Tabellen (5)

Name	Typ	Schema
Benutzer		CREATE TABLE "Benutzer" ("ID" INTEGER NOT NULL, "username" TEXT NOT NULL, "password_hash" TEXT NOT NULL, "is_admin" INTEGER NOT NULL DEFAULT 0, PRIMARY KEY("ID" AUTOINCREMENT))
ID	INTEGER	"ID" INTEGER NOT NULL
username	TEXT	"username" TEXT NOT NULL
password_hash	TEXT	"password_hash" TEXT NOT NULL
is_admin	INTEGER	"is_admin" INTEGER NOT NULL DEFAULT 0
LOHNABRECHNUNG		CREATE TABLE LOHNABRECHNUNG (LOHN_ID INTEGER PRIMARY KEY AUTOINCREMENT, -- unique row ID PERS_ID INTEGER NOT NULL, -- link to PERSON MONAT TEXT NOT NULL, -- e.g., '2025-09' for the payroll month STUNDENSATZ REAL, -- hourly rate WOCHENSTUNDEN REAL, -- weekly working hours BRUTTO REAL, -- monthly gross salary MEHRSTUNDEN0 REAL DEFAULT 0, MEHRSTUNDEN25 REAL DEFAULT 0, MEHRSTUNDEN50 REAL DEFAULT 0, ÜBERSTUNDEN50 REAL DEFAULT 0, ÜBERSTUNDEN100 REAL DEFAULT 0, SONDERZAHLUNGEN REAL DEFAULT 0, SACHBEZUG REAL DEFAULT 0, DIÄTEN REAL DEFAULT 0, REISEKOSTEN REAL DEFAULT 0, FREIBETRAGSBESCHEID REAL DEFAULT 0, PENDLERPAUSCHALE REAL DEFAULT 0, PENDLEREURO REAL DEFAULT 0, ANZAHL_KINDER_AVAB INTEGER DEFAULT 0, ANSPRUCH_FABO REAL DEFAULT 0, GEWERKSCHAFTSMITGLIED BOOLEAN DEFAULT 0, JAHRESSECHSTEL REAL DEFAULT 0, FOREIGN KEY (PERS_ID) REFERENCES PERSON(PERS_ID))
LOHN_ID	INTEGER	"LOHN_ID" INTEGER
PERS_ID	INTEGER	"PERS_ID" INTEGER NOT NULL
MONAT	TEXT	"MONAT" TEXT NOT NULL
STUNDENSATZ	REAL	"STUNDENSATZ" REAL
WOCHENSTUNDEN	REAL	"WOCHENSTUNDEN" REAL
BRUTTO	REAL	"BRUTTO" REAL
MEHRSTUNDEN0	REAL	"MEHRSTUNDEN0" REAL DEFAULT 0
MEHRSTUNDEN25	REAL	"MEHRSTUNDEN25" REAL DEFAULT 0
MEHRSTUNDEN50	REAL	"MEHRSTUNDEN50" REAL DEFAULT 0
ÜBERSTUNDEN50	REAL	"ÜBERSTUNDEN50" REAL DEFAULT 0
ÜBERSTUNDEN100	REAL	"ÜBERSTUNDEN100" REAL DEFAULT 0
SONDERZAHLUNGEN	REAL	"SONDERZAHLUNGEN" REAL DEFAULT 0
SACHBEZUG	REAL	"SACHBEZUG" REAL DEFAULT 0
DIÄTEN	REAL	"DIÄTEN" REAL DEFAULT 0
REISEKOSTEN	REAL	"REISEKOSTEN" REAL DEFAULT 0
FREIBETRAGSBESCHEID	REAL	"FREIBETRAGSBESCHEID" REAL DEFAULT 0
PENDLERPAUSCHALE	REAL	"PENDLERPAUSCHALE" REAL DEFAULT 0
PENDLEREURO	REAL	"PENDLEREURO" REAL DEFAULT 0
ANZAHL_KINDER_AVAB	INTEGER	"ANZAHL_KINDER_AVAB" INTEGER DEFAULT 0
ANSPRUCH_FABO	REAL	"ANSPRUCH_FABO" REAL DEFAULT 0
GEWERKSCHAFTSMITGLIED	BOOLEAN	"GEWERKSCHAFTSMITGLIED" BOOLEAN DEFAULT 0
JAHRESSECHSTEL	REAL	"JAHRESSECHSTEL" REAL DEFAULT 0
MITARBEITER		CREATE TABLE "MITARBEITER" ("EMPL_ID" integer, "PERS_ID" INTEGER, "EMPL_ENTRYDATE" TEXT NOT NULL, "EMPL_BRUTTOGEHALT" integer NOT NULL, "EMPL_EXITDATE" TEXT, "EMPL_VALID_FROM" TEXT DEFAULT (datetime('now')), "EMPL_VALID_TO" INTEGER DEFAULT '2100-01-01 12:12:12', PRIMARY KEY("EMPL_ID"), FOREIGN KEY("PERS_ID") REFERENCES "PERSON"("PERS_ID"))
EMPL_ID	integer	"EMPL_ID" integer
PERS_ID	INTEGER	"PERS_ID" INTEGER
EMPL_ENTRYDATE	TEXT	"EMPL_ENTRYDATE" TEXT NOT NULL
EMPL_BRUTTOGEHALT	integer	"EMPL_BRUTTOGEHALT" integer NOT NULL
EMPL_EXITDATE	TEXT	"EMPL_EXITDATE" TEXT
EMPL_VALID_FROM	TEXT	"EMPL_VALID_FROM" TEXT DEFAULT (datetime('now'))

