Asset Management

Datenbank Projektarbeit

Juan Bernstein & Oliver Czabala

Inhalt

1. Aufgabenbeschreibung

Der Kunde wünscht sich eine neue Datenbank um seine Arbeitsplätze zu inventarisieren. In diesem Inventarsystem soll ersichtlich sein, wo ein Mitarbeitender seinen Arbeitsplatz hat und welche Geräte ihm zugewiesen worden sind.

Ein Arbeitsplatz besteht meistens aus einem Notebook, einem Monitor sowie Peripherie.

Die Modelle der Geräte sind genormt, daher gibt es viele Geräte die vom selben Modell sind, jedoch ein abweichendes Anschaffungsdatum somit auch ein abweichendes Garantieablaufdatum.

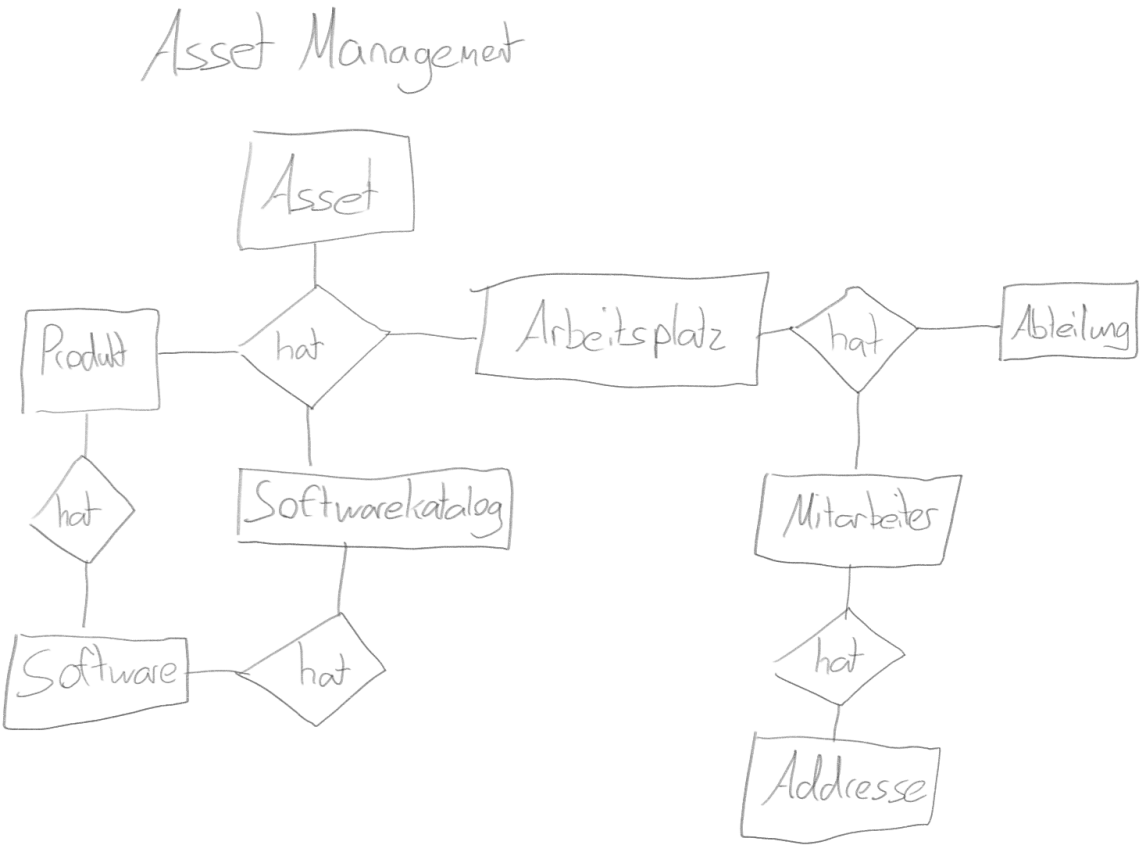
Um kosten bei der Softwarebeschaffung einzusparen, soll ebenfalls möglich sein einem Mitarbeiter verschiedene Software zuzuordnen.

