

PRIMARY KEY (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 `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;	I	
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, Punkt bei Löschen '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), O 'PSaldoAcc' int(10) NOT NULL, O 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;	D	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.
CREATE TABLE IF NOT EXISTS `BuchungOnline` (`ID` int(10) unsigned NOT NULL AUTO_INCREMENT, `Mnr` int(10) unsigned NOT NULL , `ÜberwDatum` date NOT NULL , `SollKtoNr` int(5) unsigned NOT NULL DEFAULT '0' , `HabenKtoNr` int(5) unsigned NOT NULL DEFAULT '0' , `Punkte` int(10) NOT NULL, O `Tan` int(5) unsigned NOT NULL, P `BlockNr` tinyint(2) NOT NULL DEFAULT '-1', PRIMARY KEY (ID), KEY Mnr (Mnr), FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr)	I U	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 OZBPerson in der Tabelle eine Buchung gibt und daher das EEKonto zu dieser Person gebucht ist, darf diese Person mit allen Daten innerhalb der steuerlichen Aufbewahrungspflicht (12 Jahre) nicht mehr gelöscht werden.

			Physisches Löschen nur, wenn sie in Tabelle Bankverbindung nicht referenziert wird.
CREATE TABLE IF NOT EXISTS `Bankverbindung` PRIMARY KEY (`ID`)) ENGINE=InnoDB DEFAULT CHARSET=latin1	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln.	Physisches Löschen möglich s. Ausnahmen OZBPerson
'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;	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson
CREATE TABLE IF NOT EXISTS `OZBKonto` ('KtoNr` int(5) unsigned NOT NULL, 'GueltigBis` datetime NOT NULL, 'Mnr` int(10) unsigned NOT NULL, 'KtoEinrDatum` date DEFAULT NULL, 'Waehrung` char(3) NOT NULL DEFAULT 'EUR', '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;	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson
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', PRIMARY KEY (KtoNr, GueltigVon), FOREIGN KEY (BankID) REFERENCES Bankverbindung(ID), FOREIGN KEY (SachPnr) REFERENCES Person(Pnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1	H (D-C)	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln. Physisches Löschen möglich s. Ausnahmen OZBPerson
CREATE TABLE IF NOT EXISTS `ProjektGruppe` ('Pgnr` tinyint(2) unsigned, 'ProjGruppenBez` varchar(50), PRIMARY KEY ('Pgnr`)	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.	Die Tabelle wird nicht historisiert. Objekte können eingefügt und geändert werden. Physisches Löschen nur wenn sie in Tabelle ZE-Konto nicht benötigt wird.

KEY NotarPnr (NotarPnr), KEY Mnr (Mnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;	H	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Ausnahmen s. OZBPerson)
CREATE TABLE IF NOT EXISTS `Student` (`Mnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `AusbildBez` varchar(30), `InstitutName` varchar(30), `Studienort` varchar(30), `Studienbeginn` date, `Abschluss` char(20), `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY (Mnr, GueltigVon), FOREIGN KEY (Mnr) REFERENCES OZBPerson (Mnr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;	H	Eine studentische Mitgliedschaft kann beendet werden, wenn ein Student die Rolle (Tabelle Person) ändert, z.B. Mitglied/Gesellschafter wird.
CREATE TABLE IF NOT EXISTS `Foerdermitglied` (`Pnr` int(10) unsigned NOT NULL, `GueltigVon` datetime NOT NULL, `GueltigBis` datetime NOT NULL, `Region` varchar(30) Default NULL, `Foerderbeitrag` decimal(5,2) Default NULL, `MJ` enum ('m', 'j') `SachPnr` int(10) unsigned DEFAULT NULL, PRIMARY KEY Pnr (Pnr, GueltigVon)) ENGINE=InnoDB DEFAULT CHARSET=latin1;	H	Physisches Löschen nicht möglich
CREATE TABLE IF NOT EXISTS `Veranstaltungsart` (`VANr` int(11) unsigned NOT NULL AUTO_INCREMENT, `VABezeichnung` varchar(30) COLLATE utf8_unicode_ci DEFAULT NULL, PRIMARY KEY (VANr)) ENGINE=InnoDB DEFAULT CHARSET=latin1;	D	Einfügen/Ändern/Löschen nur logisch nach den Historisierungsregeln (Ausnahmen s. OZBPerson)
CREATE TABLE IF NOT EXISTS `Veranstaltung` (`Vnr` int(11) unsigned NOT NULL AUTO_INCREMENT, `VANr` int(11) unsigned NOT NULL, `VADatum` date NOT NULL, `VAOrt` varchar(30) COLLATE utf8_unicode_ci DEFAULT NULL, `SachPnr` int(10) unsigned, PRIMARY KEY (Vnr), FOREIGN KEY (VANr) REFERENCES Veranstaltungsart (VANr)) ENGINE=InnoDB DEFAULT CHARSET=latin1	D-R	Eine Fördermitgliedschaft kann beendet werden, wenn ein Fördermitglied die Rolle (Tabelle Person) ändert, z.B. Mitglied/Gesellschafter wird.
CREATE TABLE IF NOT EXISTS `Teilnahme` (`Pnr` int(10) unsigned NOT NULL, `Vnr` int(11) unsigned NOT NULL,	D-R	Physisches Löschen muss möglich sein, wenn das Fördermitglied nie einen der zugesagten Förderbeiträge überweist.
		Tabelle ist nicht historisiert.
		Logisches und physisches Löschen nur möglich, wenn in der nachgeordneten Tabelle (Teilnahme) kein abhängiges Objekt existiert.
		Logisches und physisches Löschen nur möglich, wenn in der nachgeordneten Tabelle (Teilnahme) kein abhängiges Objekt existiert.
		Logisches und physisches Löschen nur möglich, wenn in der nachgeordneten Tabelle (Teilnahme) kein abhängiges Objekt existiert.

