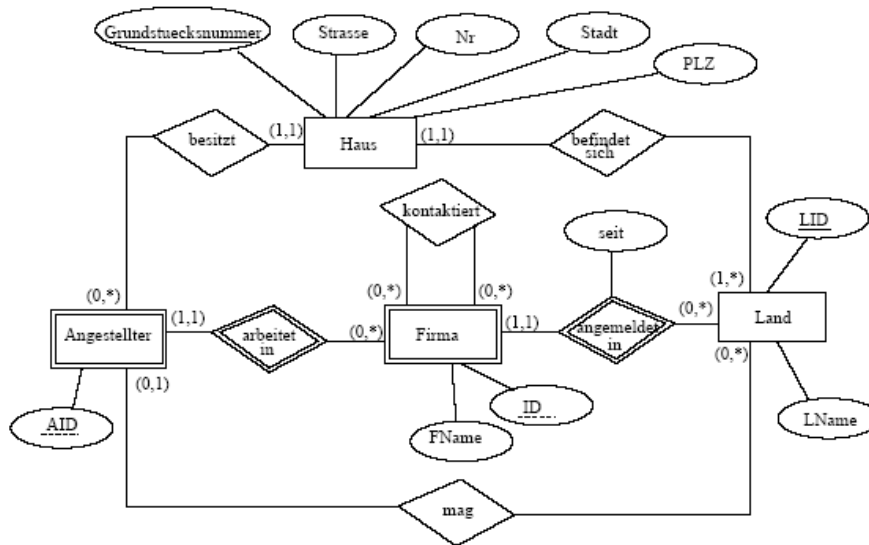


3. Übung zur Vorlesung
Einführung in Datenbanksysteme
Datenbanken für die Bioinformatik

1. Aufgabe

Geben Sie für das Modell das relationale Datenbankschema an.



2. Aufgabe

Die folgende Ausgabe beschreibt ein typisches Problem aus der *realen* Welt: Zu einem vorhandenen (dateibasierten) Informationssystem haben Sie die unten gegebenen Informationen erhalten. Das zugrunde liegende ER-Diagramm ist im Betrieb nicht mehr auffindbar. Ihre Aufgabe besteht nun darin, ein „entsprechendes“ ER-Diagramm zu entwickeln, das als Basis für eine Übernahme der Daten in eine Datenbank dienen soll. Es soll also das ER-Diagramm entwickelt werden, dessen Abbildung zu der unten gegebenen Dateistruktur hätte führen können. Vereinfachend sind die identifizierenden Felder der Dateien unterstrichen.

The order processing application to be reengineered is based on the following set F of files {person, pilot, engineer, planetype, plane, servicetype, servicework, hangar, machinegroup, machine, installedin, worksin, offers, flies}. Syntactic domains, e.g., INT, CHAR(20), DATE, are omitted in the following descriptions.

- person(personId, name, address, salary);
 a person is identified by a number (personId) and characterized by a name, an address, and a salary.
- pilot(pId, licenseNum);
 a pilot is a person, identified by a person number (pId) and additionally characterized by a license number.
- engineer(eId, ssNo, specialSkill);
 an engineer is a person, identified by a person number (eId) and additionally characterized by a social security number (ssNo) and a special salary skill.