Im Firmengebäude gibt es verschiedene Abteilungen welche zugewiesene Softwaregruppen besitzen.

Dem Kunden soll ermöglicht werden, Daten schnell auslesen zu können, Beispiele hierzu sind: z.B. zu sehen wann welche Geräte ersetzt würden müssen bei einem LifeCycle von 3 Jahren, sowie auch eine Statistik wie viele Mitarbeiter welche Software benutzen.

Es soll auch ersichtlich sein wann welche Software in welcher Softwareversion installiert worden ist, um zu sehen welche Geräte aktualisiert werden müssen.

2. ER-Modell



Beschreibung:

Das Asset Management ist folgendermassen unterteilt:
Die Hauptkategorien die wir bewirtschaften sind Mitarbeiter, sowie dazugehörige Assets.

Die Assets selber werden einem Arbeitsplatz zugeordnet, solch ein Arbeitsplatz hat zum Beispiel einen Laptop, einen Monitor, Eingabegeräte sowie eine Dockingstation.

Dem Asset selber ist ebenfalls Software zugeordnet, damit wir eine Kontrolle haben welche Software auf welchem Gerät installiert ist.

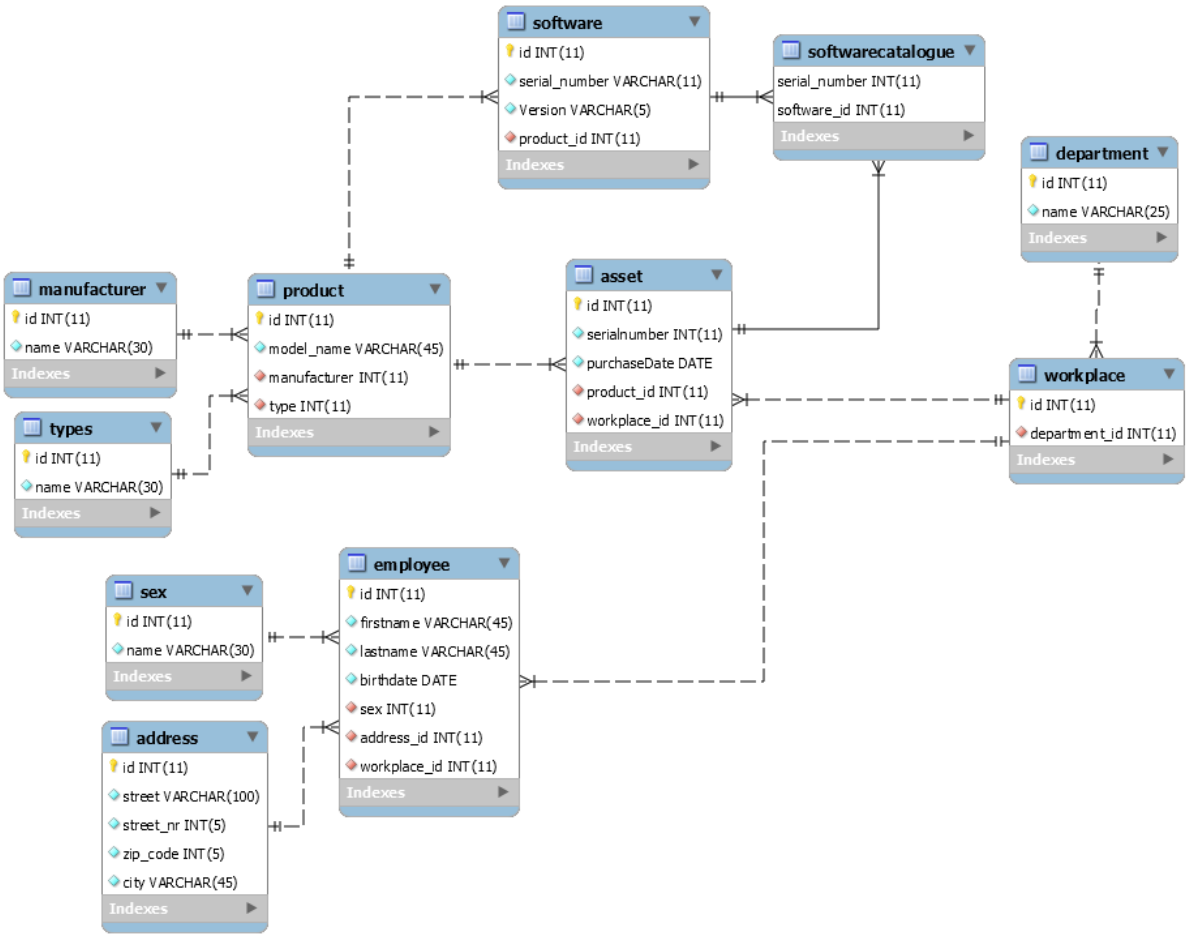
Mitarbeiter haben einen festen Arbeitsplatz / Büro und solch ein Arbeitsplatz ist einer Abteilung zugeordnet. Ein Mitarbeiter hat jedoch auch eine Adresse sowie persönliche Informationen.

