

Händler-Datenbank (SQL-Beispiel)

Wechseln zu: [Navigation](#), [Suche](#)

Dieser Artikel erfüllt die [GlossarWiki-Qualitätsanforderungen](#):

Korrektheit: 4 (größtenteils überprüft)	Umfang: 5 (wesentliche Fakten vorhanden)	Quellenangaben : 5 (vollständig vorhanden)	Quellenarten: 5 (ausgezeichnet)	Konformität: 5 (ausgezeichnet)
---	--	---	---	--

Das Händler-liefert-Ware-Beispiel ist das klassische Beispiel, um viele Konzepte von [Datenbanksystemen](#) und [SQL](#) zu demonstrieren. Daher wird dieses Beispiel auch in der Vorlesung [Multimedia-Datenbanksysteme](#) verwendet.

Inhaltsverzeichnis

- [1 Datenmodell/ER-Diagramm](#)
- [2 Datenbankschema](#)
- [3 Datenbankschema \(SQL\)](#)
- [4 Beispieldaten](#)
- [5 SQL-Beispiele](#)
- [6 Quellen](#)
- [7 Siehe auch](#)

1 Datenmodell/ER-Diagramm



Datenmodell in [UML](#)-Notation

Ein Händler liefert bestimmte Waren. Er kann durch die Händler-ID `h_id` eindeutig identifiziert werden und besitzt weitere Attribute, wie Name, Ortschaft etc. Die zugehörigen **Entities** (**Objekte**) werden durch den **Entity-Typ** (die **Klasse**) `haendler` definiert.

Welche Waren es gibt, ist durch einen weiteren Entity-Typ festgelegt: `ware`. Eine Ware ist durch die Waren-ID `w_id` eindeutig festgelegt. Weitere Attribute, wie Typ (Gemüse, Fleisch, Wurst etc.) und genaue Bezeichnung (Kohlrabi, Rinderlende, Cervelat etc.), beschreiben die jeweilige Ware näher.

Zwischen den Entity-Typen `haendler` und `ware` besteht eine Beziehung: `liefert`. Diese legt fest, welcher Händler welche Ware zu welchen Konditionen liefert. Ein Lieferant liefert eine Ware normalerweise zu einem bestimmten Preis. Zusätzlich kann eine gewisse Lieferzeit (in Tagen) angegeben werden. Es ist auch erlaubt, dass ein Händler ein und dieselbe Ware zu mit unterschiedlichen Preisen unterschiedlich schnell liefert. Um dies zu ermöglichen, muss das Attribut `l_preis` zum originären Primärschlüssel (`h_id`, `w_id`) der Beziehung `liefert` hinzugefügt

werden.



Datenmodell in ER-Notation

2 Datenbankschema

Aus dem obigen ER-Diagramm wird nach dem in [Kowarschick \(MMDB-Skript\)](#) beschriebenen Verfahren ein Datenbankschema erstellt. Ein Stern (*) bedeutet dabei **NULLABLE**.

```
haendler: h_id: INTEGER, h_name: VARCHAR(50), h_ortschaft*: VARCHAR(30)
          {PK: s_id}
          {UNIQUE: h_name, h_ortschaft}

ware:     w_id: INTEGER, w_typ: VARCHAR(30), w_bezeichnung: VARCHAR(30)
          {PK: w_id}
          {UNIQUE: w_typ, w_bezeichnung}

liefert:  h_id: INTEGER, w_id: INTEGER, l_preis: NUMERIC(8,2),
          l_lieferzeit*: SMALLINT
          {PK: s_id, w_id, l_preis}
          {FK: h_id -> haendler: h_id}
          {FK: w_id -> ware: w_id}
          {l_lieferzeit > 0}
```

