

1. Liste der Attribute und deren Beschreibung/Definition

Person

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertevorrat	Beispiel
<code>`Pnr` int(10) unsigned NOT NULL,</code>	PK1	Nummer der Person	1 bis 99999	13
<code>`GueltigVon` datetime NOT NULL,</code>	PK2	Fachliche Gültigkeit von		2013-01-01:...
<code>`GueltigBis` datetime NOT NULL,</code>		Fachliche Gültigkeit bis		9999-12-31:...
<code>`Rolle` enum('G','M','P','S','F') DEFAULT NULL,</code>		Rolle der Person	G=Gesellschafter M=Mitglied P=Partner S=Student F=Fördermitglied	
<code>`Name` varchar(20) NOT NULL,</code>		Name der Person	Länge 30	Dresdner Bank
<code>`Vorname` varchar(15) NOT NULL DEFAULT '',</code>		Vorname der Person	Länge 20	Maria-Therese
<code>`Geburtsdatum` date DEFAULT NULL,</code>		Geburtsdatum	Gültiges Kalenderdatum	13.06.1983
<code>`EMail` varchar(255) DEFAULT NULL,</code>		Email-Adresse	50 Zeichen	IreneVoss@web.de
<code>`SperrKZ` tinyint(2) unsigned NOT NULL DEFAULT '0',</code>		0 = Logon freigegeben 1 = Logon gesperrt (nach 3 Tan-Falscheingaben)	0 oder 1	0
<code>`SachPnr` int(10) unsigned DEFAULT NULL,</code>		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 9999	13

Adresse

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertevorrat	Beispiel
<code>`Pnr` int(10) unsigned NOT NULL,</code>	PK1	Nummer der Person	1 bis 9999	13
<code>`GueltigVon` datetime NOT NULL,</code>	PK2	Fachliche Gültigkeit von		2013-01-01:...
<code>`GueltigBis` datetime NOT NULL,</code>		Fachliche Gültigkeit bis		9999-12-31:...
<code>`Strasse` varchar(50) NOT NULL,</code>		Name der Strasse		Rosenstrasse
<code>`Nr` varchar(10) DEFAULT NULL,</code>		Hausnummer		134A
<code>`PLZ` int(5) unsigned DEFAULT NULL,</code>		Postleitzahl	➔ ändern char(10)	6M4X3A
<code>`Ort` varchar(50) NOT NULL,</code>		Name des Postortes	char(20) 20 genügt	Etobicoke
<code>`Vermerk` varchar(100) DEFAULT NULL,</code>		Zustellvermerk	char(30) 30 genügt	bei Müller
<code>`SachPnr` int(10) unsigned DEFAULT NULL,</code>		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

OZBPerson

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertevorrat	Beispiel
`Mnr` int(10) unsigned NOT NULL,	PK	Mitgliedsnummer (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`UeberPnr` int(10) unsigned,		Nummer der Person über die das Mitglied zur o/ZB kam	1 bis 9999	150
`Passwort` varchar(35) DEFAULT NULL,		Passwort aus prod. System (md5-verschlüsselt)		
`email` varchar(255) NOT NULL,		email-Adresse der OZBPerson	char(20) ausreichend	IreneVoss@web.de
`PWAendDatum` date DEFAULT NULL,		Datum der letzten Passwortänderung	Kalenderdatum > Aufnahmedatum	20.06.2013
`Antragsdatum` date DEFAULT NULL,		Datum auf dem Aufnahmeantrag	Kalenderdatum < Aufnahmedatum	15.01.2013
`Aufnahmedatum` date DEFAULT NULL,		Datum des o/ZB-Kreisbeschlusses bzgl. Aufnahme des Mitglieds	Kalenderdatum	13.02.2013
`Austrittsdatum` date DEFAULT NULL,		Jahresende der Kündigung	Immer Jahresultimo	31.12.2013
`Schulungsdatum` date DEFAULT NULL,		Besuchsdatum des Schulungskurses	Kalenderdatum	15.03.2012
`Gesperrt` tinyint(2) unsigned NOT NULL DEFAULT '0',		Klären: Duplikat zu SerrKZ??		
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

Mitglied

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertevorrat	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:...
`RVDatum` date default NULL,		Datum der Rahmenvereinbarung für Mitglieder	Kalenderdatum > Aufnahmedatum	
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