3. Bestimmung der Entitäten

Die folgenden Entitäten wurden mit der Aufgabenbeschreibung erarbeitet.

Entität	Beschreibung	Beispiel
Product	Produkte werden hier erfasst	HP Elitebook 830 G5
Software	Verfügbare Software	Microsoft Office 365
Asset	Assets sind die Eigenschaften einzelner Geräte	Kaufdatum, Garantieablauf
Software Catalogue	Softwaregruppen welche zugeordnet werden können	Softwaregruppe Human Ressources
Workplace	Arbeitsplatz zugeordnet an User mit diversen Assets	Büro 1.01
Employee	Daten zum Mitarbeiter	Hans Mustermann
Department	Institute / Abteilung	Human Ressources
Address	Addresse der Employees	Hans Muster, Musterstrasse 1, 1111 Musterort

4. ER-Diagramm



Beschreibung:

Das ER-Diagramm wurde auf Basis von der erstellten Datenbank erstellt, dies wurde mit dem Programm MySQL Workbench per Reverse Engineering erstellt.

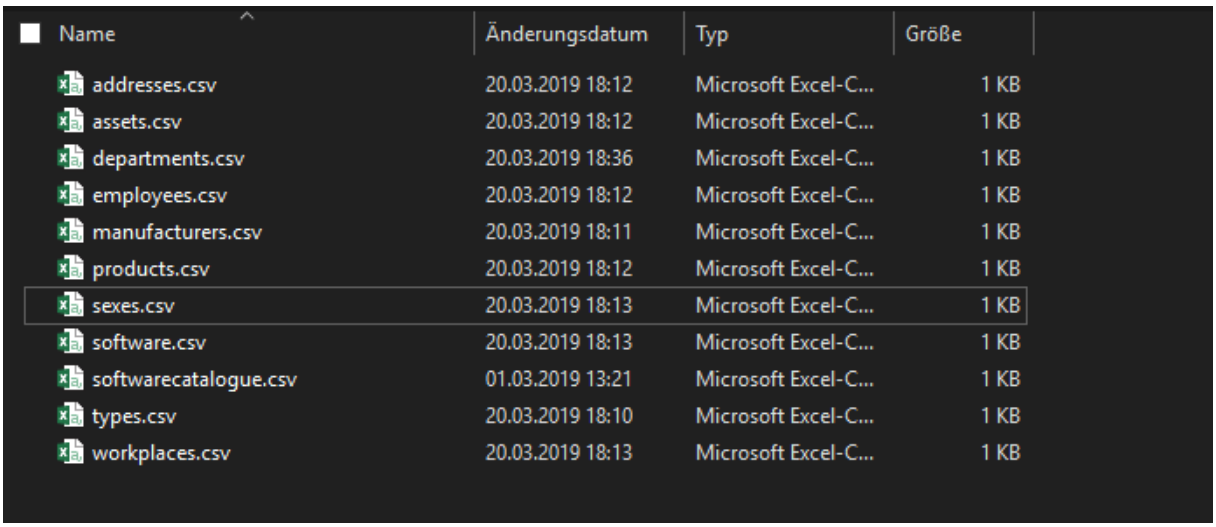
5. Erstellung der Entitäten

```
DROP DATABASE IF EXISTS Assetmgmt;
CREATE DATABASE Assetmgmt;
USE Assetmgmt;

CREATE TABLE product (
   id INT NOT NULL AUTO_INCREMENT,
   model_name VARCHAR(45) NOT NULL,
   manufacturer INT NOT NULL,
   type INT NOT NULL,
   PRIMARY KEY (id)
);
CREATE TABLE manufacturer (
   id INT NOT NULL AUTO_INCREMENT,
   name VARCHAR(30) NOT NULL,
   PRIMARY KEY (id)
);
CREATE TABLE types (
   id INT NOT NULL AUTO_INCREMENT,
```

```
  name VARCHAR(30) NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE asset (
  id INT NOT NULL AUTO_INCREMENT,
  serialnumber INT NOT NULL,
  purchaseDate DATE NOT NULL,
  product_id INT NOT NULL,
  workplace_id INT NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE software (
  id INT NOT NULL AUTO_INCREMENT,
  serial_number varchar(11) not null,
  Version VARCHAR(5) NOT NULL,
  product_id INT NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE softwarecatalogue (
  serial_number INT NOT NULL,
  software_id INT NOT NULL
);
CREATE TABLE workplace (
  id INT NOT NULL AUTO_INCREMENT,
        department_id INT NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE department (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(25) NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE employee (
  id INT NOT NULL AUTO_INCREMENT,
  firstname VARCHAR(45) NOT NULL,