3 Datenbankschema (SQL)

Aus dem obigen Datenbankschema leitet sich folgende SQL-**DDL**-Befehle ab (Syntax getestet mit [SQLite](#) und [PostgreSQL](#)):

```
/* Alte Tabellen (in der richtigen Reihenfolge) löschen: */
```

```
DROP TABLE IF EXISTS liefert;  
DROP TABLE IF EXISTS ware;  
DROP TABLE IF EXISTS haendler;
```

```
/* Tabellen erstellen */
```

```
CREATE TABLE haendler  
  (h_id          INTEGER      NOT NULL,  
   h_name        VARCHAR(30) NOT NULL,  
   h_ortschaft   VARCHAR(50),  
  
   CONSTRAINT pk_haendler  
     PRIMARY KEY (h_id),  
  
   CONSTRAINT unique_haendler_name_ortschaft  
     UNIQUE (h_name, h_ortschaft)  
  );
```

```
CREATE TABLE ware  
  (w_id          INTEGER      NOT NULL,  
   w_typ         VARCHAR(30) NOT NULL DEFAULT 'Sonstiges',  
   w_bezeichnung VARCHAR(50) NOT NULL,  
  
   CONSTRAINT pk_ware  
     PRIMARY KEY (w_id),  
  
   CONSTRAINT unique_haendler_typ_bezeichnung  
     UNIQUE (w_typ, w_bezeichnung)  
  );
```

```
CREATE TABLE liefert  
  (h_id          INTEGER      NOT NULL,  
   w_id          INTEGER      NOT NULL,  
   l_preis       NUMERIC(8,2) NOT NULL,  
   l_lieferzeit  SMALLINT,    /* Tage */  
  
   CONSTRAINT pk_liefert  
     PRIMARY KEY (h_id, w_id, l_preis),  
  
   CONSTRAINT fk_liefert_haendler  
     FOREIGN KEY (h_id) REFERENCES haendler (h_id),  
  
   CONSTRAINT fk_liefert_ware  
     FOREIGN KEY (w_id) REFERENCES ware (w_id),  
  
   CONSTRAINT check_liefert_lieferzeit  
     CHECK (l_lieferzeit > 0)
```

```
);
```

4 Beispieldaten

```
INSERT INTO haendler(h_id, h_name, h_ortschaft)
```

```
VALUES
```

```
(1, 'Maier', 'Königsbrunn'),  
(2, 'Müller', 'Königsbrunn'),  
(3, 'Maier', 'Augsburg'),  
(4, 'Huber', NULL),  
(5, 'Schmidt', 'Hamburg')
```

```
;
```

```
INSERT INTO ware(w_id, w_typ, w_bezeichnung)
```

```
VALUES
```

```
(1, 'CPU', 'Pentium IV 3,8'),  
(2, 'CPU', 'Celeron 2,6'),  
(3, 'CPU', 'Athlon XP 3000+'),  
(4, 'RAM', 'SDRAM 1GB'),  
(5, 'Sonstiges', 'Eieruhr')
```

```
;
```

```
INSERT INTO liefert(h_id, w_id, l_preis, l_lieferzeit)
```

```
VALUES
```

```
(1, 1, 200.00, 1),  
(1, 1, 194.00, 6),  
(1, 2, 100.00, NULL),  
(1, 3, 150.00, 7),  
(1, 4, 10.00, 1),  
(1, 5, 5.00, 1),  
(2, 1, 160.00, NULL),  
(2, 1, 190.00, 1),  
(2, 2, 180.00, NULL),  
(2, 3, 170.00, 4),  
(3, 1, 195.00, 2),  
(3, 2, 190.00, 1),  
(4, 1, 150.00, 3),  
(4, 3, 180.00, 5),  
(4, 3, 199.99, 1)
```

```
;
```

5 SQL-Beispiele

[Identität](#)

[Projektion](#)

[Selektion](#)

[Join](#)

[Unteranfragen](#)

6 Quellen

Kowarschick (MMDB-Skript): [Wolfgang Kowarschick](#); Vorlesung Multimedia-Datenbanksysteme – Sommersemester 2018; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2018; Quellengüte: 4 (Skript)

Kowarschick (MMDB): [Wolfgang Kowarschick](#); Vorlesung „Multimedia-Datenbanksysteme“; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2016; Quellengüte: 3 (Vorlesung)

7 Siehe auch

[Händler-Datenbank \(SQL-Beispiel\)/Identität](#)

[Händler-Datenbank \(SQL-Beispiel\)/Projektion](#)

[Händler-Datenbank \(SQL-Beispiel\)/Selektion](#)

[Händler2-Datenbank \(SQL-Beispiel\)](#) (komplexere Version dieser Datenbank)

Kategorien:

[PostgreSQL-Beispiel](#)

[Praktikum:MMDB](#)

Diese Seite wurde zuletzt am 10. Oktober 2019 um 15:23 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