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

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	

`Abschluss` char(20),		Bezeichnung des angestrebten Abschlusses		Dipl.-Pädagogin
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

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

Fördermitglied

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Pnr` int(10) unsigned NOT NULL,	PK1	Personennummer	1 bis 99999	13
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01:...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31:...
`Region` varchar(30),		Region, zu der das Fördermitglied gehört		Hamburg
`Foerderbeitrag` decimal(5,2),		Beitrag, zu dem sich das Fördermitglied verpflichtet hat		30,00
`MJ` Neues Attribut → einfügen		Zahlungs-/Abbuchungsturnus	m = monatlich j = jährlich	j
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

Veranstaltungart

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`VANr` int(11) unsigned NOT NULL,	PK	Nummer der Veranstaltungsart	1 bis 20	3
`VABezeichnung` varchar(30) COLLATE utf8_unicode_ci DEFAULT NULL,		Bezeichnung der Veranstaltungsart		Vortrag

Veranstaltung

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Vnr` int(11) unsigned NOT NULL,	PK	Nummer der Veranstaltung	1 bis 999	50
`VANr` int(11) unsigned NOT NULL,	FK*)	Nummer der Veranstaltungsart	1 bis 20	3
`VADatum` date NOT NULL,		Datum der Veranstaltung	Kalenderdatum	
`VAOrt` varchar(30) COLLATE utf8_unicode_ci DEFAULT		Ort der Veranstaltung		Vaihingen/Enz

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

*) FOREIGN KEY (VANr) REFERENCES Veranstaltungsart(VANr)

Teilnahme

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`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	l = 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 Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`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 'l',		Berechtigung des Partners	l = leseberechtigt v = voll	v

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

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

Telefon

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Pnr` int(10) unsigned NOT NULL ,	PK1	Personennummer	1 bis 99999	13
`LfdNr` tinyint(2) unsigned NOT NULL ,	PK2	Laufende Nummer des Anschlusses	1 bis max 5	2
`TelefonNr` varchar(15) DEFAULT NULL,		Vorwahl-Telefonnummer		0711-3000500
`TelefonTyp` char(6) DEFAULT NULL,		Typ des Telefonanschlusses	tel = Festnetztelefon fax = Faxanschluss mob = Mobiltelefon	mob

Tanliste

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Mnr` int(10) unsigned NOT NULL,	PK1	Mitgliedsnummer (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`ListNr` tinyint(2) unsigned NOT NULL ,	PK2	Nummer der ausgegebenen Tanliste	1 bis 999	2
`TanListDatum` date NOT NULL,		Ausgabedatum der Tanliste	Kalenderdatum > Aufnahmedatum	
`Status` enum ('n','d','a'),		Status der Tanliste	n = neu a = aktiviert d = deaktiviert	a

Tan

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Mnr` int(10) unsigned NOT NULL,	PK1 FK1*)	Mitgliedsnummer (gleicher Nummernkreis wie Pnr)	1 bis 9999	13
`ListNr` tinyint(2) unsigned NOT NULL ,	PK2 FK2	Nummer der ausgegebenen Tanliste	1 bis 999	2
`TanNr` int(10) unsigned NOT NULL,	PK3	Nummer der Tan	1 bis 40	12
`Tan` int(5) unsigned NOT NULL,		Tan 6-stellig	000001 bis 999999	428934

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

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

Bank

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`BLZ` int(8) unsigned NOT NULL,	PK	Bankleitzahl	8-stellig	12030000
`BIC` char(11),		Bank International Code	11-stellig	GENODES1RMA
`BankName` varchar(255) DEFAULT NULL		Name des Kreditinstituts	50-stellig	DKB Bank