  lastname VARCHAR(45) NOT NULL,
  birthdate DATE NOT NULL,
  sex INT NOT NULL,
  address_id INT NOT NULL,
        workplace_id INT NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE sex (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(30) NOT NULL,
  PRIMARY KEY (id)
);
CREATE TABLE address (
  id INT NOT NULL AUTO_INCREMENT,
  street VARCHAR(100) NOT NULL,
  street_nr INT(5) NOT NULL,
  zip_code INT(5) NOT NULL,
  city VARCHAR(45) NOT NULL,
  PRIMARY KEY (id)
);

ALTER TABLE product ADD FOREIGN KEY (manufacturer) REFERENCES manufacturer(id);
ALTER TABLE product ADD FOREIGN KEY (type) REFERENCES types(id);
ALTER TABLE asset ADD FOREIGN KEY (product_id) REFERENCES product(id);
ALTER TABLE asset ADD FOREIGN KEY (workplace_id) REFERENCES workplace(id);
ALTER TABLE software ADD FOREIGN KEY (product_id) REFERENCES product(id);
ALTER TABLE softwarecatalogue ADD constraint primary key (serial_number,software_id);
ALTER TABLE softwarecatalogue ADD FOREIGN KEY (serial_number) REFERENCES asset(id);
ALTER TABLE softwarecatalogue ADD FOREIGN KEY (software_id) REFERENCES software(id);
ALTER TABLE employee ADD FOREIGN KEY (workplace_id) REFERENCES workplace(id);
ALTER TABLE employee ADD FOREIGN KEY (sex) REFERENCES sex(id);
ALTER TABLE employee ADD FOREIGN KEY (address_id) REFERENCES address(id);
ALTER TABLE workplace ADD FOREIGN KEY (department_id) REFERENCES department(id);
```

6. Dateien einlesen / CSV Import

Im Script werden folgende Dateien eingelesen:



Name	Änderungsdatum	Typ	Größe
addresses.csv	20.03.2019 18:12	Microsoft Excel-C...	1 KB
assets.csv	20.03.2019 18:12	Microsoft Excel-C...	1 KB
departments.csv	20.03.2019 18:36	Microsoft Excel-C...	1 KB
employees.csv	20.03.2019 18:12	Microsoft Excel-C...	1 KB
manufacturers.csv	20.03.2019 18:11	Microsoft Excel-C...	1 KB
products.csv	20.03.2019 18:12	Microsoft Excel-C...	1 KB
sexes.csv	20.03.2019 18:13	Microsoft Excel-C...	1 KB
software.csv	20.03.2019 18:13	Microsoft Excel-C...	1 KB
softwarecatalogue.csv	01.03.2019 13:21	Microsoft Excel-C...	1 KB
types.csv	20.03.2019 18:10	Microsoft Excel-C...	1 KB
workplaces.csv	20.03.2019 18:13	Microsoft Excel-C...	1 KB

Die CSV Dateien sollten im Folder:

«C:/CSV» gespeichert werden, dort werden sie im Script importiert.

