

1 Tabellen erstellen und das Entity-Relationship-Modell

1.1 CREATE TABLE

Aufgabe 9.1 Das erste CREATE TABLE verstehen

Erklären Sie was der folgende SQL-Befehl genau tut. Ziehen Sie dabei das Glossar im folgenden Abschnitt [Glossar zu SQL-Create](#) (oder eine SQL-Referenz Ihres Vertrauens) zu Rate.

Tipp: Es ist vielleicht zunächst hilfreich herauszufinden, wie die neue Tabelle und ihre Spalten heißen werden.

```
CREATE TABLE photos (
  id          INT(4) UNSIGNED NOT NULL      AUTO_INCREMENT,
  user_id     INT(4) UNSIGNED NOT NULL,
  description VARCHAR(255)    NOT NULL,
  url         VARCHAR(255)    NOT NULL,
  created_at  TIMESTAMP       NOT NULL      DEFAULT now(),
  updated_at  TIMESTAMP       NOT NULL      DEFAULT now(),
  PRIMARY KEY (id),
  FOREIGN KEY (user_id)
    REFERENCES users(id)
    ON DELETE CASCADE
)
```

1.1.1 Glossar zu SQL-Create

- **AUTO_INCREMENT** bedeutet, dass bei jedem neu eingefügten Datensatz dieser Wert automatisch um 1 hochgezählt wird. Darum braucht man sich also nicht zu kümmern.
- **DEFAULT** regelt, welchen Wert ein Datensatz bekommt, wenn beim Anlegen des Datensatzes kein Wert angegeben wird. (engl. *default*: Standard, Vorgabe)
- **FOREIGN KEY**: Diese Spalte ist ein Verweis auf eine andere Tabelle. Jeder Wert in dieser Spalte befindet sich in einer Spalte der Zieltabelle. Diese Spalte heißt in der aktuellen Tabelle **Fremdschlüssel**. Die Tabelle, auf die verwiesen wird, heißt **Zieltabelle**. Bei InstaHub ist die Zieltabelle die Tabelle `users` und die Spalte, in der dort gesucht werden soll ist die Spalte `id`. Wie fast immer ist der referenzierte Fremdschlüssel in der Zieltabelle der dortige Primärschlüssel.

lüssel (s. PRIMARY KEY).

Jeder Wert eines Fremdschlüssels muss zwingend in der Zieltabelle enthalten sein. Das kontrolliert die Datenbank. Und damit sie das kann, deklariert man Fremdschlüssel als FOREIGN KEY. In der Tabelle Städte verweisen die Werte in der Spalte Land_ID auf die die Tabelle Länder und zwar auf die Spalte Land_ID. In der Tabelle Länder ist Land_ID der **Primärschlüssel**. In der Tabelle Städte ist die Spalte Land_ID ein **Fremdschlüssel**, der auf eine Zeile in einer anderen Tabelle verweist.

Das Prinzip sehen Sie im folgenden Schaubild:

Stadt_ID	Name	Land_ID	Einwohner
1	Elmshorn	1	49618
2	New York	2	853673
3	Leipzig	1	581980

Land_ID	Name	Einwohner
1	Deutschland	83000000
2	USA	327774344

Fremdschlüssel verweisen auf eine andere Tabelle

- INT ist ein Datentyp. Jede Spalte kann nur Werte einer bestimmten Art enthalten. Es geht also nicht, dass in einer Spalte mal eine Zahl, mal ein Text und dann wieder ein Datum steht. INT (4) bedeutet, dass in der Spalte ganze Zahlen (engl. *integer*) von -2.147.483.648 bis +2.147.483.647 stehen können. Da sie hier aber als UNSIGNED deklariert wurden, sind sie immer positiv und können von 0 bis 4.294.967.295 reichen. Die Zahl in Klammern ist von untergeordneter Bedeutung. Sie gibt an, wie viele Stellen zur Ausgabe der Zahl bereit gestellt werden sollen.
- NULL bedeutet nicht "0", sondern "nichts, leer". NULL ist nicht gleich etwas anderem, auch nicht dem Wert 0. Wenn eine Tabellenspalte in jeder Zeile einen Wert haben soll, kann man NOT NULL notieren, dann wird die Datenbank dort unter keinen Umständen erlauben, nichts einzutragen. Notiert man NULL, erlaubt man auch leere Tabellenzellen.
- ON DELETE: Dies ist Teil der Deklaration eines Fremdschlüssels, siehe FOREIGN KEY. Diese Regel beantwortet die Frage, was passiert, wenn der Zieldatensatz gelöscht wird. ON DELETE CASCADE bedeu-

tet, dass beim Löschen eines Datensatzes in der Tabelle `USERS` alle zugehörigen Datensätze in der Tabelle `photos` gelöscht werden. Würde man dies nicht vereinbaren, müssten zunächst alle Fotos des*er Benutzer*in gelöscht werden, damit der Datensatz in der Tabelle `users` gelöscht werden kann.