Bankverbindung

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT ,	PK1	Identifikationsnummer	Wird generiert	20
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01:...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31:...
`Pnr` int(10) unsigned NOT NULL ,		Personennummer	1 bis 9999	13
`BankKtoNr` varchar(255) NOT NULL,		Kontonummer des o/ZB-Mitglieds bei der Bank		1011507447
`IBAN` char(20),		International Bank Account Number	22-stellig	DE83 6006 0000 5610 02
`BLZ` int(10) unsigned NOT NULL,	FK*)	Bankleitzahl	8-stellig	12030000
`SachPnr` int(10) unsigned DEFAULT NULL,		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
`KtoNr` int(5) unsigned NOT NULL ,	PK1	o/ZB-Kontonummer des Mitglieds	10001 bis 79999 und 90001 bis 99999	90038
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01:...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31:...
`Mnr` int(10) unsigned NOT NULL ,	FK1*	Mitgliedsnummer (gleicher	1 bis 9999	13

)	Nummernkreis wie Pnr)		
`KtoEinrDatum` date DEFAULT NULL,		Einrichtungsdatum des o/ZB-Kontos für das Mitglied	Kalenderdatum > Aufnahmedatum < erstes Belegdatum	
`Waehrung` char(3) NOT NULL DEFAULT 'STR',		Währung in der das o/ZB-Konto geführt werden soll	EUR STR CAR, usw. (eine exklusive Aufzählung)	STR
`WSaldo` decimal(10,2) DEFAULT NULL,		Währungssaldo zum Zeitpunkt der letzten Kontenbewegung		1513,12
`PSaldo` int(11) DEFAULT NULL,		Punktesaldo zum Zeitpunkt der letzten Kontenbewegung	ganzzahlige Werte	-24500,00
`SaldoDatum` date DEFAULT NULL,		Datum der letzten Kontenbewegung	Kalenderdatum <= BelegDatum der letzten Buchung zu Konto KtoNr	
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

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

EEKonto

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
` <u>KtoNr</u> ` int(5) unsigned NOT NULL ,	PK1	o/ZB-Kontonummer des Mitglieds für die laufende Rechnung (EE-Konto)	10001 bis 79999 und 90001 bis 99999	90038
` <u>GuelteigVon</u> ` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01:...
`GuelteigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31:...
`BankID` int(10) unsigned NOT NULL,			Identnummer der zugeordneten Bankverbindung	35
`Kreditlimit` decimal(5,2) NOT NULL DEFAULT '0.00',		Kreditlimit oder Verfügungsrahmen für dieses Konto	0 bis 10000,00 (Mikrokredite)	6000
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

Projektgruppe

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Pgnr` 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,		Enddatum 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	ganzzahlig	210423

		benötigte Leihpunkte gem. ZE-Vertrag		
`Tats_Leihpunkte` int(11) DEFAULT NULL,		nach Abrechnung der ZE tatsächlich benötigte Leihpunkte	Ganzzahlig	120345
`Sicherung` varchar(200) DEFAULT NULL,		Kurzbeschreibung der Sicherungsart	varchar(50) genügt	3 Teilbürgschaften über je 1000
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

*) FOREIGN KEY (Pgnr) REFERENCES Projektgruppe(Pgnr),

Bürgschaft

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`Pnr_B` int(10) unsigned NOT NULL ,	PK1	Nummer der Person die für eine ZE bürgt	1 bis 99999	13
`Mnr_G` int(10) unsigned NOT NULL ,	PK2 FK*)	Nummer der Gesellschafter-Person für deren ZE Pnr_B bürgt	1 bis 99999	13
`GueltigVon` datetime NOT NULL,	PK3	Fachliche Gültigkeit von		2013-01-01:...
`GueltigBis` datetime NOT NULL,		Fachliche Gültigkeit bis		9999-12-31:...
`ZENr` char(10) NOT NULL,		Vertragsnummer (ZE) für die gebürgt wird		D130115, bzw. ZE130115 in Stgt, bzw. Bonn
`SichAbDatum` datetime DEFAULT NULL,		Vertraglicher Beginn der Bürgschaft	Kalenderdatum >= ZEAbDatum	
`SichEndDatum` datetime DEFAULT NULL,		Vertragliches Ende der Bürgschaft	Kalenderdatum > SichAbDatum	
`SichBetrag` decimal(10,2) DEFAULT NULL,		Betrag mit dem gebürgt wird	Teilbetrag von ZEBetrag	1000,00
`SichKurzbez` varchar(200) DEFAULT NULL,		Kurzbezeichnung der Bürgschaftsart nach ZE-Vertrag		Einzelbürgschaft
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 9999	13