Regeln:

I = Insert

U = Update

D = Delete

D-R = on Delete Restrict

D-C = on Delete Cascade

U-R = on Update Restrict

H = Historisierung

Eingabe ist möglich
Änderung der Fachattribute möglich

Physisches Löschen erlaubt

Löschen zurückweisen, wenn es in der übergeordneten Tabelle mindestens ein zugeordnetes Objekt gibt

Änderungen zusammen mit allen zur gleichen Nummer gehörenden Objekten

Einfügen/Ändern/Löschen durch Fortschreiben der Historie

SQL (erforderliche Änderungen sind rot markiert)

```
CREATE TABLE IF NOT EXISTS `Person` (
    `Pnr` int(10) unsigned NOT NULL, -
    `GuelttigVon` datetime NOT NULL, -
    `GuelttigBis` 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, -
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Regel

Beschreibung

Einfügen, Ändern und Löschen nur auf logischer Ebene nach den Historisierungsregeln.

Löschen:

Eine Person, die **nicht**

- o/ZB-Person (s. hierzu OZBPerson)

- Fördermitglied

- Partner (zu o/ZB-Person)

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

Regel

Beschreibung

Adressen werden historisiert und dürfen physisch nicht gelöscht werden.

Regel

Beschreibung

Ausnahmen s. unter Person und OZBPerson)

Regel

Beschreibung

(D-C)

```
CREATE TABLE IF NOT EXISTS `OZBPerson` (
    `Mnr` int(10) unsigned NOT NULL, -
    `UeberPnr` int(10) unsigned NOT NULL, -
    `Passwort` varchar(35) DEFAULT NULL, -
    `email` varchar(255) NOT NULL, -
    `PWAendDatum` date DEFAULT NULL, -
    `Antragsdatum` date DEFAULT NULL, -
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Regel

Beschreibung

OZBPersonen werden **nicht** historisiert und können eingefügt, geändert werden.

Regel

Beschreibung

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

P = Pflicht
O = Optional

	(an die Buchung gebucht wurde)	
'Buchungstext'	varchar(50) NOT NULL,	Buchungstext
'Sollbetrag'	decimal(10,2) NOT NULL,	Sollbetrag der Buchung
'Habenbetrag'	decimal(10,2) NOT NULL,	Habenbetrag der Buchung
'SollKtoNr'	int(5) unsigned NOT NULL,	Konto das belastet wird
'HabenKtoNr'	int(5) unsigned NOT NULL,	Konto auf dem gutgeschrieben wird
'WSaldoAcc'	decimal(10,2) NOT NULL,	akkumulierter Währungssaldo
'Punkte'	int(10),	Punktwert zur vorangehenden Buchung
'PSaldoAcc'	int(10) NOT NULL,	akkumulierter Punktwert

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

BuchungOnline

Create Table Anweisung

Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
PK	Identnummer der Onlinebuchung	1 bis 100000	
'Mnr'	int(10) unsigned NOT NULL , FK*)	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
'HabenKtoNr'	int(5) unsigned NOT NULL DEFAULT '0'	Konto auf dem gutgeschrieben wird	10001 bis 79999 und 90001 bis 99999
'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
'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)

