOT NULL,		GP für Mitgliederverw.	0 oder 1	
`RW` tinyint(1) unsigned NOT NULL,		GP für Rechnungswesen	0 oder 1	
`ZE` tinyint(1) unsigned NOT NULL,		GP für Zusatznahmen	0 oder 1	
`OeA` tinyint(1) unsigned NOT NULL,		GP für Öffentlicharbeit	0 oder 1	

Umlage

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Jahr` tinyint(4) unsigned NOT NULL,	PK	Laufzeit der ZE in Jahren	2 bis 30	3
`RDU` decimal(5,2) NOT NULL,		Risikodeckungsumlage Prozentsatz der Rest-ZE zum Jahresbeginn	0,00 bis 5,00	0,50
`KDU` decimal(5,2) NOT NULL DEFAULT '0.00',		Kostendeckungsumlage Prozentsatz der Rest-ZE zum Jahresbeginn	0,00 bis 5,00	0,00

2. Create Table Anweisungen

<pre>'Aufnahmedatum` date DEFAULT NULL, `Austrittsdatum` date DEFAULT NULL, `Schulungsdatum` date DEFAULT NULL, `Gesperrt` tinyint(2) unsigned NOT NULL DEFAULT 0, `SachPnr` int(10) unsigned DEFAULT NULL,</pre> <p>technische Attribute für Ruby:</p> <pre>encrypted_password` varchar(64) NOT NULL DEFAULT '\$2a\$10\$qrVqv4bhCfd4Ld649LoS.xIC/ qK8GBdSXAS47ApqglevhPQL.H7K', `reset_password_token` varchar(128) DEFAULT NULL, `reset_password_sent_at` datetime DEFAULT NULL, `remember_created_at` datetime DEFAULT NULL, `sign_in_count` int(10) default 0, `current_sign_in_at` datetime DEFAULT NULL, `last_sign_in_at` datetime DEFAULT NULL, `current_sign_in_ip` varchar(16) DEFAULT NULL, `last_sign_in_ip` varchar(16) DEFAULT NULL, `created_at` datetime DEFAULT NULL, `updated_at` datetime DEFAULT NULL, PRIMARY KEY (Mnr) FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr) ENGINE=InnoDB CHARSET=latin1;</pre>	D-C	<p>2. Wenn eine Person als Mitglied (vorbehaltlich) aufgenommen und ein Konto mit Kontenklassenzuordnung (s. KKLVerlauf) eingerichtet wurde und nach 6 Wochen kein Zahlungseingang (Einzahlung) erfolgt ist, wird die OZBPerson in allen Tabellen (Person, OZBPerson, OZBKonto, EEKonto, KKLVerlauf, Bankverbindung, Adresse, Telefon) wieder gelöscht, d.h. es wird der Zustand vor Aufnahme wieder hergestellt.</p> <p>Klären, wofür das Attribut Gesperrt benötigt wird.</p>
<pre>CREATE TABLE IF NOT EXISTS `Mitglied` (`Mnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `RVDatum` date default NULL, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Mnr,GueltigVon), FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre> <pre>CREATE TABLE IF NOT EXISTS `Gesellschafter` (`Mnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `FALfNr` char(20), P `FASteuerNr` char(15), P `FAIdNr` char(15) Default NULL, P `Wohnsitzfinanzamt` varchar(50), P `NotarPnr` int(10) unsigned Default NULL, P `BeurkDatum` date, O `SachPnr` int(10) unsigned Default NULL, P PRIMARY KEY (Mnr,GueltigVon), FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr),</pre>	H	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Aussnahmen s. OZBPerson).</p> <p>Eine Mitgliedschaft kann (zum 31.12. eines Kalenderjahres) logisch beendet werden, z.B. wenn ein Mitglied die Rolle (s. Tabelle Person) ändert, wenn er z.B. Gesellschafter wird.</p> <p>Physisches Löschen, s. Ausnahmen (OZBPerson)</p>
	H	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Aussnahmen s. OZBPerson).</p> <p>Eine Gesellschaft kann beendet werden, wenn ein Gesellschafter die Rolle (s. Tabelle Person) ändert, wenn er z.B. Mitglied wird.</p> <p>Physisches Löschen nicht möglich</p>