Die CSV Dateien werden kommagetrennt eingelesen, sollten Fehler bei der Eingabe entstehen entsteht eine Fehlermeldung welche verständlicherweise den Fehler ausgibt, Beispiel hierfür ist eine Meldung wie: «Wert fehlt», das kann dadurch passieren, dass ein Wert zu wenig hineingeschrieben worden ist.

Die Trennung geschieht via Kommazeichen und ein neuer Satz wird mit einer neuen Zeile (Break) voneinander getrennt.

Beispiel Inhalt anhand der Datei addresses.csv:

```
0,Baslerstrasse,14,4555,City1
0,Genfstrasse,8,4566,City2
0,Strassestrasse,4,4577,City3
0,Partystrasse,1,4588,City4
```

Script

```
/*Insert Data*/
LOAD DATA INFILE 'C:/CSV/types.csv'
INTO TABLE types
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/manufacturers.csv'
INTO TABLE manufacturer
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/products.csv'
INTO TABLE product
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/software.csv'
INTO TABLE software
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';
```

```
LOAD DATA INFILE 'C:/CSV/addresses.csv'
INTO TABLE address
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/departments.csv'
INTO TABLE department
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/workplaces.csv'
INTO TABLE workplace
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/sexes.csv'
INTO TABLE sex
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/employees.csv'
INTO TABLE employee
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/assets.csv'
INTO TABLE asset
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:/CSV/softwarecatalogue.csv'
INTO TABLE softwarecatalogue
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';
```

7. Views

View 1

Die erste View zeigt auf einem Blick eine Hardwareliste welche Hersteller sowie Modellname beinhaltet.

Script

```
create view Hardware_List as
select manufacturer.name as 'Hersteller', product.model_name as 'Modellname', asset.serialnumber as 'Serialnumber' from product
join manufacturer on manufacturer.id
join asset on asset.id
where product.id = asset.product_id and product.manufacturer = manufacturer.id;
```

View 2

Die zweite View erzeugt eine Auflistung aller Hardware die ein bestimmter Mitarbeiter zugeordnet hat.

Script

```
create view Employee_Hardware_List as
select employee.firstname as 'Firstname', employee.lastname as 'Lastname', asset.serialnumber as 'Seriennummer', product.model_name
as 'Modell' from asset
join product on product.id
join workplace on workplace.id
join employee on employee.id
where asset.product_id = product.id and asset.workplace_id = workplace.id and employee.workplace_id = workplace.id;
```

View 3

Die dritte View erzeugt eine Auflistung welche Mitarbeiter in einem Departement welche Assets hat.

Script

```
create view Departement_Employee_List as
select department.name as 'Department', employee.firstname as 'Firstname', employee.lastname as 'Lastname', workplace.id as 'Büro',
asset.serialnumber as 'Asset', product.model_name as 'Modell' from employee
join workplace on workplace.id
join department on department.id
join asset on asset.id
join product on product.id
where employee.workplace_id = workplace.id and workplace.department_id = department.id and asset.workplace_id = workplace.id and
asset.product_id = product.id;
```

