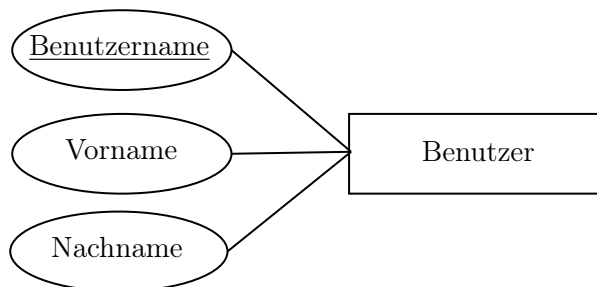


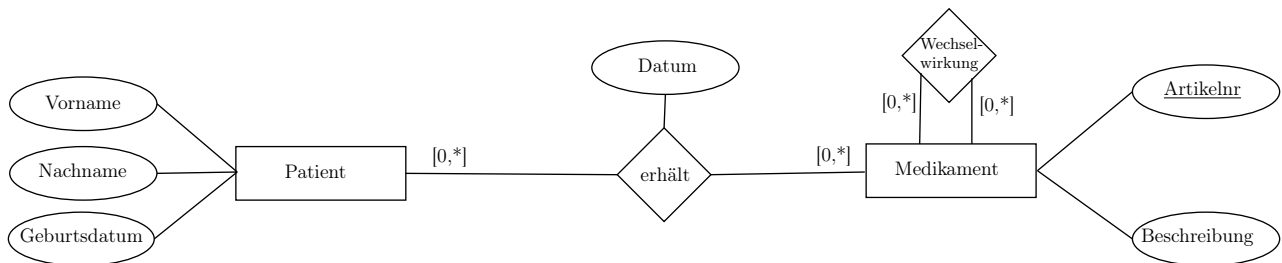
## Aufgabe 1 *ER- und Relationenmodell*

1.1 Ergänzen Sie den folgenden Ausschnitt des ER-Modells eines Online-Shops, sodass die folgenden Bedingungen erfüllt werden:

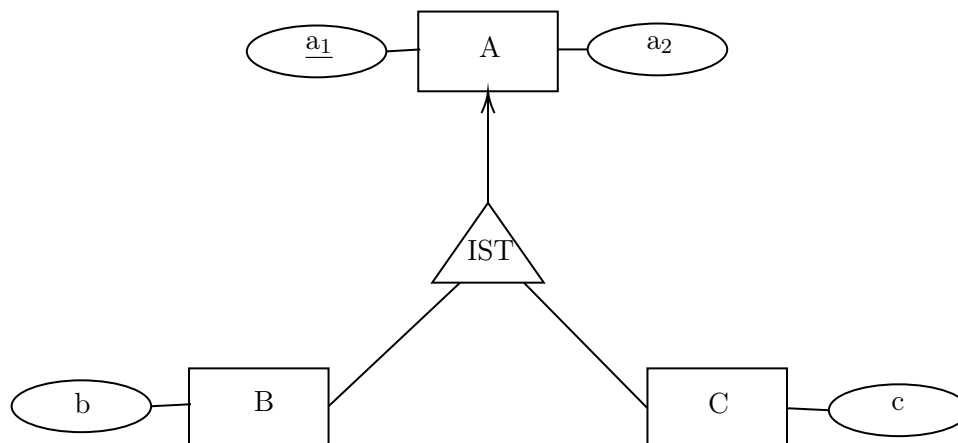
- Es gibt Benutzer. Diese werden mit einem eindeutigen Benutzernamen und einem Vornamen und Nachnamen abgespeichert.
- Außerdem gibt es Kategorien von Produkten. Jede Kategorie hat einen eindeutigen Namen.
- Produkte werden mit einer eindeutigen Artikelnummer, sowie einem Namen, einer Beschreibung und einem Preis abgespeichert.
- Ein Benutzer kann beliebig viele Kategorien favorisieren, und jede Kategorie kann von mehreren Benutzern favorisiert werden.
- Jedes Produkt gehört mindestens einer Kategorie, und jede Kategorie enthält mindestens ein Produkt.
- Zudem hat ein Produkt beliebig viele Exemplare, welche mit einer eindeutigen Seriennummer versehen sind. Jedes Exemplar gehört zu genau einem Produkt.
- Ein Produkt kann beliebig vielen anderen Produkten ähneln.
- Ein Benutzer kann beliebig viele Exemplare kaufen, doch jedes Exemplar kann von maximal einem Benutzer gekauft werden.



