

10.10.2019

Übungsblatt 1 – Lösungsvorschlag

Diskussionsgrundlage Proseminar

In diesem Semester werden wir die Proseminar-Stunden intensiv für das gemeinsame Diskutieren und Lösen von Beispielen verwenden. Daher enthält jedes Übungsblatt zunächst einen Abschnitt mit Beispielen und Aufgaben, dieser sollte als Diskussionsgrundlage für das Proseminar dienen. Die Beispiele werden wir während des Proseminars lösen, sie dienen als Vorbereitung für die zu lösenden Aufgaben und Klausuren.

Die Proseminar-Stunden sollen für Sie ein Forum darstellen. Lösen Sie die im Proseminar gestellten Aufgaben in Gruppen und diskutieren Sie die verschiedenen Möglichkeiten. Gestalten Sie mit, fragen Sie Ihre Proseminar-Leiterin/Ihren Proseminar-Leiter, Ihre Kolleginnen und Kollegen und nützen Sie die Zeit, um sich intensiv über die Inhalte auszutauschen.

Wichtig: Stellen Sie Fragen (auch wenn sie ihnen noch so “blöd” erscheinen), nur so können wir die noch offenen Themen und Fragen gemeinsam beantworten. Für diese Aufgaben und Diskussionen werden keinerlei Noten vergeben.

Für eine optimale Lernumgebung bitten wir Sie, pro Gruppe mindestens ein Notebook zur Übung zu bringen, um beispielsweise SQL-Abfragen direkt ausprobieren zu können.

Tools

Im Folgenden beschreiben wir kurz jene Tools, die Sie für die Bearbeitung der Aufgaben benötigen werden.

Diagramme

In diesem Proseminar werden wir häufig ER-Diagramme zeichnen. Als Editor für Diagramme können wir zum Beispiel yEd¹ empfehlen. Sollten Sie sich für diesen Editor entscheiden, so können Sie auf das Template `ER-master.graphmlOLAT` zurückgreifen, um schneller einzelne ER-Elemente zu zeichnen. Selbstverständlich können Sie auch gerne andere Editoren wie Draw.io² verwenden oder Zeichnungen von Hand anfertigen (diese können Sie für die Abgabe von Lösungen fotografieren).

Datenbankmanagementsystem

Als Datenbankmanagementsystem (DBMS) verwenden wir PostgreSQL 11³. Alle Aufgaben auf diesem System erstellt und getestet werden. Von einer Verwendung von MySQL/MariaDB⁴ raten wir für dieses Proseminar ab, da sich die Funktionalität und der SQL Syntax von PostgreSQL und MySQL/MariaDB leicht unterscheidet. Es steht Ihnen natürlich frei, für die Ausarbeitung Ihrer Aufgaben ein beliebiges DBMS zu verwenden, solange Ihre Abgaben auf PostgreSQL 11 ausführbar sind.

¹<https://yworks.com/products/yed>

²<https://about.draw.io>

³<https://postgresql.org/docs/11/index.html>

⁴<https://mariadb.org>

SQL Client

Als SQL Client können wir Ihnen pgAdmin⁵ empfehlen. Natürlich steht es Ihnen auch hier frei, ein anderes Tool wie zum Beispiel DBeaver⁶ zu wählen.

Docker (Optional)

Um das DBMS und den SQL Client (pgAdmin 4) möglichst einfach, abgekapselt und unkompliziert auf Ihrem Computer einzurichten, haben wir ein auf Docker⁷ basiertes Setup vorbereitet. Installieren Sie dazu die Docker Engine⁸ und Docker Compose⁹. Kopieren Sie im Anschluss das vorbereitete `docker-compose.OLATyaml` in einen Ordner Ihrer Wahl und führen Sie es mit `docker-compose` aus. Da es sich dabei um ein Command Line Tool handelt, muss die Ausführung des Befehls im Terminal erfolgen. Wechseln Sie dazu im Terminal in den Ordner mit der heruntergeladenen Datei und führen Sie den Befehl `docker-compose up` aus. Dieser Befehl führt die Instruktionen im `docker-compose.OLATyaml` aus. Dazu werden zuerst die Docker Images `postgres:11.5` und `dpage\pdadmin4` heruntergeladen (sofern diese nicht schon auf Ihrem System vorhanden sind). Anschließend werden die zwei Services `db` und `pgadmin` konfiguriert, gestartet und ihre Ports auf ihr lokales System gemapped. Alternativ können Sie `docker-compose up` auch mit dem zusätzlichen Parameter `-d` im detached Modus ausführen. Um die Container anschließend zu beenden, müssen Sie den Befehl `docker-compose down` ausführen.