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

Kontenklasse

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`KKL` char(1) NOT NULL,	PK	Kontenklasse	A bis Z	A
`KKLEinrDatum` 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-` = Saldo vortrag	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 Buchungsgbelegs	Kalenderdatum	
`BuchDatum` date NOT NULL,		Datum des Buchungslaufs	Kalenderdatum	

		(an die Buchung gebucht wurde)		
`Buchungstext` varchar(50) NOT NULL,		Buchungstext		0038 Bos Karl
`Sollbetrag` decimal(10,2) NOT NULL,		Sollbetrag der Buchung		100,00
`Habenbetrag` decimal(10,2) NOT NULL,		Habenbetrag der Buchung		100,00
`SollKtoNr` int(5) unsigned NOT NULL,		Konto das belastet wird	10001 bis 79999 und 90001 bis 99999	20038
`HabenKtoNr` int(5) unsigned NOT NULL,		Konto auf dem gutgeschrieben wird	10001 bis 79999 und 90001 bis 99999	70038
`WSaldoAcc` decimal(10,2) NOT NULL,		akkumulierter Währungssaldo		5736,24
`Punkte` int(10),		Punktwert zur vorangehenden Buchung	Ganzzahlig (+/-)	365
`PSaldoAcc` int(10) NOT NULL,		akkumulierter Punktwert	Ganzzahlig (+/-)	-23453

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

BuchungOnline

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,	PK	Identnummer der Onlinebuchung	1 bis 100000	
`Mnr` int(10) unsigned NOT NULL ,	FK*)	Nummer des Mitglieds, das die Onlinebuchung ausgeführt hat	1 bis 9999	13
`UeberwDatum` date NOT NULL,		Datum der Online-Punkteüberweisung	Kalenderdatum	
`SollKtoNr` int(5) unsigned NOT NULL DEFAULT '0',		Konto das belastet wird	10001 bis 79999 und 90001 bis 99999	70038
`HabenKtoNr` int(5) unsigned NOT NULL DEFAULT '0'		Konto auf dem gutgeschrieben wird	10001 bis 79999 und 90001 bis 99999	70013
`Punkte` int(10) NOT NULL,		Anzahl der Punkte die überwiesen wurden	ganzzahlig	10000
`Tan` int(5) unsigned NOT NULL,		Tan, die für die Onlineüberweisung verwendet wurde	000001 bis 999999	428934
`BlockNr` tinyint(2) NOT NULL DEFAULT '-1',		Nummer des Buchungsblocks (Buchungen werden blockweise vom Server auf den Fibu-PC heruntergeladen)	1 bis 10000	15

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

Sonderberechtigung

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

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

Geschaeftsprozess

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
<code>`ID` int(11) unsigned NOT NULL AUTO INCREMENT,</code>	PK	Identnummer	1-1000	14
<code>`Beschreibung` varchar(200) NOT NULL,</code>		Kurzbeschreibung des Geschäftsprozesses		Einlage/Entnahme- konten bearbeiten
<code>`IT` tinyint(1) unsigned NOT NULL,</code>		GP für Informatiker	0 oder 1	
<code>`MV` tinyint(1) unsigned NOT NULL,</code>		GP für Mitgliederverw.	0 oder 1	
<code>`RW` tinyint(1) unsigned NOT NULL,</code>		GP für Rechnungswesen	0 oder 1	
<code>`ZE` tinyint(1) unsigned NOT NULL,</code>		GP für Zusatzentnahmen	0 oder 1	
<code>`OeA` tinyint(1) unsigned NOT NULL,</code>		GP für Öffentlichsarbeit	0 oder 1	

Umlage

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

2. Create Table Anweisungen

Regeln:

I	= Insert	Eingabe ist möglich
U	= Update	Änderung der Fachattribute möglich
D	= Delete	Physisches Löschen erlaubt
D-R	= on Delete Restrict	Löschen zurückweisen, wenn es in der übergeordneten Tabelle mindestens ein zugeordnetes Objekt gibt
D-C	= on Delete Cascade	Löschen zusammen mit allen zur gleichen Nummer gehörenden Objekten
U-R	= on Update Restrict	Änderungen zurückweisen, wenn es Abhängigkeiten gibt
H	= Historisierung	Einfügen/Ändern/Löschen durch Fortschreiben der Historie