II

<pre>'TeilnArt' enum ('a','e','u','l','<u>d</u>'), `SachPnr` int(10) unsigned, PRIMARY KEY (Pnr,Vnr), FOREIGN KEY (Vnr) REFERENCES Veranstaltung(Vnr), KEY Pnr (Pnr)</pre>	(D-C)	der übergeordneten Tabelle (Person) kein abhängiges Objekt existiert.
<pre>) ENGINE=InnoDB DEFAULT CHARSET=latin1; CREATE TABLE IF NOT EXISTS 'Partner' (`Mnr` int(10) unsigned NOT NULL , `GueltigVon` datetime NOT NULL , `GueltigBis` datetime NOT NULL , `MnrP` int(10) unsigned NOT NULL , `Berechtigung` char(1) NOT NULL DEFAULT '1' , `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Mnr,GueltigVon), FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr), KEY MnrO (MnrO)</pre>	H	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Ausnahmen s. OZBPerson).
<pre>) ENGINE=InnoDB DEFAULT CHARSET=latin1; CREATE TABLE IF NOT EXISTS 'Telefon' (`Pnr` int(10) unsigned NOT NULL , `LfdNr` tinyint(2) unsigned NOT NULL , `TelefonNr` varchar(15) DEFAULT NULL, `TelefonTyp` char(6) DEFAULT NULL, PRIMARY KEY (Pnr,LfdNr) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	I U D	Telefonverbindungen werden nicht historisiert und können daher eingegeben, geändert und auch physisch gelöscht werden, wenn sie nicht mehr zutreffen oder aus fachlicher Sicht gebraucht werden.
<pre>) ENGINE=InnoDB DEFAULT CHARSET=latin1; CREATE TABLE IF NOT EXISTS 'TanListe' (`Mnr` int(10) unsigned NOT NULL , `ListNr` tinyint(2) unsigned NOT NULL , `Status` enum ('n','d','a'), PRIMARY KEY (Mnr,ListNr) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	I (D-C)	Tanlisten werden nicht historisiert und werden von der MV generiert, ausgedruckt und den Mitgliedern auf An-for-de-rung zugeschickt (Status='n'). Mit Status n können Tanlisten mit allen abhängigen Objekten in Tabelle Tan auch physisch wieder gelöscht werden. Aktiviert das Mitglied die Tanliste (Status a) oder wurde sie bereits deaktiviert (d) dürfen die Liste und die abhängigen Tan nicht mehr gelöscht werden.
<pre>) ENGINE=InnoDB DEFAULT CHARSET=latin1; CREATE TABLE IF NOT EXISTS 'Tan' (`Mnr` int(10) unsigned NOT NULL , `ListNr` tinyint(2) unsigned NOT NULL , `TanNr` int(10) unsigned NOT NULL , `Tan` int(5) unsigned NOT NULL , `VerwendetAm` date DEFAULT NULL, `Status` enum ('o','x'), PRIMARY KEY (Mnr,ListNr,TanNr), FOREIGN KEY (Mnr, ListNr) REFERENCES Tanliste(Mnr, ListNr) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	D-C U-R	Abhängige Tabelle von Tanliste, s. dort. Die Attribute von Tan dürfen mit Ausnahme von VerwendetAm nicht geändert werden.
<pre>) ENGINE=InnoDB DEFAULT CHARSET=latin1; CREATE TABLE IF NOT EXISTS 'Bank' (`BLZ` int(8) unsigned NOT NULL, `BIC` char(11), <u>P</u> <u>Defalt</u> <u>#</u><u>#</u><u>#</u><u>#</u></pre>	I U D-R	Die Tabelle Bank wird nicht historisiert. Objekte können eingefügt und geändert werden.