Hinweis



Für die meisten Linux Distributionen sollte es ausreichend sein, die Pakete aus den System Repositories zu installieren. Sollten diese nicht vorhanden sein oder es andere Probleme geben (z.B. zu alte Versionen), dann können Sie auch die für ihre Distribution passende Anleitung von [docker docs^a](#) verwenden.

Für Windows sollten Sie ebenfalls die entsprechende Anleitung auf [docker docs^a](#) verwenden. Zusätzlich könnte Ihnen auch das Docker Beginner Tutorial 4–6 von Raghav Pal^b oder eine Google Suche helfen.

^a<https://docs.docker.com/install/#supported-platforms>

^b<https://www.youtube.com/playlist?list=PLhW3qG5bs-L99pQsZ74f-LC-t0EsBp2rK>

Sobald die Services erfolgreich gestartet wurden, können Sie die pgAdmin4 Weboberfläche unter <http://localhost:8080> erreichen. Als E-Mail Adresse wurde `postgres` vergeben und das Passwort lautet `SuperSecret`. Dort können Sie eine Datenbankverbindung einrichten. Verwenden Sie dafür den Server-Dialog wie in der pdAdmin Dokumentation¹⁰ beschrieben. In der Weboberfläche ist die PostgreSQL Datenbank unter dem Hostname `db` erreichbar. Der Benutzer ist `postgres` und das Passwort lautet `SuperSecret`. Zusätzlich ist der Port des PostgreSQL Service (`db`) auch auf ihren `localhost` gemapped, Sie können die Datenbank also auch von ihrem System aus (z.B. mit DBeaver) erreichen, indem Sie als Hostname `localhost` verwenden der Benutzer und das Passwort bleiben dabei gleich.

⁵<https://pgadmin.org>

⁶<https://dbeaver.io>

⁷<https://www.docker.com>

⁸<https://docs.docker.com/engine>

⁹<https://docs.docker.com/compose>

¹⁰<https://www.pgadmin.org/docs/pgadmin4/latest/connecting.html>

Hinweis



Sollten Sie Docker Toolbox verwenden, wird Docker in einer virtuellen Maschine (VirtualBox) ausgeführt. Das bedeutet, dass Sie anstelle von localhost die entsprechende IP Adresse verwenden müssen. Führen Sie dafür den Befehl `docker-machine ip default` aus, um die IP Adresse auszulesen.

Sollte die IP beispielsweise 192.168.99.100 sein, können Sie mit <http://192.168.99.100:8080> auf die pgAdmin4 Oberfläche zugreifen.

Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

- a) ☐ ★ Überlegen und Diskutieren Sie (kurz) ob es Ihnen möglich ist die weiter oben am Zettel erwähnten Tools zu Installieren um diese zu Verwenden. Die Einrichtung dieser Tools sollte unter Linux, MacOS und Windows möglich sein, das Setup wurde allerdings nur unter Linux getestet.

Das DBMS, der SQL Client und Docker stehen auf den ZID Rechnern nicht zur Verfügung daher sollten Sie diese Tools auf ihrem eigenen Computer installieren. Sollte Ihnen das aus irgendeinem Grund nicht möglich sein (z.B. Sie besitzen keinen eigenen Computer) dann klären Sie bitte das weitere Vorgehen mit ihrer Kursleiterin / ihrem Kursleiter ab.

- b) ☐ ★ Sie sollten mehrere Dateien erstellen und diese dann auf OLAT abgeben.
- Erstellen Sie eine Textdatei mit dem Namen `textdatei.txt` und schreiben Sie etwas beliebiges hinein.
 - Erstellen Sie eine weitere Textdatei `query.sql` die die folgende SQL-Query enthält (achtung die Zahlen 1 und 2 sind Zeilennummern und gehören nicht zum Inhalt der Datei):

```
1  select *  
2  from pg_database;
```
 - Erstellen Sie ein PDF-Dokument `text.pdf` z.B. mit Microsoft Word, LibreOffice Writer, LaTeX oder Google Docs welches einen beliebigen Text enthält.
 - Erstellen Sie eine PDF-Datei `diagramm.pdf` oder ein Foto `diagramm.jpg` das ein simples ER-Diagramm enthält.