SQL (erforderliche Änderungen sind rot markiert)	Regel	Beschreibung
<pre>CREATE TABLE IF NOT EXISTS `Person` (`Pnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `Rolle` enum('G','M','P','S','F') DEFAULT NULL, `Name` varchar(20) NOT NULL, `Vorname` varchar(15) NOT NULL DEFAULT '', `Geburtsdatum` date DEFAULT NULL, `Email` varchar(255) DEFAULT NULL, `SperrKZ` tinyint(2) unsigned NOT NULL DEFAULT '0', `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Pnr,GueltigVon)) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	<p>H</p> <p>D-C</p>	<p>Einfügen, Ändern und Löschen nur auf logischer Ebene nach den Historisierungsregeln.</p> <p>Löschen: Eine Person, die nicht</p> <ul style="list-style-type: none"> - o/ZB-Person (s. hierzu OZBerson) - Bürge - Fördermitglied - Partner (zu o/ZB-Person) <p>ist, kann auch physisch gelöscht werden. In diesem Fall gilt die Löschregel cascade, d.h. es werden in allen Tabellen alle Objekte mit dieser Pnr/Mnr physisch gelöscht</p>
<pre>CREATE TABLE IF NOT EXISTS `Adresse` (`Pnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `Strasse` varchar(50) DEFAULT NULL, `Nr` varchar(10) DEFAULT NULL, `PLZ` int(5) unsigned DEFAULT NULL, `Ort` varchar(50) DEFAULT NULL, `Vermerk` varchar(100) DEFAULT NULL, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Pnr,GueltigVon)) ENGINE=InnoDB DEFAULT CHARSET=latin1</pre>	<p>H</p> <p>(D-C)</p>	<p>Adressen werden historisiert und dürfen physisch nicht gelöscht werden.</p> <p>Ausnahmen s. unter Person und OZBPerson)</p>
<pre>CREATE TABLE IF NOT EXISTS `OZBPerson` (`Mnr` int(10) unsigned NOT NULL , `UeberPnr` int(10) unsigned , `Passwort` varchar(35) DEFAULT NULL, `email` varchar(255) NOT NULL, `PWAendDatum` date DEFAULT NULL, `Antragsdatum` date DEFAULT NULL,</pre>	<p>U</p> <p>D-R</p>	<p>OZBPersonen werden nicht historisiert und können eingefügt, geändert werden.</p> <p>Regel für physisches Löschen:</p> <ol style="list-style-type: none"> 1. Wenn es zu der Mnr eine Buchung gegeben hat. Darf die Person nicht mehr gelöscht werden (s. dazu Einführung des Fremdschlüssel Mnr in Tabelle

<pre> `Aufnahmedatum` date DEFAULT NULL, `Austrittsdatum` date DEFAULT NULL, `Schulungsdatum` date DEFAULT NULL, `Gesperrrt` tinyint(2) unsigned NOT NULL DEFAULT '0', `SachPnr` int(10) unsigned DEFAULT NULL, technische Attribute für Ruby: `encrypted_password` varchar(64) NOT NULL DEFAULT '\$2a\$10\$qGrVqv4bHcfd4Ld649LoS.xIc/ gK8GBdSXAS47AQpgleVhPQL.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)) ENGINE=InnoDB DEFAULT CHARSET=latin1; </pre>	D-C	<p>Buchung)</p> <p>2. Wenn eine Person als Mitglied (vorbehaltlich) aufgenommen und ein Konto mit Kontenklassenzuordnung (s. KKLVerlauf) eingerichtet wurde und nach 6 Wochen kein Zahlungseingang (Einlage) 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 Gesperrrt 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>	H (D)	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Ausnahmen 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>
<pre> CREATE TABLE IF NOT EXISTS `Gesellschafter` (`Mnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `FALfdNr` char(20), `FASteuerNr` char(15), `FAIdNr` char(15) Default NULL, `Wohnsitzfinanzamt` varchar(50), `NotarPnr` int(10) unsigned Default Null, `BeurkDatum` date, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Mnr,GueltigVon), FOREIGN KEY (Mnr) REFERENCES OZBPerson(Mnr), </pre>	H	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Ausnahmen 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>