<pre>) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT=''; CREATE TABLE IF NOT EXISTS `ZEKonto` (`KtoNr` int(5) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `Pgnr` tinyint(2) unsigned NOT NULL, `EEKtoNr` int(5) unsigned NOT NULL, `ZENR` char(10), `ZEAbDatum` date DEFAULT NULL, `ZEEndDatum` date DEFAULT NULL, `ZEBetrag` decimal(10,2) DEFAULT NULL, `Laufzeit` tinyint(4) unsigned NOT NULL, `ZahlModus` char(1) DEFAULT 'M', <u>GueltigVon</u>(<u>-</u>,<u>9,1</u>) `TilgRate` decimal(10,2) NOT NULL DEFAULT '0.00', `NachsparRate` decimal(10,2) NOT NULL DEFAULT '0.00', `KDURate` decimal(10,2) NOT NULL DEFAULT '0.00', `RDURate` decimal(10,2) NOT NULL DEFAULT '0.00', `ZEStatus` char(1) NOT NULL DEFAULT 'A', <u>SachPnr</u> int(10) unsigned DEFAULT NULL, `Kalk_Leihpunkte` int(11) DEFAULT NULL, `Tats_Leihpunkte` int(11) DEFAULT NULL, `Sicherung` varchar(200) DEFAULT NULL, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (KtoNr, GueltigVon), FOREIGN KEY (Pgnr) REFERENCES Projektgruppe (Pgnr), KEY EKtoNr (ZEKtoNr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	H	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln.</p> <p>Diese Tabelle wird nur von Programmprozessen bewirtschaftet (Insert/Update)</p> <p>Physisches Löschen nicht möglich</p>
<pre>CREATE TABLE IF NOT EXISTS `Buergschaft` (`Pnr_B` int(10) unsigned NOT NULL, `Mnr_G` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `ZENR` char(10) NOT NULL, `SichAbDatum` datetime DEFAULT NULL, `SichEndDatum` datetime DEFAULT NULL, `SichBetrag` decimal(10,2) DEFAULT NULL, `SichKurzbez` varchar(200) DEFAULT NULL, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Pnr_B, Mnr_G, GueltigVon), FOREIGN KEY (Mnr_G) REFERENCES OZBPerson (Mnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	H	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln.</p> <p>Physisches Löschen nicht möglich</p>
<pre>CREATE TABLE IF NOT EXISTS `KontenKlasse` (`KKL` char(1) NOT NULL, `KKLADEinrDatum` date NOT NULL, `Prozent` decimal(5,2) unsigned NOT NULL,) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	I	<p>Die Tabelle wird nicht historisiert. Objekte können eingefügt und geändert werden.</p> <p>Physisches Löschen nicht möglich.</p>

) ENGINE=InnoDB DEFAULT CHARSET=latin1;			
CREATE TABLE IF NOT EXISTS `Sonderberechtigung` (I		werden.
`ID` int(11) unsigned NOT NULL AUTO_INCREMENT,			Tabelle ist nicht historisiert
`Mnr` int(11) unsigned NOT NULL,			Eine neue Berechtigung für ein Mitglied kann eingegeben
`Email` varchar(40) NOT NULL,			werden
`Berechtigung` enum ('IT', 'MV', 'RW', 'ZE', 'OeA') NOT NULL,			Die Attribute können (mit Ausnahme von Mnr) verändert
PRIMARY KEY (`ID`)			werden.
FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr)			Berechtigungen können jederzeit zurückgenommen
) ENGINE=InnoDB DEFAULT CHARSET=latin1;	D		(physisch gelöscht werden)
INSERT INTO sonderberechtigung VALUES (0, 13, "tkienle@t-online.de", "IT");			
CREATE TABLE IF NOT EXISTS `Geschaeftsprozess` (I		
`ID` int(11) unsigned NOT NULL AUTO_INCREMENT,			Tabelle ist nicht historisiert
`Beschreibung` varchar(200) NOT NULL,			Eine neuer Geschäftsprozess kann eingegeben werden
`IT` tinyint(1) unsigned NOT NULL,			Die Attribut (Zuordnung zu AGs) kann geändert werden.
`MV` tinyint(1) unsigned NOT NULL,			Ein Geschäftsprozess kann nicht gelöscht werden.
`RW` tinyint(1) unsigned NOT NULL,			
`ZE` tinyint(1) unsigned NOT NULL,			
`OeA` tinyint(1) unsigned NOT NULL,			
PRIMARY KEY (`ID`)	U		
) ENGINE=InnoDB DEFAULT CHARSET=latin1;	D-R		
CREATE TABLE IF NOT EXISTS `Umlage` (U		
`Jahr` tinyint(4) unsigned NOT NULL,			Die Umlagentabelle gilt für ein Kalenderjahr d.h. die
`RDU` decimal(5,2) NOT NULL,			Werte für RDU und KDU werden am Anfang eines jeden
`KDU` decimal(5,2) NOT NULL <u>DEFAULT '0.00'</u> ,			Kalenderjahrs neu eingegeben
PRIMARY KEY (`Jahr`)			Derzeit ist diese Tabelle eine standalone-Tabelle und
) ENGINE=MyISAM DEFAULT CHARSET=latin1;			wird erst mit dem GP 'ZE berechnen' in das Datenmodell integriert.