- planeType(modelName, capacity, weight);
a planeType is identified by a model name (modelName) and characterized by the capacity and the weight of the planes of this type.
- plane(model, regId, homeLocation);
a plane is identified by a registration number (regId) and a model name, planes are characterized by their geographical home locations.
- serviceType(sId, name, avgDuration, costs, machineId);
a serviceType is identified by a number (sId). It is characterized additionally by its name, the average duration of the service, the costs of the service and the machine number (machineId), that is needed to carry out services of this type.
- serviceWork(planeModel, engineer, planeregId, service, startDate, duration);
the file for serviceWork contains for a given combination of service type (referenced by service), engineer, and airplane the services carried out. Each service started at a given day and had a given duration e.g., "30 hours have been taken for service 7 at March 27, 1999 on the airplane 4711 of type 'Boeing 747-121'".
- hangar(hId, area);
a hangar is identified by a hangar number (hId). It is characterized by its area.
- machineGroup(mId, responsEngineer);
a machineGroup is identified by a machine group number (mId). An engineer is responsible for the machine group.
- machine(mId, num, lastCheck);
a machine is identified by a machine group number (mId) and a running number per machine group (num). It is characterized by the time of the last security check.
- installedIn(hangar, machinegroup, since);
the file installedIn contains for each hangar the corresponding machine group it holds and the date since when this machine group is installed there.
- worksIn(engineer, hangar);
the worksIn file contains for each engineer the corresponding hangars s/he is allowed to work in.
- offers(engineer, sid, quality);
the file offers contains for each engineer the service types s/he can offer and in which quality the service can be offered.
- flies(pilot, planeType, flightHours);
the flies file contains for each pilot the corresponding plane type s/he can fly and the information about the flight hours accumulated by the pilot for that particular plane type.

Das E/R-Diagramm sollte zumindest enthalten: Entity types (starke und schwache) mit Attributen und Schlüsseln, etwaige Hierarchiebeziehungen, Beziehungen mit Kardinalitäten in min-max-Notation und alle notwendigen Attribute.

3. Aufgabe

Die Tabellen Student und Vortragstermin werden mit folgenden SQL Befehlen erzeugt:

```
create table Student (  
  Name varchar2(10),  
  Geburtsdatum date,  
  Geschlecht char CONSTRAINT nn_Geschlecht NOT NULL,  
  StudentId number(5),  
  CONSTRAINT StudentId_pk PRIMARY KEY (StudentId),  
  CONSTRAINT Geschlecht_ck CHECK (Geschlecht in ('M', 'W'))  
);
```

```
create table Vortragstermin(  
  TerminID int,  
  Beginn Date,  
  Ende Date,  
  Beschreibung varchar(80),  
  VortragenderID number(5),  
  CONSTRAINT TerminID_pk PRIMARY KEY (TerminID),  
  CONSTRAINT VortragenderID_fk  
  FOREIGN KEY (VortragenderID) REFERENCES Student(StudentId)  
);
```

- (a) Welche der folgenden Datensätze können in die Tabellen, die anfänglich leer sind, eingefügt werden? Falls das Einfügen klappt, ist die Zeile als vorhanden zu betrachten. Falls ein Datensatz nicht eingefügt werden kann, geben Sie eine Begründung an.
- i. ('Fichte', '12-MAY-66', 0, 3456) → Student
 - ii. ('Carnap', '07-FEB-75', 'W', 6456) → Student
 - iii. (3, '11-MAY-00', '11-MAY-00', 'Ethik', 6455) → Vortragstermin
 - iv. ('Feuerbach', '07-FEB-11', 'M', 6455) → Student
 - v. (99, '01-MAY-24', '13-MAY-00', 'Erkenntnistheorie', 6455) → Vortragstermin
 - vi. ('Kant', '30-JUL-70', 'M', 6456) → Student
 - vii. (3, '14-MAY-00', '11-MAY-00', 'Glaube und Wissen', 3456) → Vortragstermin
 - viii. ('Fichte', '12-MAY-66', 'M', 3457) → Student
- (b) Auf welche Kardinalitäten lassen die Tabellendefinitionen schliessen? Begründen Sie kurz.
- (c) Wie müssen die Tabellen umgeformt werden, um die folgenden Eigenschaften auszudrücken:
- Jede/r Studierende muss mindestens einen Vortrag halten, mehrere sind auch möglich.
Jeder Vortrag kann von mehreren Studierenden zusammen gehalten werden, mindestens jedoch von einem/einer.

Beschreiben Sie Ihre Überlegungen detailliert. Geben Sie die resultierenden Tabellen mit allen notwendigen Bedingungen (constraints) an.