View 4

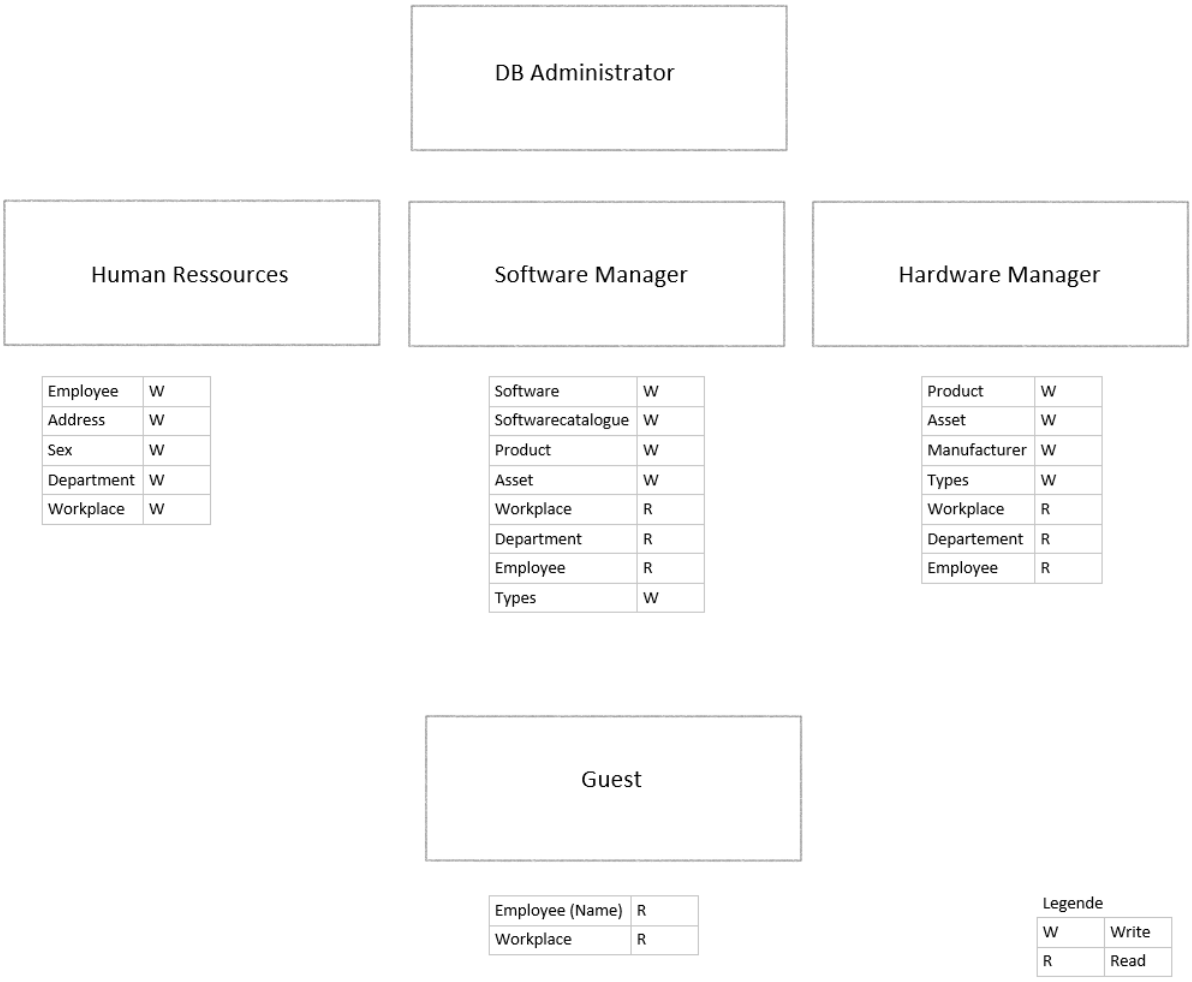
Die vierte View zeigt an welche Software am meisten von einem bestimmten Departement genutzt wird.

Script

```
create view Most_Used_Software as
select department.name as 'Department', product.model_name as 'Software Name', count(product.id) as 'Anzahl' from department
join workplace on workplace.id
join asset on asset.id
join softwarecatalogue on softwarecatalogue.serial_number
join software on software.id
join product on product.id
where department.id = workplace.department_id
and asset.workplace_id = workplace.id
and asset.id = softwarecatalogue.serial_number
and softwarecatalogue.software_id = software.id
and software.product_id = product.id
group by department.name;
```

8. Rights Management

Damit nicht alle Benutzer immer volle Rechte besitzen, haben wir uns für einen Approach entschieden bei dem es verschiedene Benutzergruppen gibt.