**1.2** Überführen Sie den folgenden Ausschnitt eines ER-Modells in das Relationenmodell. Ihr Relationenmodell soll minimal sein und Nullwerte vermeiden. Bei Verschmelzungen reicht es, wenn nur die betroffenen Relationen/Entitäten vor und nach der Verschmelzung dargestellt werden. Primärschlüssel werden sowohl im ER- als auch im Relationenmodell unterstrichen dargestellt. Fremdschlüssel sollen überstrichen dargestellt werden.



**1.3** Im Übungsbetrieb haben Sie vier Arten der Darstellung einer IST-Beziehung im Relationenmodell kennengelernt. Betrachten Sie das folgende Beispiel einer Spezialisierung. Bilden Sie das Relationenmodell für alle vier Möglichkeiten, die in der Übung besprochen wurden. Primärschlüssel sollen unterstrichen und Fremdschlüssel überstrichen dargestellt werden. Achten Sie darauf eigene Attribute aussagekräftig zu benennen.



## Aufgabe 2      *Relationenalgebra, Tupelkalkül, SQL*

Gegeben sei das folgende Relationenmodell:

TANKSTELLE( <u>ID</u> , Standort, <u>Unternehmen</u> )
UNTERNEHMEN( <u>Name</u> , Hauptsitz, Jahresumsatz)
KRAFTSTOFF(Bezeichnung, Aggregatzustand, Einheit)
ERZEUGER( <u>ID</u> , <u>Effizienzklasse</u> , Land)
BIETET_AN( <u>Tankstelle</u> , <u>Kraftstoff</u> , Zeitpunkt, Preis)
PRODUZIERT( <u>Erzeuger</u> , <u>Kraftstoff</u> , Jahr, Menge)
BEZIEHT( <u>Unternehmen</u> , <u>Erzeuger</u> , <u>Kraftstoff</u> , Jahr, Menge)

<i>Fremdschlüsselbedingungen:</i>
<i>Unternehmen</i> ist Fremdschlüssel bzgl. UNTERNEHMEN.Name
<i>Tankstelle</i> ist Fremdschlüssel bzgl. TANKSTELLE.ID
<i>Kraftstoff</i> ist Fremdschlüssel bzgl. KRAFTSTOFF.Bezeichnung
<i>Erzeuger</i> ist Fremdschlüssel bzgl. ERZEUGER.ID

### 2.1 Formulieren Sie die folgenden Anfragen in der **Relationenalgebra**:

- (a) Geben Sie die Namen aller Unternehmen aus, die ihren Hauptsitz in *London* haben.  
*Hinweis: Standort und Hauptsitz speichern hier einfachheitshalber nur die Stadt.*
- (b) Geben Sie die Tankstellen (ID) aus, die Kraftstoffe mit dem Aggregatzustand *flüssig* anbieten.  
Dabei darf keine Tankstelle doppelt gelistet werden.
- (c) Geben Sie die Unternehmen (Name) aus, die Kraftstoffe ausschließlich von Erzeugern der Effizienzklasse A bekommen haben.  
*Hinweis: Standort und Hauptsitz speichern hier einfachheitshalber nur die Stadt.*
- (d) Geben Sie alle Erzeuger aus, die im Jahr *2003* sowohl *Erdgas* als auch *Superbenzin* hergestellt haben.

### 2.2 Formulieren Sie die folgenden Anfragen im **Tupelkalkül**:

- (a) Geben Sie alle Tankstellen (ID) aus, die Superbenzin für mindestens *1.3*, aber maximal *1.5* Euro angeboten haben (zu beliebigen Zeitpunkten).
- (b) Geben Sie alle Unternehmen (alle Attribute) aus, die bei keinem Erzeuger Kraftstoff eingekauft haben, der auch *Erdgas* produziert. Betrachten Sie dabei alle Jahre.