Geben Sie nun die zuvor erstellten Dateien in OLAT¹¹ für das Blatt 0 ab.

Sollten dabei Probleme auftreten, dann lassen Sie sich bitte von Ihren Kolleginnen und Kollegen oder der Kursleiterin und -leiter helfen.

Diese Unteraufgabe sollte jeder selbst ausführen können, um sicherzustellen, dass jeder in OLAT Abgaben machen kann. Dies sollte einem Punkteverlust vorbeugen, da verspätete Abgaben aus dem Hausaufgabenteil nicht akzeptiert werden können werden.

¹¹<https://lms.uibk.ac.at/auth/RepositoryEntry/4559733019/CourseNode/100429179039345>

Hinweis



Geben Sie nur **UTF-8** encodierte Textdateien ab. Verwenden Sie daher einen modernen Texteditor zum Erstellen dieser Dateien. Geeignet/empfohlene Editoren sind VIM^a, Emacs^b, Sublime^c, Atom^d, gedit^e oder Notepad++^f.

^a<http://www.vim.org>

^b<https://www.gnu.org/software/emacs/>

^c<https://www.sublimetext.com/>

^d<https://atom.io/>

^e<https://wiki.gnome.org/Apps/Gedit>

^f<https://notepad-plus-plus.org/>

c) ☐ ★ Diskutieren Sie bitte die folgenden Punkte:

- Gegeben ist eine textuelle Beschreibung des zu speichernden Szenarios. Was ist eine sinnvolle Schritt-für-Schritt Vorgehensweise für den Entwurf eines ER-Modells?
- Worin liegt der Unterschied zwischen einer Entität und einer Beziehung? Kann eine Beziehung (Relationship) auch eine Entität sein?
- Was sind sinnvolle Schlüssel für Personen? (Name, Geburtsdatum, Adresse, Sozialversicherungsnummer, Matrikelnummer, Kombination daraus oder auch ein Surrogatschlüssel?)