<pre> `TeilnArt` enum ('a','e','u','l','m'), `SachPnr` int(10) unsigned, PRIMARY KEY (Pnr,Vnr), FOREIGN KEY (Vnr) REFERENCES Veranstaltung(Vnr), KEY Pnr (Pnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1; </pre>	(D-C)	<p>der übergeordneten Tabelle (Person) kein abhängiges Objekt existiert.</p> <p>Physisches Löschen, s. Ausnahmen (OZBPerson)</p>
<pre> CREATE TABLE IF NOT EXISTS `Partner` (`Mnr` int(10) unsigned NOT NULL , `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `MnrO` Pnr_P` int(10) unsigned NOT NULL , `Berechtigung` char(1) NOT NULL DEFAULT 'l' , `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Mnr,GueltigVon), FOREIGN KEY (Mnr) REFERENCES OZBPerson(Mnr), KEY MnrO (MnrO)) ENGINE=InnoDB DEFAULT CHARSET=latin1; </pre>	H	<p>Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Ausnahmen s. OZBPerson).</p> <p>Physisches Löschen nicht möglich</p>
<pre> 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	<p>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.</p>
<pre> 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)	<p>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.</p>
<pre> 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	<p>Abhängige Tabelle von TanListe, s. dort.</p> <p>Die Attribute von Tan dürfen mit Ausnahme von VerwendetAm nicht geändert werden.</p>
<pre> CREATE TABLE IF NOT EXISTS `Bank` (`BLZ` int(8) unsigned NOT NULL, `BIC` char(11), </pre>	I U D-R	<p>Die Tabelle Bank wird nicht historisiert.</p> <p>Objekte können eingefügt und geändert werden.</p>

<pre>`BankName` varchar(255) DEFAULT NULL, PRIMARY KEY (`BLZ`)) ENGINE=InnoDB DEFAULT CHARSET=latin1</pre>		Physisches Löschen nur, wenn sie in Tabelle Bankverbindung nicht referenziert wird.
<pre>CREATE TABLE IF NOT EXISTS `Bankverbindung` (`ID` int(10) unsigned NOT NULL AUTO_INCREMENT , `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `Pnr` int(10) unsigned NOT NULL , `BankKtoNr` varchar(255) NOT NULL, `IBAN` char(20), `BLZ` int(10) unsigned NOT NULL, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (ID, GueltigVon), FOREIGN KEY (Pnr) REFERENCES Person(Pnr), FOREIGN KEY (BLZ) REFERENCES Bank(BLZ)) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson
<pre>CREATE TABLE IF NOT EXISTS `OZBKonto` (`KtoNr` int(5) unsigned NOT NULL , `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `Mnr` int(10) unsigned NOT NULL , `KtoEinrDatum` date DEFAULT NULL, `Waehrung` char(3) NOT NULL DEFAULT 'STR', `WSaldo` decimal(10,2) DEFAULT NULL, `PSaldo` int(11) DEFAULT NULL, `SaldoDatum` date DEFAULT NULL, `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (KtoNr,GueltigVon) FOREIGN KEY (Mnr) REFERENCES OZBPerson(Mnr),) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson
<pre>CREATE TABLE IF NOT EXISTS `EEKonto` (`KtoNr` int(5) unsigned NOT NULL , `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `BankID` int(10) unsigned NOT NULL, `Kreditlimit` decimal(5,2) NOT NULL DEFAULT '0.00', `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (KtoNr,GueltigVon), FOREIGN KEY (BankID) REFERENCES Bankverbindung(ID), FOREIGN KEY (SachPnr) REFERENCES Person(Pnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1</pre>	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson
<pre>CREATE TABLE IF NOT EXISTS `ProjektGruppe` (`Pgnr` tinyint(2) unsigned, `ProjGruppenBez` varchar(50), PRIMARY KEY (`Pgnr`)</pre>	I U D-R	Die Tabelle wird nicht historisiert. Objekte können eingefügt und geändert werden. Physisches Löschen nur wenn sie in Tabelle ZE-Konto nicht benötigt wird.