Name	Typ	Schema
EMPL_VALID_TO	INTEGER	"EMPL_VALID_TO" INTEGER DEFAULT '2100-01-01 12:12:12'
PERSON		CREATE TABLE "PERSON" ("PERS_ID" integer NOT NULL, "PERS_SURNAME" text NOT NULL, "PERS_FIRSTNAME" text NOT NULL, "PERS_BIRTHDATE" text NOT NULL, "PERS_STREET" text, "PERS_HOUSENR" integer, "PERS_FLOOR" text, "PERS_ZIP" integer, "PERS_PLACE" text, "PERS_REC_ID" INTEGER NOT NULL, PRIMARY KEY("PERS_REC_ID"))
PERS_ID	integer	"PERS_ID" integer NOT NULL
PERS_SURNAME	text	"PERS_SURNAME" text NOT NULL
PERS_FIRSTNAME	text	"PERS_FIRSTNAME" text NOT NULL
PERS_BIRTHDATE	text	"PERS_BIRTHDATE" text NOT NULL
PERS_STREET	text	"PERS_STREET" text
PERS_HOUSENR	integer	"PERS_HOUSENR" integer
PERS_FLOOR	text	"PERS_FLOOR" text
PERS_ZIP	integer	"PERS_ZIP" integer
PERS_PLACE	text	"PERS_PLACE" text
PERS_REC_ID	INTEGER	"PERS_REC_ID" INTEGER NOT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"

Indizes (0)

Name	Typ	Schema

Ansichten (0)

Name	Typ	Schema

Trigger (2)

Name	Typ	Schema
trg_update_mitarbeiter		CREATE TRIGGER trg_update_mitarbeiter BEFORE UPDATE ON MITARBEITER FOR EACH ROW WHEN OLD.EMPL_VALID_TO = '2100-01-01 12:12:12' -- only the current row BEGIN -- 1. Close the old row by setting EMPL_VALID_TO to now UPDATE MITARBEITER SET EMPL_VALID_TO = datetime('now') WHERE EMPL_ID = OLD.EMPL_ID; -- 2. Insert a new row with updated values and a new validity range INSERT INTO MITARBEITER (PERS_ID, EMPL_ENTRYDATE, EMPL_BRUTTOGEHALT, EMPL_EXITDATE, EMPL_VALID_FROM, EMPL_VALID_TO) VALUES (NEW.PERS_ID, NEW.EMPL_ENTRYDATE, NEW.EMPL_BRUTTOGEHALT, NEW.EMPL_EXITDATE, datetime('now'), '2100-01-01 12:12:12'); -- 3. Prevent the original UPDATE from overwriting the old row SELECT RAISE(IGNORE); END
trg_update_person		CREATE TRIGGER trg_update_person BEFORE UPDATE ON PERSON FOR EACH ROW WHEN OLD.VALID_TO = '2100-01-01 12:12:12' -- only current row BEGIN -- 1. Close the old record UPDATE PERSON SET VALID_TO = datetime('now') WHERE PERS_REC_ID = OLD.PERS_REC_ID; -- 2. Insert a new record with updated values INSERT INTO PERSON (PERS_ID, PERS_SURNAME, PERS_FIRSTNAME, PERS_BIRTHDATE, PERS_STREET, PERS_HOUSENR, PERS_FLOOR, PERS_ZIP, PERS_PLACE, VALID_FROM, VALID_TO) VALUES (OLD.PERS_ID, -- keep same employee ID NEW.PERS_SURNAME, NEW.PERS_FIRSTNAME, NEW.PERS_BIRTHDATE, NEW.PERS_STREET, NEW.PERS_HOUSENR, NEW.PERS_FLOOR, NEW.PERS_ZIP, NEW.PERS_PLACE, datetime('now'), -- new valid from '2100-01-01 12:12:12' -- new valid to); -- 3. Prevent the original update from overwriting the old record SELECT RAISE(IGNORE); END