Lösung

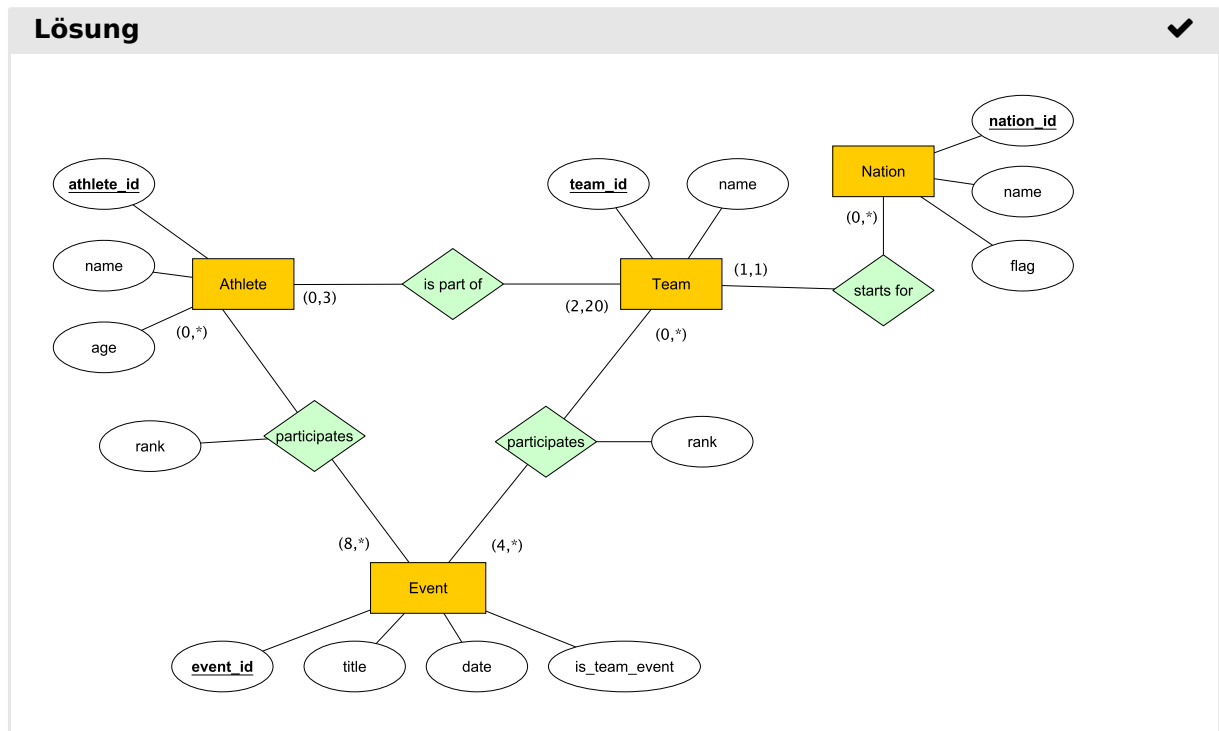


- Herangehensweise: (i) Entitäten feststellen, (ii) Extrahieren der Attribute, (iii) Schlüssel festlegen, (iv) Relationships herausarbeiten, (v) ER-Modell überprüfen und iterativ verbessern.
- Entität ist ein Objekt (unterscheidbar von anderen Objekten), eine Beziehung ist zwischen Entitäten definiert. Eine Beziehung kann aber auch eine Entität bilden (vgl. n:m Beziehungen).
- Schlüssel müssen stets eindeutig sein (vgl. Folie 6 in Foliensatz 2: "Schlüsseleigenschaft ist ein semantisches Kriterium, sie kann nicht vom momentanen Datenbestand abgeleitet werden"). Daher ist eine Kombination aus Name und Geburtsdatum oder Adresse nicht ausreichend für eine eindeutige Identifikation (diese Bedingung muss auch in Zukunft stets gelten). Daher ist beispielsweise eine Sozialversicherungsnummer oder Matrikelnummer (für Studierende) ein denkbarer Schlüssel. Die Sozialversicherungsnummer darf (in Österreich), bis auf einzelne Ausnahmen, aus rechtlichen Gründen nicht verwendet werden und die Matrikelnummer kann nur verwendet werden, wenn ausschließlich Studenten aus Österreich gespeichert werden. Daher ist ein Surrogatschlüssel (ein künstlicher (synthetischer) Schlüssel) die vermutlich beste Wahl, da dieser immer eindeutig ist.

d) ☐ ★ Verwenden Sie die *min-max-Notation*, um folgenden Sachverhalt über Sportveranstaltungen zu modellieren. Verwenden Sie englische Bezeichner.

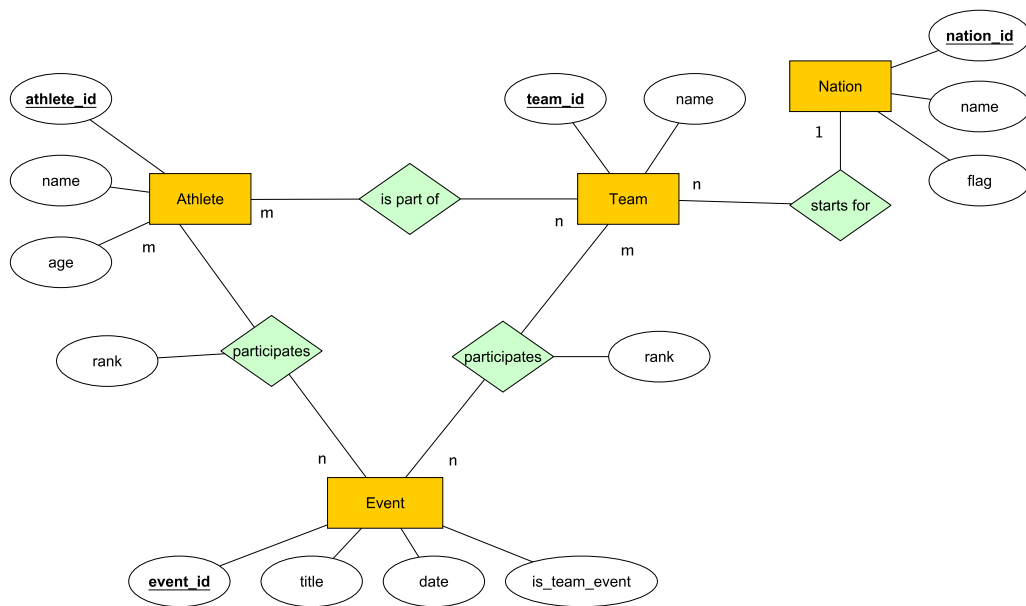
- Ein Athlet hat einen Namen und ein Alter.
- Ein Athlet kann bis zu drei Teams angehören muss dies aber nicht.
- Ein Team hat einen Namen und besteht aus mindestens 2 und maximal 20 Athleten. Jedes Team startet für genau eine Nation, für welche der Name und die Flagge gespeichert werden soll.

