

Lagerdatenbank Krankenhaus

Hussam Naji und Nikita Ostrovskii

Informatik in Kultur und Gesundheit, Hochschule für Technik und Wirtschaft

Einführung	2
Design	2
Anwendungsbereich	2
ER Modell	2
Weitere Erklärungen zu den Attributen	6
Implementierung	7
Beispielhafte Daten	8
SQL Solution	10
Verwendung der Datenbank	12
Zugangsrechte	12
Anträge stellen	12
Fazit	14
Quellen	15

Einführung

Unser Projekt befasst sich mit der Entwicklung einer Datenbank für eine medizinische Einrichtung zur Organisation, Speicherung und Verwaltung des Krankenhausinventars. Die Relevanz des Projekts hängt mit der Modellierung der Datenbank für einen spezifischen Berufszweig zusammen, in dem neben den üblichen logistischen Aufgaben auch die Besonderheiten der medizinischen Versorgung berücksichtigt werden müssen. Da beide Projektteilnehmer über Berufserfahrung in diesem Bereich verfügen, ist die Entwicklung eines Datenbankprojekts für ein Krankenhauslager ein sehr interessantes Unterfangen für uns, da es sowohl eine Umsetzung der Theorie in die Praxis als auch eine Vorbereitung für die Entwicklung größerer Projekte darstellt.

Design

Bei der Entwicklung einer Datenbank für ein Krankenhauslager ist es wichtig zu berücksichtigen, dass es nicht nur um die Lagerung von Krankenhausbedarf geht, sondern auch um dessen Organisation, die regelmäßige Auffüllung, die Führung strenger Aufzeichnungen und die Organisation der Zugänglichkeit für das Personal. Im Gegensatz zu kommerziellen Unternehmen besteht die Hauptaufgabe des Krankenhauslagers darin, die Patienten mit allem Notwendigen zu versorgen: Die Einrichtung muss stets über die notwendigen Medikamente, brauchbare Geräte und Kleidung für die Patienten in gutem Zustand verfügen. Dieser Querschnitt durch die Anforderungsanalyse zeigt uns bereits, dass die Erstellung einer Datenbank für ein Krankenhaus in hohem Maße davon abhängt, wie es verwaltungstechnisch organisiert ist und welche Präventionsmaßnahmen es ergreift. In einer weiteren Studie haben wir mehrere Ärzte in Berlin befragt, die uns mindestens zwei mögliche Optionen für die Organisation des Depots angeboten haben.

Anwendungsbereich

Unsere Datenbank wird nicht nur für die Buchhaltung und die Lagerung des

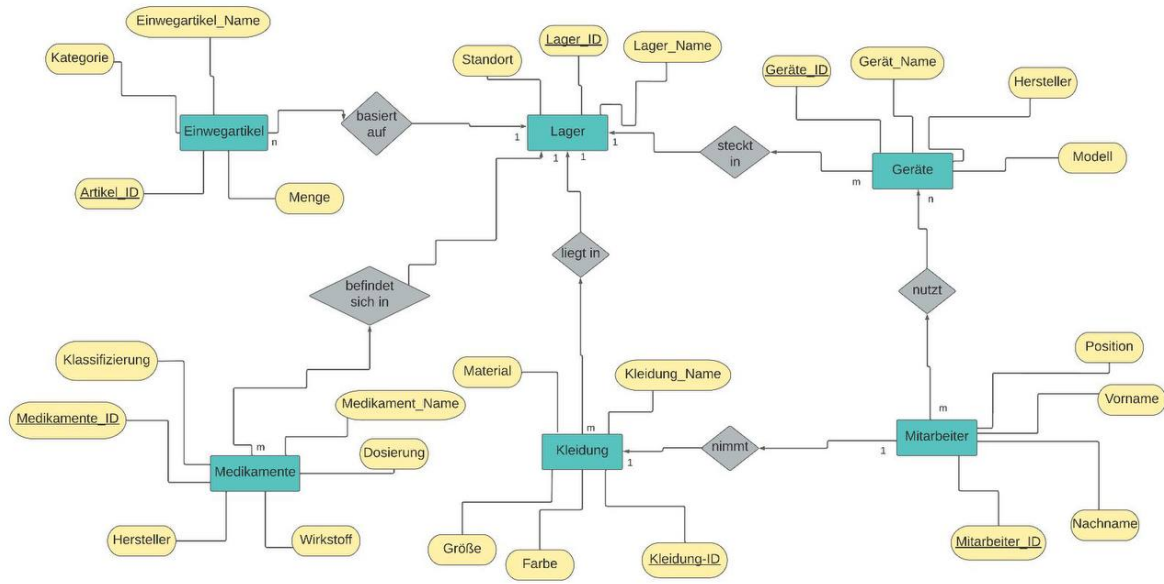
Krankenhausinventars verwendet, sondern auch für die administrative Kontrolle desselben. Dies bedeutet, dass eine Datenbank erstellt werden muss, auf die sowohl das Verwaltungs- und medizinische Personal des Krankenhauses als auch die einzelnen Mitarbeiter der Stationen, die für die Verwaltung und Auffüllung der Bestände zuständig sind, Zugriff haben. Natürlich sollte der Datenbankadministrator auch Fragen der Sicherheit, der Datentransparenz und des Zugriffs für die oben genannten Gruppen berücksichtigen. Wenn wir weiter in die Richtung dieser Idee gehen, könnten wir auch über die endgültige Umsetzung des Projekts in Form eines interaktiven Dashboards sprechen, das auf einem gemeinsamen taktischen Monitor den Mitarbeitern des Krankenhauses und der Abteilungen Informationen darüber liefert, welche Artikel sich im Bestand befinden, welcher Materialverbrauch im Krankenhaus stattfindet und zu welchen Terminen welcher Mitarbeiter Materialien verbraucht.