**2.3** Formulieren Sie die folgenden Anfragen in **SQL**:

- (a) Geben Sie das Jahr aus, in dem erstmalig *Ökostrom* produziert wurde.
- (b) Geben Sie die Unternehmen (Name, Jahresumsatz, durchschnittliche Bezugsmenge) absteigend sortiert nach ihrer durchschnittlichen Bezugsmenge im Jahr *2019* von Kraftstoffen aus, die in *Litern* gemessen werden.
- (c) Die neuen Jahresumsätze liegen vor. Das Unternehmen *Spriti* hat nun einen Jahresumsatz von *205 000 000 000* Euro. Pflegen Sie diese neue Information in die Datenbank ein.
- (d) Der Preis für *Superbenzin* soll sich bei der Tankstelle mit der ID *6932* auf *1.8* erhöhen. Der Zeitpunkt soll in etwa die aktuelle Zeit sein.  
*Hinweis: Der Zeitpunkt ist als SQL Datetime in der Form YYYY-MM-DD HH:MI:SS gespeichert. Beispiel: 2019-12-24 13:37:00. BIETET\_AN enthält die Preishistorie und der zuletzt eingefügte (Zeipunkt) Preis ist der aktuelle.*

### Aufgabe 3      *Normalformen*

**3.1** Gegeben sei die Relation  $R_1(A, B, C, D, E, F, G, H, I)$  mit der folgenden Menge  $S$  von funktionalen Abhängigkeiten:

$$S = \{ \begin{array}{lll} AB & \rightarrow & EF, \\ C & \rightarrow & I, \\ BI & \rightarrow & DF, \\ G & \rightarrow & H, \\ H & \rightarrow & B \end{array} \}$$

- (a) Bestimmen Sie die Hüllen der Attributmengen  $\{B, C\}$  und  $\{A, I\}$  bezüglich  $S$ .
- (b) Bestimmen Sie für die Relation  $R_1$  alle Schlüsselkandidaten und begründen Sie, warum es keine weiteren Schlüsselkandidaten geben kann.
- (c) Geben Sie für die 1., 2. und 3. Normalform, sowie die BCNF an, ob sich die Relation  $R_1$  in der jeweiligen Normalform befindet. Begründen Sie Ihre Antworten! Gehen Sie davon aus, dass die Attribute atomare Wertebereiche haben.
- (d) Kann man die funktionale Abhängigkeit  $BF \rightarrow I$  aus der vorgegebenen Menge  $S$  von funktionalen Abhängigkeiten ableiten? Begründen Sie Ihre Antwort.

**3.2** Gegeben sei die Relation  $R_2(A, B, C, D, E, F, G, H, I, J)$  mit der folgenden Menge  $S$  von funktionalen Abhängigkeiten:

$$S = \{ \begin{array}{lll} A & \rightarrow & BF, \\ ABC & \rightarrow & EG, \\ EBI & \rightarrow & F, \\ BG & \rightarrow & AH, \\ BH & \rightarrow & I, \\ F & \rightarrow & E, \\ B & \rightarrow & FGI \end{array} \}$$

Überführen Sie die Relation  $R_2$  mittels des Syntheseverfahrens in die 3. Normalform. Geben Sie dabei den vollständigen Lösungsweg an!

## Aufgabe 4 *SQL-Injection*

Sie sind der Hacker *Kuno Knack* und haben ein Konto bei der *Entenbank*. Das *Git Repository* der Bank wurde falsch konfiguriert. Deshalb ist der Python-Code des Webportals jetzt öffentlich verfügbar. Beim Lesen des Codes fällt Ihnen auf, dass für die Formatierung der Eingabe bei SQL-Anfragen sogenannte *f-strings* verwendet wurden. Hierbei werden Ausdrücke zwischen Mengenklammern in Strings evaluiert und durch den Wert ersetzt. Zum Beispiel wird der String `'1 + 2 = {1 + 2}'` zu `'1 + 2 = 3'`. Sie vermuten, dass eine Tabelle *KUNDEN*(*KontoNr*, *Name*, *Kontostand*, *Passwort*, *Spenden*) in der Datenbank existiert. Eine **beispielhafte** Ausprägung könnte etwa so aussehen:

KontoNr	Name	Kontostand	Passwort	Spenden
1	Dagobert Duck	999999999	Gold	0
2	Donald Duck	-10	Daisy	2
3	Kuno Knack	0	1234	0

**4.1** Die *Entenbank* erlaubt es ihren Kunden, ein neues Passwort zu setzen. Die Benutzereingabe für die Variable **neues\_passwort** wird ungeprüft aus dem Web-Interface übernommen. Die Variable **kontonr** wird anderswo im Code gesetzt und kann von Ihnen nicht modifiziert werden.

Geben Sie ein Passwort an, mit dem Ihr **Kontostand** auf den Wert **999999** gesetzt wird.

- Beachten Sie dabei, dass außer Ihrem Kontostand und Passwort keine anderen Daten geändert werden dürfen, denn sonst würde Ihr Betrug schnell auffallen.
- Beachten Sie weiterhin, dass die SQL-Abfrage mehrzeilig ist, aber dass die Zeichen -- nur einen Zeilenkommentar einleiten.

```
UPDATE
    KUNDEN
SET
    Passwort = '{neues_passwort}'
WHERE
    KontoNr = '{kontonr}'
```

**4.2** Um Fehler aufzudecken überprüft die *Entenbank* jeden Abend, ob sich die gesamte Geldmenge aller Konten, abzüglich der Änderungen durch eingehende und ausgehende Überweisungen, nicht verändert hat. Um Ihre Spuren zu verwischen müssen Sie also einen anderen Kontostand entsprechend verringern, damit Ihr Diebstahl ausgeglichen wird. Es ist bekannt, dass der Kunde *Dagobert Duck* einen ausreichen hohen Kontostand hat.

Die Bank bietet ein Formular an, um Geld an wohltätige Organisationen zu spenden. geben Sie eine Eingabe für die Variable **spende** in der nachfolgenden SQL-Anfrage an, sodass der Kontostand vom Kunden mit **Namen** *Dagobert Duck* um den Wert **999999** reduziert wird.

```
UPDATE KUNDEN SET Kontostand =
Kontostand - {spende}, Spenden = Spenden + 1 WHERE KontoNr = {kontonr}
```

## Aufgabe 5      *Transaktionen*

**5.1** Überprüfen Sie, ob die folgenden Schedules konfliktserialisierbar sind. Zeichnen Sie hierzu den Konfliktgraphen jedes Schedules und geben Sie zu jeder Kante des Konfliktgraphens ein Konfliktpaar an.

Wenn der Schedule serialisierbar ist, dann geben sie alle äquivalenten seriellen Schedules an.

- (a)  $S_1 := w_3(a)r_1(b)r_2(c)r_1(a)r_3(b)r_3(c)w_1(b)r_2(b)w_2(c)$
- (b)  $S_2 := r_1(b)r_2(a)w_3(c)r_1(c)r_2(b)r_3(c)w_1(b)w_2(a)r_3(a)w_1(a)w_3(b)$
- (c)  $S_3 := r_2(c)w_2(c)r_2(b)r_1(c)r_1(b)w_1(b)w_1(a)r_3(a)r_3(b)w_3(a)$

**5.2** Können unter dem konservativen Zwei-Phasen-Sperrprotokoll Verklemmungen (Deadlocks) auftreten? Begründen Sie Ihre Antwort.

**5.3** Ergänzen Sie bei den folgenden Schedules Read-Lock-, Write-Lock- und Unlock-Anweisungen nach dem (normalen) Zwei-Phasen-Sperrprotokoll. Achten Sie darauf, Sperren so früh wie möglich abzugeben.

- (a)  $T_1 := r_1(a)r_1(c)w_1(a)r_1(b)w_1(c)$
- (b)  $T_2 := r_2(b)w_2(b)r_2(a)r_2(b)$