- **PRIMARY KEY:** Vereinfacht gesehen hat jede Tabelle eine Spalte, in der kein Wert doppelt vorkommt. Diese Spalte nennt man **Primärschlüssel**. Wenn man hier nach einem Wert sucht findet man also immer keine oder genau eine Zeile, niemals aber mehrere. Mit der Deklaration **PRIMARY KEY** geben Sie an, welche Spalte der Primärschlüssel ist.
- **TIMESTAMP** ist ein Datum mit Uhrzeit. Das frühest mögliche Datum ist 1970, das spätest mögliche 2038.
- **VARCHAR** ist ebenfalls ein Datentyp (s. **INT**). Bei **VARCHAR** handelt es sich um eine Zeichenkette variabler Länge. Die maximale Länge wird in Klammern angegeben.

1.1.2 Weitere SQL-Datentypen

SQL verfügt über zahlreiche Datentypen. Leider wird nicht jeder Datentyp von jedem SQL-Server unterstützt oder gleich interpretiert. Die Datentypen in der folgenden Auswahl wird zwar in diesem Kapitel nicht verwendet, ist aber trotzdem wichtig:

Datentyp	Bedeutung
<code>DECIMAL(p, s)</code>	Eine Zahl mit festem Nachkommateil. <i>s</i> gibt die Zahl der Nachkommastellen an, <i>p</i> die Zahl der Stellen insgesamt. Der Datentyp ist sehr geeignet für Geldbeträge.
<code>CHARACTER LARGE OBJECT</code>	Wie VARCHAR , aber es gibt keinerlei obere Grenze, so dass ganze Textdokumente gespeichert werden können. Umständliche Handhabung, daher wird VARCHAR bevorzugt.
<code>FLOAT</code>	Zahl mit gleitend vielen (engl. <i>to float</i>) Nachkommastellen, wie beim Taschenrechner. Achtung: Es kommen Rundungsungenauigkeiten vor. Der Datentyp ist geeignet für technische Zahlen und nicht für Geldbeträge.
<code>SMALL INT</code>	Kleiner Ganzzahlwert (meist -32.768 bis +32.767)
<code>CHAR(n)</code>	Zeichenkette der festen Länge <i>n</i> . Die Nutzung ist weniger aufwändig als VARCHAR .

Aufgabe 9.2 CREATE TABLE photos

1. *Erzeugen* Sie die Tabelle `photos` so wie zu Anfang dieses Kapitls angegeben. Achten Sie besonders auf die Datentypen und die Namen der Spalten.
2. *Füllen* Sie die Tabelle mit Hilfe der von Ihrer Lehrkraft bereitgestellten Datei mit Daten!
3. *Schauen* Sie sich in Ihrem InstaHub um!
4. *Überlegen* Sie sich, wie die Fotos in den Hub gekommen sind. Sie haben ja schließlich nur eine Textdatei hochgeladen...

1.2 Das Entity-Relationship-Modell

Wir haben nun drei Tabellen in unserer Datenbank. Um den Überblick zu behalten, wäre es schön, die Tabellenstrukturen grafisch darzustellen. Hierzu gibt es das Entity-Relationship-Modell, kurz **ER-Modell** oder ERM. Ein ER-Modell besteht aus den folgenden vier Elementen:

1. **Gegenstände (Entities)**: Eine *Entität* (engl. *entity*) stellt eine Klasse von *Objekten* der *realen* Welt im ER-Modell ab. Z. B. Produkt, Kunde, Schüler oder Informatikkurs. (Sie sehen, dass Objekte nicht unbedingt gegenständlich sein müssen.) Entities werden in ER-Modellen als Rechtecke dargestellt:



Schüler

Informatikkurs

Entitäten werden im ERM als Rechtecke dargestellt

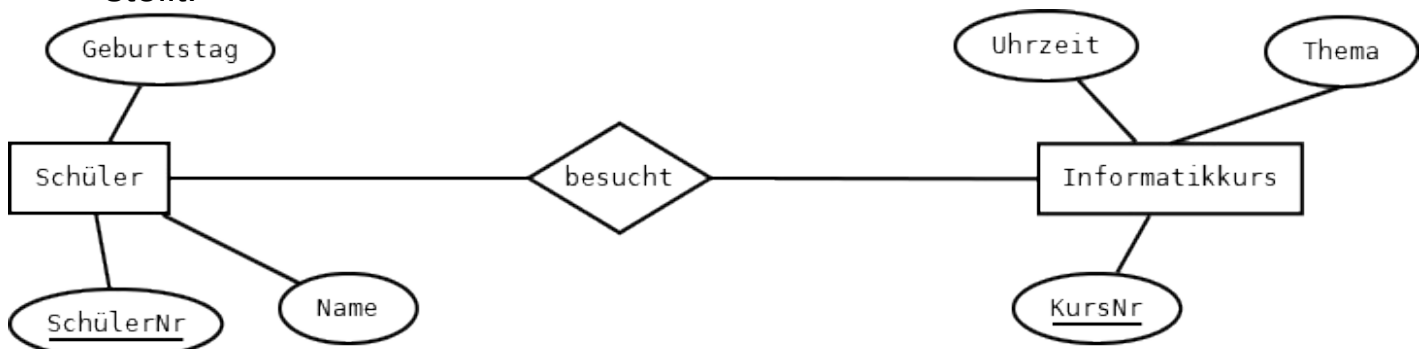
2. **Attribute**: Entitäten und Beziehungen haben *Eigenschaften*, die durch *Attribute* beschrieben werden. Diese Attribute findet man in der Datenbank in den Tabellenspalten. Typische Eigenschaften für die Entität Produkt könnten z. B. Bezeichnung, Gewicht, Preis und Artikelnummer sein.

Attribute werden in ER-Modellen als Ovale an die Entitäten angehängt. Die Primärschlüssel heißen im ER-Modell eigentlich *identifizierende Attribute* und werden unterstrichen:



Attribute werden als Ovale angehängt

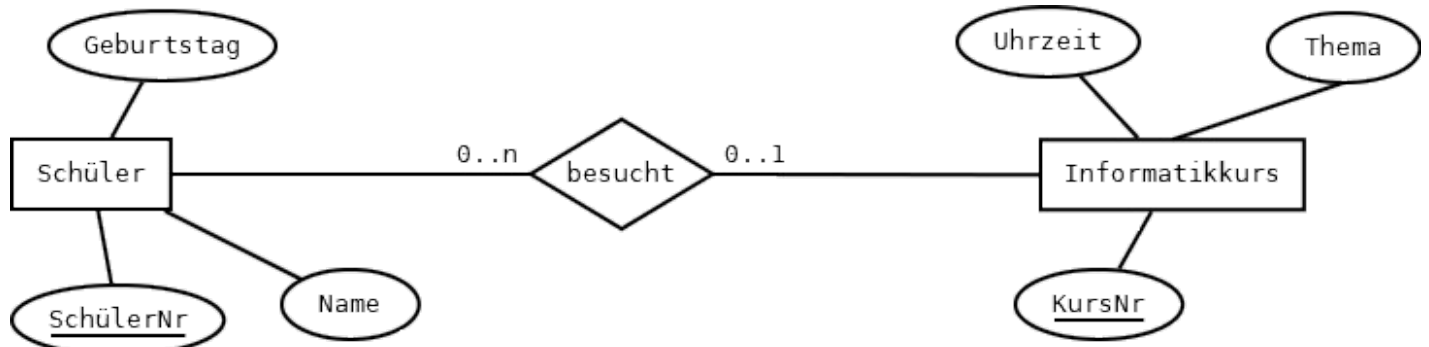
3. **Beziehungen (Relationships):** Ein *Relationship* stellt eine Beziehung zwischen den Objekten der realen Welt dar. Z. B. die Beziehungen *kaufte* zwischen Kunde und Produkt *besucht* zwischen den Entitäten Schüler und Informatikkurs. Relationships verbinden *immer* genau zwei Entitäten. Relationships werden als Raute dargestellt:



Relationships werden durch Rauten dargestellt

4. **Kardinalität:** Beziehungen verbinden nicht immer genau ein Exemplar einer Entität mit genau einem Exemplar einer anderen Entität. So kann ein Kunde mehrere Produkte gekauft haben. Kardinalitäten werden leider sehr verschieden notiert. Wir verwenden hier die in der folgenden Abbildung verwendete Notation: Ein bestimmte*r Schüler*In hat genau einen (1 . . 1) Informatikkurs und jeder Informatikkurs wird von 0 bis beliebig vielen Schüler*Innen (0 . . n) besucht.)

1 Tabellen erstellen und das Entity-Relationship-Modell



Kardinalitäten geben an, wie viele Exemplare an dem Relationship beteiligt sind

Bitte beachten Sie: Die **Fremdschlüssel** werden im ER-Modell *nicht* notiert, nur die **Primärschlüssel**. Die Fremdschlüssel werden erst beim Umsetzen der ER-Modells in Tabellen ergänzt. Im obigen Beispiel müsste in der Tabelle Informatikkurs der Primärschlüssel der Entität Schüler, SchülerNr eingefügt werden. Wir werden uns in Abschnitt 10.3 und in Kapitel 17 näher mit diesem Thema beschäftigen.

Aufgabe 9.3 Das ER-Modell von InstaHub (1)

Erstellen Sie ein ER-Modell Ihrer InstaHub-Datenbank! Bei den Attributen wählen Sie einige exemplarische aus.