<code>`Tats_Leihpunkte` int(11) DEFAULT NULL,</code>	benötigte Leihpunkte gem. ZE-Vertrag	
<code>`Sicherung` varchar(200) DEFAULT NULL,</code>	nach Abrechnung der ZE tatsächlich benötigte Leihpunkte	Ganzzahlig 120345
<code>`SachPnr` int(10) unsigned DEFAULT NULL,</code>	Kurzbeschreibung der Sicherungsart	varchar(50) genügt 3 Teilbürgschaften über je 1000
	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
<code>`Pnr_B` int(10) unsigned NOT NULL ,</code>	PK1	Nummer der Person die für eine ZE bürgt	1 bis 99999	13
<code>`Mnr_G` int(10) unsigned NOT NULL ,</code>	PK2 FK*)	Nummer der Gesellschafter- Person für deren ZE Pnr_B bürgt	1 bis 99999	13
<code>`GueltigVon` datetime NOT NULL,</code>	PK3	Fachliche Gültigkeit von		2013-01-01:...
<code>`GueltigBis` datetime NOT NULL,</code>		Fachliche Gültigkeit bis		9999-12-31:...
<code>`ZENr` char(10) NOT NULL,</code>		Vertragsnummer (ZE) für die gebürgt wird		D130115, bzw. ZE130115 in Stgt., bzw. Bonn
<code>`SichAbDatum` datetime DEFAULT NULL,</code>		Vertraglicher Beginn der Bürgschaft	Kalenderdatum >=	
<code>`SichEndDatum` datetime DEFAULT NULL,</code>		Vertragliches Ende der Bürgschaft	Kalenderdatum > SichAbDatum	
<code>`SichBetrag` decimal(10,2) DEFAULT NULL,</code>		Betrag mit dem gebürgt wird	Teilbetrag von	1000,00
<code>`SichKurzbez` varchar(200) DEFAULT NULL,</code>		Kurzbezeichnung der Bürgschaftsart nach ZE- Vertrag		Einzelbürgschaft
<code>`SachPnr` int(10) unsigned DEFAULT NULL,</code>		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 9999	13

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

Kontenklasse

Stand: 19.07.2013

*

		Nummernkreis wie Pnr)	
		Einrichtungsdatum des o/ ZB-Kontos für das Mitglied	Kalenderdatum > Auf- nahmedatum < 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)
`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	-24500,00
`SaldoDatum` date DEFAULT NULL,		Datum der letzten Kontenbewegung	
`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
`GueltigVon` datetime NOT NULL,	PK2	Fachliche Gültigkeit von		2013-01-01:...
`GueltigBis` 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 100000,00 (Mikrokredite)	6000
`SachPnr` int(10) unsigned DEFAULT NULL,		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13

<code>'SachPnr` int(10) unsigned DEFAULT NULL,</code>	Personennummer des Sachbearbeiters, der eine Änderung durchführt	berechtigtigt 1 bis 99999	13
---	--	------------------------------	----

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

Telefon

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

Tanliste

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

Tan

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

<code>'Abschluss'</code> char(20),		Bezeichnung des angestrebten Abschlusses	1 bis 99999	Dipl.-Pädagogin
<code>'SachPnr'</code> 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
<code>'Pnr'</code> int(10) unsigned NOT NULL,	PK1	Personennummer	1 bis 99999	13
<code>'GueltigVon'</code> datetime NOT NULL,	PK2	Fachliche Gültigkeit von	2013-01-01:...	
<code>'GueltigBis'</code> datetime NOT NULL,		Fachliche Gültigkeit bis	9999-12-31:...	
<code>'Region'</code> varchar(30),		Region, zu der das Fördermitglied gehört	Hamburg	
<code>'Foerderbeitrag'</code> decimal(5,2),		Beitrag, zu dem sich das Fördermitglied verpflichtet hat	30,00	
<code>'MJ'</code>		Zahlungs-/Abbuchungsturnus	m = monatlich j = jährlich	j
Neues Attribut → einfügen		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13
<code>'SachPnr'</code> int(10) unsigned DEFAULT NULL,				

Veranstaltungart

Create Table Anweisung	Keys	Fachliche Beschreibung	Wertebereich/Vorrat	Beispiel
<code>'VANL'</code> int(11) unsigned NOT NULL,	PK	Nummer der Veranstaltungsart	1 bis 20	3
<code>'VABeziechnung'</code> varchar(30) COLLATE utf8_unicode_ci		Bezeichnung der Veranstaltungssart		Vortrag
DEFAULT NULL,				

Veranstaltung

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

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
`PWEndDatum` 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-Aufnahmeschlusses 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		Klären: Duplikat zu SerrKZ??		
DEFAULT '0',		Personennummer des Sachbearbeiters, der eine Änderung durchführt	1 bis 99999	13
`SachPnr` int(10) unsigned DEFAULT NULL,				

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)