Employee	W
Address	W
Sex	W
Department	W
Workplace	W

Software	W
Softwarecatalogue	W
Product	W
Asset	W
Workplace	R
Department	R
Employee	R
Types	W

Product	W
Asset	W
Manufacturer	W
Types	W
Workplace	R
Departement	R
Employee	R

Employee (Name)	R
Workplace	R

Legende

W	Write
R	Read

- DB Administrator: Hat volle Berechtigungen
- Human Ressources, Softwaremanager sowie Hardware Manager: Sind dem DB Administrator direkt unterstellt und haben auf ihre benötigten Orte Berechtigungen.
- Guest: Kann sehen wo ein Mitarbeiter seinen Arbeitsplatz hat.

Beispiel im Script für einen User in den Human Ressources Abteilung:

Script

```
DROP USER IF EXISTS hr@’localhost’;
create user hr@’localhost’ identified by '123';
grant all privileges on employee to hr@’localhost’;
grant all privileges on address to hr@’localhost’;
grant all privileges on sex to hr@’localhost’;
grant all privileges on department to hr@’localhost’;
grant all privileges on workplace to hr@’localhost’;
```

9. Stored Procedures

Neuer Mitarbeiter anlegen

Mit dieser Prozedur soll ein neuer Mitarbeiter erfasst werden, benötigte Attribute sind folgende:

Entität **Employee**

Attribut	Datentyp	Beschreibung
Firstname	VARCHAR(45)	Vorname
Lastname	VARCHAR(45)	Nachname
Sex	ENUM	Geschlecht
Birthday	DATE	Geburtsdatum

Entität **Address**

Attribut	Datentyp	Beschreibung
Street	VARCHAR(100)	Adresse
HouseNr	VARCHAR(5)	Hausnummer
PLZ	INT	Postleitzahl
City	VARCHAR(45)	Stadt / Dorf

Script

```
DROP procedure IF EXISTS `newEmployee`;
DELIMITER //
CREATE PROCEDURE newEmployee(
Firstname varchar (45),
Lastname varchar (45),
Sex varchar(15),
Birthday date,
Street varchar(100),
HouseNr int,
PLZ int,
City varchar(45))
BEGIN
        insert into address (street,street_nr,zip_code,city)
        values (Street,HouseNr,PLZ,City);

        select @adressID := a.id from address as a
        where a.street = Street AND a.zip_code = PLZ
        Limit 1;

        select @sexID := s.id from sex as s
        where s.name = Sex;

        select @workplace_id := w.id from workplace as w
        join employee as e on w.id <> e.workplace_id
        group by e.id limit 1;

        insert into employee (firstname,lastname,birthdate,sex,address_id,workplace_id)
        values (Firstname,Lastname,Birthday,@sexID,@adressID,@workplace_id);
END//
DELIMITER ;
```

Neues Asset erfassen

Mit dieser Prozedur wird ein neues Asset angelegt mit dem Kaufdatum

Entität **Asset**

Attribut	Datentyp	Beschreibung
Serialnumber	INT	Seriennummer des Gerätes
PurchaseDate	DATE	Kaufdatum

Script

```
DROP procedure IF EXISTS `newAsset`;
DELIMITER //
CREATE PROCEDURE newAsset(
Firstname varchar (45),
Lastname varchar (45),
Serialnumber int(11),
BuyDate date,
ModellName varchar(45))
BEGIN
        select @productID := p.id from product as p
        where p.model_name = ModellName;

        select @workplaceID := e.workplace_id from employee as e
        where e.firstname = Firstname AND e.lastname = Lastname;

        insert into asset (serialnumber,purchaseDate,product_id,workplace_id)
        values (Serialnumber,BuyDate,@productID,@workplaceID);
END//
DELIMITER ;
```

10. Anhang

-MySQL Script als Datei «Database_Asset.sql»

-ER Diagramm als Bilddatei