	Thema: SQL - DDL		
	Klasse: IFA12B	Datum: 09.11.2021	Fach: AWP

Aus der Lagerhalle in die Datenbank

Nachdem Sie das ER-Modell für die neue Datenbank erstellt haben, diese in das Relationenschema überführt und daraus die Datenbanktabellen abgeleitet haben, ist es nun an der Zeit, die eigentliche Datenbank zu erstellen. Die Projektverantwortlichen bei der LIFtOff GmbH haben sich für den Einsatz von MariaDB entschieden.

Aus dem bereits vorliegenden Relationenschema müssen nun die passenden SQL-Statements abgeleitet werden, mit denen sich die benötigten Tabellen, Spalten, Primär- und Fremdschlüssel erstellen lassen.



Hubstapler (ID, Hersteller, Modell, Seriennummer, Akkustand, Maximallast, Hubhöhe)

Ladestation (ID, Hersteller, Seriennummer, Ladeplatz)

Regalreihe (ID, Nummer, Höhe, Halle_ID)

Halle (ID, Nummer)

Hubstapler_bedient_Regalreihe (ID, Hubstapler_ID, Regalreihe_ID, Arbeitstag)

Ladevorgang (ID, Hubstapler_ID, Ladestation_ID, Beginn, Ende)


SQL: Data Definition Language (DDL)

Die Datenbanksprache **SQL** (Structured Query Language) unterteilt man in mehrere Sprachkomponenten:

- DDL Data Definition Language
- DQL Doctrine Query Language
- DML Data Manipulation Language
- DCL Data Control Language

Manche Quellen erwähnen noch eine weitere Komponente:

- TCL Transaction Control Language

	Thema: SQL - DDL		
	Klasse: IFA12B	Datum:	Fach: AWP

Recherchieren Sie zu den nachfolgend angegebenen zentralen Schlüsselwörtern der DDL. Machen Sie sich mit der Funktionalität der Schlüsselwörter, der zugehörigen Syntax und den zugehörigen Datentypen und Parametern vertraut.

- CREATE table halle (ID int primary key, Nummer int not null,
foreign key(liegenschaft_ID) references liegenschaft(ID));

- DROP


- ALTER

Sollte Ihnen noch Zeit übrig bleiben, können Sie diese nutzen um sich auch zu diesen Schlüsselwörtern zu informieren:

- TRUNCATE

- COMMENT

- RENAME

	Thema: SQL - DDL		
	Klasse: IFA12B	Datum:	Fach: AWP

Recherchehinweise

<https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>



https://de.wikipedia.org/wiki/Data_Definition_Language



Datenintegrität

Für den Betrieb einer Datenbank ist es von großer Bedeutung, dass die gespeicherten Daten jederzeit integer, also **korrekt** und **vollständig** sind.

Daten dürfen als korrekt betrachtet werden, wenn sie den Sachverhalt der realen Welt, den sie abbilden, korrekt wiedergeben und wenn sie frei von logischen Widersprüchen sind.

Die Vollständigkeit der Daten ist beeinträchtigt, wenn Daten ungewollt gelöscht oder gar nicht erst gespeichert wurden.

Für Relationale Datenbanken lassen sich zudem noch eine Reihe von **Integritätsbedingungen** formulieren:

- Entitätsintegrität: Jeder Datensatz ist eindeutig über einen Primärschlüssel identifizierbar.
- referenzielle Integrität: Zu jedem Fremdschlüsselwert existiert auch ein Datensatz mit dem entsprechenden Primärschlüssel. (keine Karteileichen)
- Wertebereichsintegrität: Die Werte eines Attributes dürfen nicht außerhalb des vorgegebenen Wertebereiches liegen.


Von besonderer praktischer Bedeutung ist die referenzielle Integrität. Um diese zu gewährleisten werden mit Hilfe der DDL sogenannte Constraints definiert. (vgl. ON DELETE RESTRICT/CASCADE)

Recherchehinweise

Eine kurze und anschauliche Erläuterung zur Umsetzung in SQL finden Sie in diesem Thread:

<https://itectec.com/database/thesql-difference-between-on-delete-cascade-on-update-cascade-in-thesql/>



	Thema: SQL - DDL		
	Klasse: IFA12B	Datum:	Fach: AWP

Umsetzung

Nachdem Sie sich ausreichend zur DDL informiert haben, sollten Sie in der Lage sein, aus dem eingangs gezeigten Relationenschema SQL-Anweisungen abzuleiten, mit denen die benötigte Datenbank erzeugt werden kann.

CREATE Database LiftOff;

CREATE Table Hubstapler (

ID bigint primary key,

Hersteller varchar(100) not null, Seriennummer varchar(100) not null, Akkustand byte,

Maximallast varchar(100), Hubhöhe varchar(100))

CREATE Table Ladestationen (

ID bigint primary key, Hersteller varchar(100) not null,

Seriennummer varchar(100) not null, Ladeplatz varchar(100))

CREATE Table Regalreihe (

ID bigint primary key,

Nummer int not null, Höhe varchar(100),

foreign key(Halle_ID) references Hallen(ID))

Create Table Hallen (

ID bigint primary key, Nummer int not null)

Create Table Hubstapler_bedient_Regalreihe (

ID bigint primary key,

foreign key(Hubstapler_ID) references Hubstapler(ID),

foreign key(Regalreihe_ID) references Regalreihe(ID),

Arbeitstag bigint)

CREATE Table Ladevorgang (

ID bigint primary key,

foreign key(Hubstapler_ID) references Hubstapler(ID),

foreign key(Ladestation_ID) references Ladestationen(ID),

Beginn datetime2(0), Ende datetime2(0))