Entity-Relationship Modell

Um die Anforderungsanalyse zu vervollständigen und das ER-Modell zu erstellen, beschlossen wir, sowohl das medizinische Personal als auch die Mitarbeiter der Krankenhausverwaltung, die das Lager verwalten, zu befragen. Wir haben dann die grundlegenden Entitäten und Attribute für unser Projekt identifiziert.

Zuallererst müssen wir mit den spezifischen Bedingungen des Krankenhauses und seines Lagers beginnen: Aus medizinischen, hygienischen und organisatorischen Gründen müssen verschiedene Kategorien von Artikeln in getrennten Räumen gelagert werden. Dies liegt daran, dass Medikamente, wie auch die Blutbank, in Räumen mit einer bestimmten Temperatur aufbewahrt werden müssen, in denen auch Kühlschränke stehen müssen. Das bedeutet, dass das Krankenhauslager wiederum aus mehreren kleineren Lagern bestehen muss. Diese müssen eine eigene Identifikationsnummer haben, einen Namen, der die Kategorie der gelagerten Gegenstände widerspiegelt, und Informationen darüber, ob das Lager über freie Plätze verfügt und welche MitarbeiterInnen zum Lager beschrieben sind.

Bei einem Lager hat Mitarbeiter, die der Lagerabteilung zugewiesen sind und für die Lagerung von Gegenständen einer bestimmten Kategorie zuständig sind. Dementsprechend haben sie eine eindeutige Kennnummer, einen Namen, einen Rang und einen Status (in Betrieb bis/krankgemeldet bis/im Urlaub bis). Das Endergebnis dieser Analyse war unsere Annahme, dass die Schlüsselfigur in unserem Modell die Entität Lager sein sollte, in Bezug auf die die Lagermitarbeiter als Dienstleistungscharakter und unterstützendes Element dieses Schemas fungieren, das auf dem Funktionieren des Lagers aufbaut..