- Ein sportliches Ereignis hat einen Titel und ein Datum, und es soll zusätzlich festgehalten werden, ob es sich um ein Einzel- oder Teamereignis handelt. Es gibt keine Ereignisse, an denen sowohl einzelne Athleten als auch Teams teilnehmen.
- Sowohl Athleten als auch Teams nehmen an (unterschiedlichen) Sportereignissen teil. Dabei gilt, dass bei einem Einzelereignis mindestens 8 Athleten teilnehmen müssen und bei einem Teamereignis mindestens 4.
- Wenn Athleten oder Teams bei einem Ereignis teilnehmen, dann wird das Ergebnis (d.h. der Rang, z.B. 1., 2., 3., ...) festgehalten.



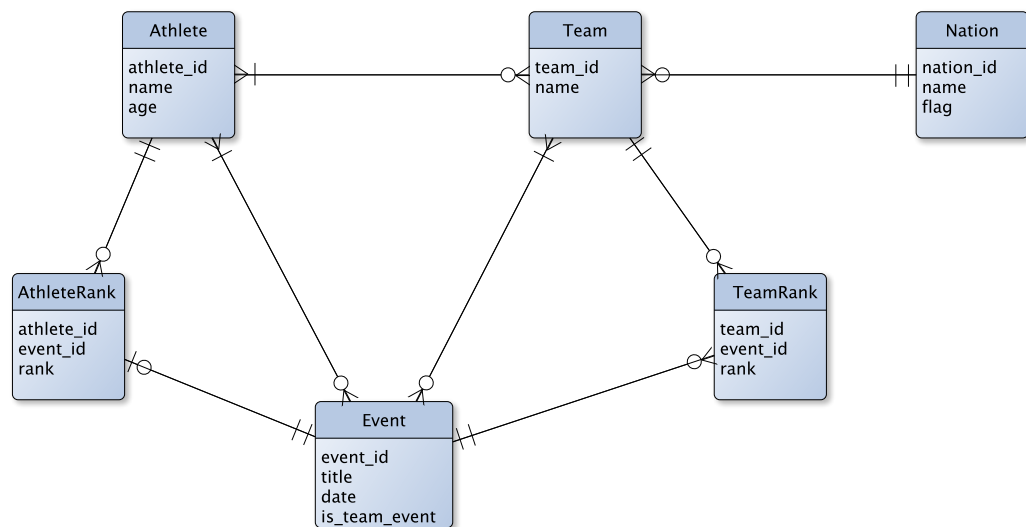
e) ☐ ★ Ersetzen Sie im vorigen Diagramm die min-max-Notation durch die Chen-Notation.

Lösung



- f) Zeichnen Sie nun das vorigen Diagramm in Crow's Foot Notation.

Lösung



- g) (Wie) Können Sie im Diagramm sicherstellen, dass bei Einzelereignissen ausschließlich einzelne Athleten teilnehmen und bei Teamereignissen ausschließlich Teams? Haben Sie das bei den vorherigen Diagrammen so gemacht?

Lösung



Die vorherigen Diagramme der Musterlösung berücksichtigen das nicht. Werden diese Diagramme strukturell nicht umgestellt, dann kann dieses Problem nicht

berücksichtigt werden und muss später in der Datenbank oder schon in der Applikationsschicht mit geeigneten Methoden sichergestellt werden. Sind hingegen Änderungen erlaubt, dann könnte der Sachverhalt mittels isa-Hierarchie umgesetzt werden. Dabei werden SingleEvent und TeamEvent werden zwei Entitäten, die von Event erben, und die Relationships werden entsprechend umgesetzt.

Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)

Aufgabe 1 (Modellierung — IT-Dienstleister)

[5 Punkte]

Für einen IT-Dienstleister sollte eine Datenbank modelliert werden, die es erlaubt, Bestellungen für Computer zu verwalten. Dieser IT-Dienstleister liefert ausschließlich vorinstallierte Computer an seine Kunden aus. Jedem Computer wird ein Hostname zugewiesen, der diesen eindeutig identifiziert. Hier ist zu beachten, dass sich dieser Name nachträglich ändern kann. Der Dienstleister weist jedem Gerät, das ausgeliefert werden soll, eine Seriennummer zu, um eventuelle Support-Anfragen abwickeln zu können. Zusätzlich wählt der Kunde ein gewünschtes Betriebssystem und dessen Version aus. Es ist auch möglich, weitere Software auf diesem Computer installieren zu lassen, welche für das entsprechende Betriebssystem in verschiedenen Versionen vorliegen kann. Für das Betriebssystem und die installierte Software gibt es jeweils eine passende Beschreibung. Der IT-Dienstleister unterscheidet bei seinen Kunden zwischen Privat- und Firmenkunden. Beide Arten von Kunden haben eine Adresse, die aus Straße, Hausnummer, Stadt, Postleitzahl und Staat besteht. Die zwei Arten von Kunden unterscheiden sich darin, dass Firmenkunden nur einen Firmennamen enthalten, während ein Privatkunde einen Vor- und Nachnamen, sowie ein Geburtsdatum besitzt.

Sobald eine Bestellung eines Kunden eingeht, wird diese mit einem Bestelldatum, Lieferdatum und den bestellten Computern im System gespeichert. Dabei muss sichergestellt werden, dass das Lieferdatum mindestens einen Tag nach dem Bestelldatum liegt. Eine Bestellung kann auch mehrere Geräte beinhalten. Zusätzlich sollte eine Lieferadresse für die Bestellung hinterlegt werden. Als Rechnungsadresse wird stets die Adresse des Kunden verwendet. Bevor eine Bestellung akzeptiert wird, muss überprüft werden, ob die angegebene Liefer- und Rechnungsadresse auch wirklich existieren.

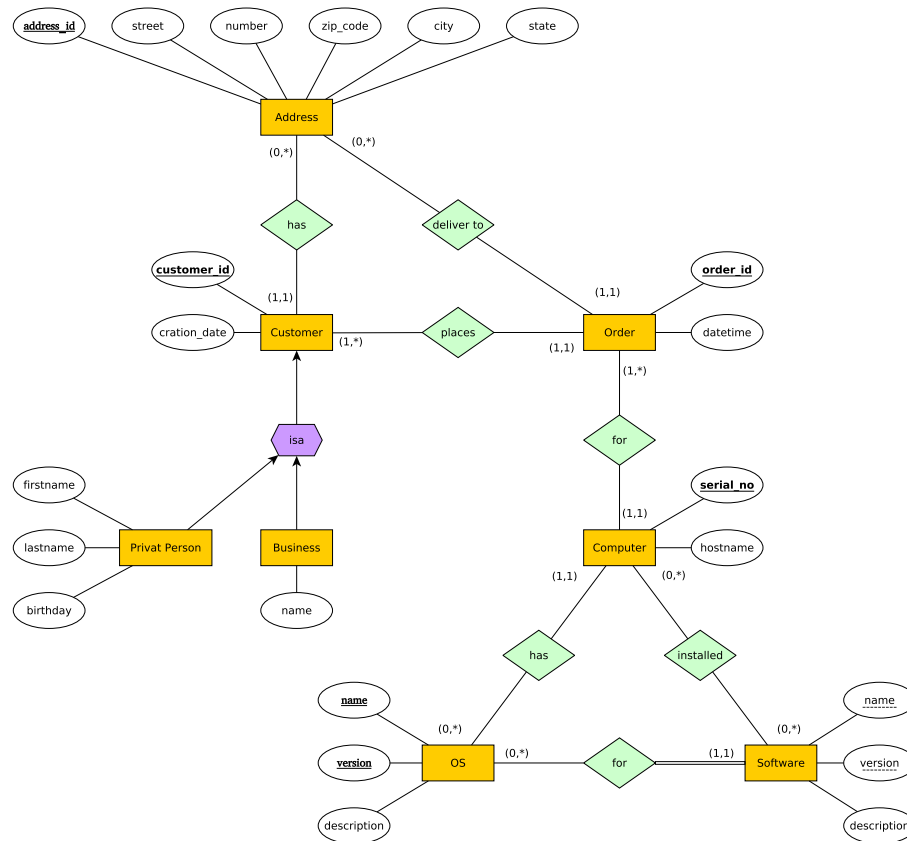
Um zu ermöglichen, dass die Daten eines Privatkunden nach einer vorgegebenen Zeit gelöscht werden können, ist es notwendig, das Registrierungsdatum zu speichern. Ein Kunde wird immer dann gelöscht, wenn es im System keine Bestellungen dieses Kunden mehr gibt. Eine Bestellung wird zwei Jahre nach erfolgter Lieferung gelöscht.