<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', `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', `SachPnr` 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 EEKtoNr (EEKtoNr)) 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 , AbEinr `Datum` date NOT NULL , `Prozent` decimal(5,2) unsigned NOT NULL, </pre>	I U	<p>Die Tabelle wird nicht historisiert. Objekte können eingefügt und geändert werden.</p> <p>Physisches Löschen nicht möglich.</p>

PRIMARY KEY (KKL)) ENGINE=InnoDB DEFAULT CHARSET=latin1;		
CREATE TABLE IF NOT EXISTS `KKLVerlauf` (`KtoNr` int(5) unsigned NOT NULL , `KKLAbDatum` date NOT NULL , `KKL` char(1) NOT NULL , PRIMARY KEY (KtoNr,KKLAbDatum), FOREIGN KEY (KKL) REFERENCES kontenklasse(KKL)) ENGINE=InnoDB DEFAULT CHARSET=latin1;	D-R	Tabelle ist rein fachlich historisiert. Logisches und physisches Löschen nur möglich, wenn in der übergeordneten Tabelle (OZBKonto) kein abhängiges Objekt existiert.
CREATE TABLE IF NOT EXISTS `Buchung` (`BuchJahr` int(4) unsigned NOT NULL , `KtoNr` int(5) unsigned NOT NULL , `BnKreis` char(2) NOT NULL , `BelegNr` int(10) unsigned NOT NULL , `Typ` char(1) NOT NULL , `Belegdatum` date NOT NULL, `BuchDatum` date NOT NULL, `Mnr` int(10) unsigned NOT NULL, `Buchungstext` varchar(50) NOT NULL, `Sollbetrag` decimal(10,2) NOT NULL, `Habenbetrag` decimal(10,2) NOT NULL, `SollKtoNr` int(5) unsigned NOT NULL, `HabenKtoNr` int(5) unsigned NOT NULL, `WSaldoAcc` decimal(10,2) NOT NULL, `Punkte` int(10), `PSaldoAcc` int(10) NOT NULL, PRIMARY KEY (BuchJahr,KtoNr,BnKreis,BelegNr,Typ), KEY KtoNr(KtoNr), FOREIGN KEY (Mnr) REFERENCES OZBPerson(Mnr) FOREIGN KEY (KtoNr) REFERENCES OZBKonto(KtoNr), FOREIGN KEY (SollKtoNr) REFERENCES OZBKonto(KtoNr), FOREIGN KEY (HabenKtoNr) REFERENCES OZBKonto(KtoNr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;	I D D-R	Tabelle ist nicht historisiert Buchungen werden aus der Fibu durch das Web-Import--Pro-gramm paketweise (z.B. 1. Bis 15. August 2013) in die Tabelle eingefügt. Sind Fehler im letzten Buchungslauf aufgetreten und der Buchungslauf muss widerrufen und wiederholt werden, dann werden alle Buchungen zu einem oder mehreren BuchDatum physisch gelöscht und neu importiert. Solange es zu einer o/ZBPerson in der Tabelle eine Buchung gibt und daher das EEKonto zu dieser Person bebucht ist, darf diese Person mit allen Daten innerhalb der steuerlichen Aufbewahrungspflicht (12 Jahre) nicht mehr gelöscht werden.
CREATE TABLE IF NOT EXISTS `BuchungOnline` (`ID` int(10) unsigned NOT NULL AUTO_INCREMENT, `Mnr` int(10) unsigned NOT NULL , `UeberwDatum` date NOT NULL, `SollKtoNr` int(5) unsigned NOT NULL DEFAULT '0', `HabenKtoNr` int(5) unsigned NOT NULL DEFAULT '0', `Punkte` int(10) NOT NULL, `Tan` int(5) unsigned NOT NULL, `BlockNr` tinyint(2) NOT NULL DEFAULT '-1', PRIMARY KEY (ID), KEY Mnr (Mnr), FOREIGN KEY (Mnr) REFERENCES OZBPerson(Mnr)	I U	Tabelle ist nicht historisiert Überweist ein Mitglied an ein anderes Mitglied Punkte, so werden die zugehörigen Buchungen in dieser Tabelle gespeichert. Bei jedem Buchungslauf lädt die AG-RW die seit dem letzten Buchungslauf angefallenen Buchungen blockweise (BlockNr) vom Server auf den PC herunter und importiert die Buchungen in die Fibu. Die Attribute der Tabelle dürfen mit Ausnahme der Zuordnung zu einem Block (BlockNr) nicht verändert

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