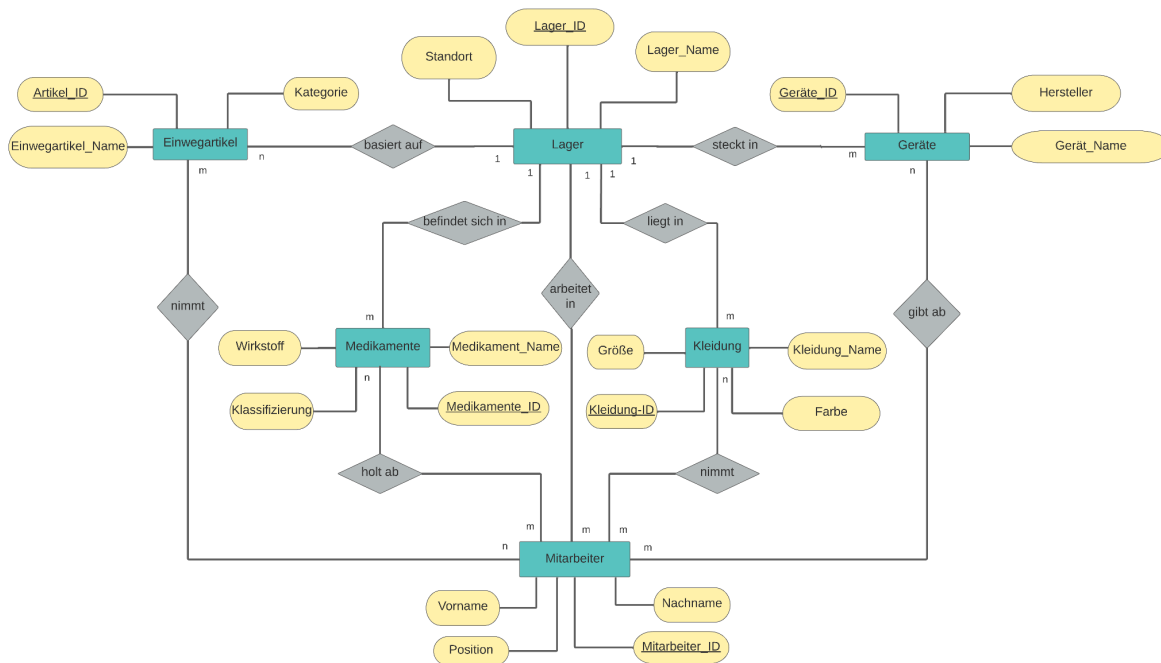
Das erste ER-Modell

Bei einer weiteren Analyse kamen wir jedoch zu dem Schluss, dass das System von unserer Seite aus weiter verbessert werden könnte, wenn wir:

1. Wir werden nicht nur das Entitätslager, sondern auch die Abteilung Mitarbeiter einrichten, die unabhängige Arbeitseinheiten darstellen sollte, die Entscheidungen vor Ort treffen. Dies würde dem System mehr Beweglichkeit verleihen.

2. Verbinden wir die beiden Entitäten Lager und Mitarbeiter durch eine ternäre Beziehung, die es uns nicht nur erlaubt, die Interaktion der Mitarbeiter mit dem Lager im Detail abzubilden, sondern auch die Ergebnisse ihrer Interaktion zu erfassen. Als zusätzliches Attribut dieser Beziehung können wir die Interaktionszeit angeben.

Für diese Option haben wir uns entschieden, da sie den Hauptpunkt der Konstruktion unseres Projekts darstellt.