- a) 3 Punkte Entwerfen Sie ein Entity-Relationship-Diagramm, das den beschriebenen Sachverhalt modelliert. Zeichnen Sie die benötigten Entitäten, ihre Attribute und die zwischen den Entitäten existierenden Beziehungen in *min-max Notation*. Verwenden Sie englische Bezeichner.

Geben Sie dieses Diagramm in Form einer Grafik (PDF, JPG oder PNG) ab.

- b) 2 Punkte Geben Sie an, welche der obigen Aussagen/Spezifikationen sich nicht mithilfe dieses ER-Modells modellieren lassen.

Geben Sie die Ihre Lösung in Form einer Textdatei ab.



Folgende Aussagen lassen sich im ER-Modell nicht modellieren:

- Sicherstellen, dass das Lieferdatum mindestens einen Tag nach dem Bestelldatum liegt.
- Sicherstellen, dass die Liefer- und Rechnungsadresse existieren.
- Das Löschen von Bestellungen nach zwei Jahren.
- Das Löschen von Privatkunden, sobald keine Bestellung mehr im System existiert.

Aufgabe 2 (Einrichten der Datenbank)

[5 Punkte]

In dieser Aufgabe sollen Sie das am Anfang dieses Zettels (unter Tools) beschriebene Datenbankmanagementsystem (vorzugsweise auf Ihrem eigenen Computer) einrichten und darauf gegebene Datenbankabfragen mit einem SQL Client ausführen.

- a) 4 Punkte Setzen Sie PostgreSQL 11.5 auf und führen Sie anschließend folgende Query, die Sie auch in `postgres_query_1.sql` ^{OLAT} finden, aus:

```
1  select *
2  from pg_config
3  where name = 'VERSION';
```

Geben Sie das Ergebnis als TXT oder CSV Datei ab.

Lösung



name	setting
VERSION	PostgreSQL 11.5 (Debian 11.5-1.pgdg90+1)

(1 row)

- b) 1 Punkt Erstellen Sie eine Datenbank mit dem Namen db_ps_sheet01 indem Sie die Query in `database_create.sql` ausführen:

```
1 create database db_ps_sheet01;
```

Spielen Sie nun den Dump in `database_dump.sql` ein und führen Sie anschließend die SQL Query in `postgres_query_2.sql` aus:

```
1 select
2     name,
3     subdivision,
4     two_letter,
5     subcountry_name
6 from
7     country
8 join subcountry on
9     country.two_letter = subcountry.country
10 where
11     two_letter = 'AT';
```

Geben Sie das Ergebnis als TXT oder CSV Datei ab.

Hinweis



Um die Datenbank zu füllen, spielen wir die Daten aus einer Datei in eine Datenbank ein. Diese Datei enthält einen sogenannten Datenbankdump (kurz Dump)^a. Eine Möglichkeit, diesen Dump einzuspielen ist, die Dump-Datei in den selben Ordner wie das `docker-compose.yml` zu kopieren und dort dann den Befehl

```
cat database_dump.sql | docker-compose exec -T db psql -U postgres -d db_ps_sheet01
```

ausführen. Dabei ist `database_dump.sql` der Name der Datei die den Dump enthält und `db_ps_sheet01` ist der Name der Zieldatenbank in die dieser Dump eingespielt werden soll. Es kann sein, dass diese beiden Argumente (Namen) an die Aufgabenstellung anzupassen sind.

^a<https://de.wikipedia.org/wiki/Datenbankdump>

Lösung



name	subdivision	two_letter	subcountry_name
Austria	Burgenland	AT	1
Austria	Kärnten	AT	2

Austria		Niederösterreich		AT		3
Austria		Oberösterreich		AT		4
Austria		Salzburg		AT		5
Austria		Steiermark		AT		6
Austria		Tirol		AT		7
Austria		Vorarlberg		AT		8
Austria		Wien		AT		9

(9 rows)

Wichtig: Laden Sie bitte Ihre Lösung in OLAT hoch und geben Sie mittels der Ankreuzliste auch unbedingt an, welche Aufgaben Sie gelöst haben. Die Deadline dafür läuft am Vortag des Proseminars um 23:59 (Mitternacht) ab.