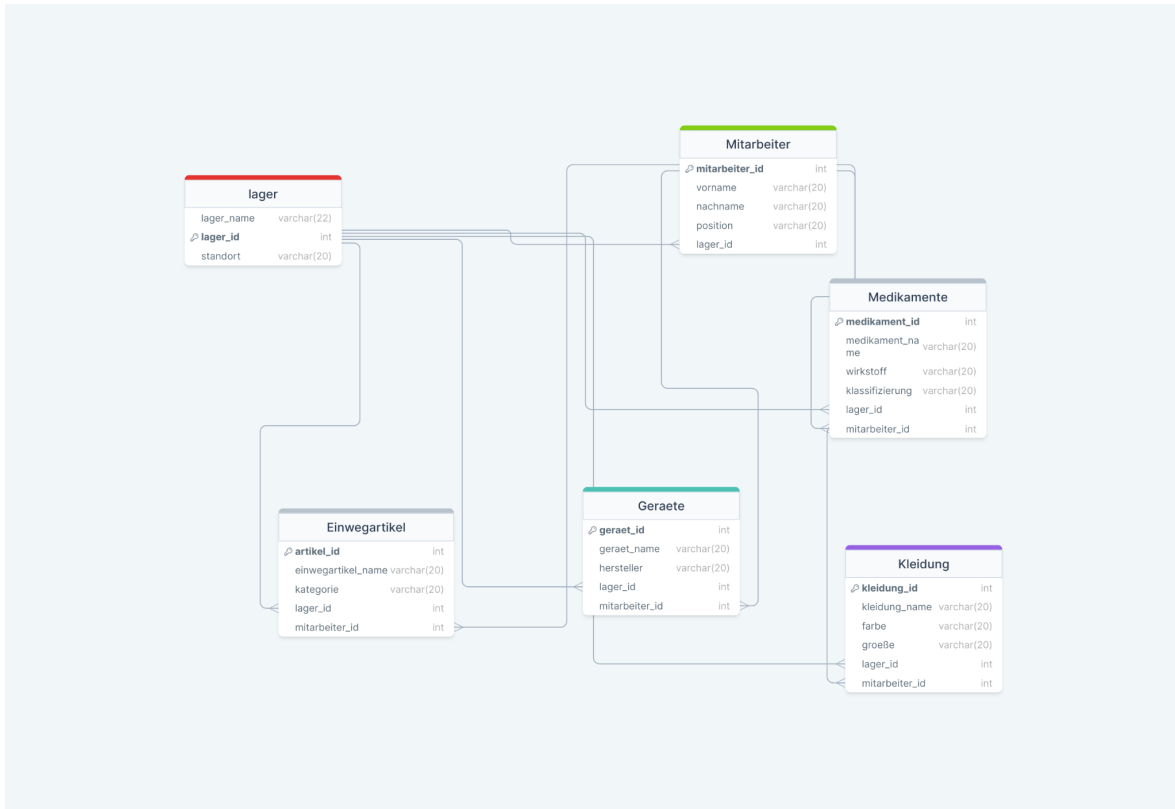


Das endliche ER-Modell

Im Gegensatz zur ersten Variante hat das daraus resultierende System folgende Vorteile:

1. Der Schwerpunkt des Systems verlagert sich auf die Interaktion zwischen dem Lagerpersonal und dem Lager.
2. Detaillierte Protokollierung der Aktionen der Mitarbeiter, die auf das Lagerinventar zugreifen.
3. Wir stellen dem Benutzer des Systems ein detaillierteres und eleganteres Layout zur Verfügung, das so nah wie möglich an der realen Situation ist und auf den natürlichen Beziehungen zwischen den Lagereinheiten basiert.

In diesem Stadium der Projektentwicklung wurde uns klar, dass wir zur weiteren Umsetzung übergehen und relationale Tabellen erstellen können, die die Attribute und Schlüssel der Entitäten in diesem ER-Modell anzeigen.



Tabellendiagramm für die Verbindung von ER-Modell mit einer PostgreSQL-Implementierung; beachten Sie, dass dies nicht auf unsere endgültige Implementierung hindeutet

Implementation

Nachdem wir unser endgültiges ER-Modell durchgearbeitet hatten, begannen wir mit der Implementierung der Datenbank selbst. Wir haben unsere Tabellenansicht aktualisiert, um den Prozess der Tabellenerstellung zu unterstützen.

Die Tabelle Lager ist eines der Lager des Krankenhauses, in dem die benötigten Artikel gelagert werden. Jedes Lager hat seine eigene eindeutige Identifikationsnummer Lager_ID, die der Primärschlüssel in dieser Tabelle ist. Außerdem verfügt die Tabelle über ein Attribut Lager_Name, das die Kategorie der im Lager gelagerten Artikel angibt. Aus medizinischen Gründen ist es verboten, Medikamente, Mitarbeiterkleidung, Einwegartikel und Geräte am selben Ort aufzubewahren. Jede dieser Kategorien sollte in einem separaten Lager aufbewahrt werden. Das Attribut Standort gibt wiederum den Standort des Lagers an, in diesem Fall kann es sich um eine Raumnummer im Krankenhausgebäude oder in einem angrenzenden Gebäude handeln. Da sich die Lager in verschiedenen Räumen befinden können, wurde beschlossen, ihre spezifische Adresse nicht als eindeutigen Identifikator zu verwenden: In diesem Fall mussten zusätzlich zur Raumnummer Adressinformationen hinzugefügt werden. Darüber hinaus stellt sich die Frage, was zu tun ist, wenn das Lager umzieht.

Die zweite Tabelle Mitarbeiter ist in unserer Datenbank für die Darstellung der für das Lager zuständigen Abteilung zuständig. Der Primärschlüssel ist auch hier die Mitarbeiter_ID, die jedem Mitarbeiter als eindeutige Nummer zugewiesen wird. Die Attribute Nachname und Vorname (fassen wir sie doch in einer Kategorie zusammen) zeigen den Namen des Mitarbeiters an, während das Attribut Position die Position des Mitarbeiters angibt, die besonders wichtig ist, weil sie auch seine Zugriffsrechte auf die Datenbank beeinflusst.

Der Kleidungstisch zeigt uns die Kleidung, die das Personal im Krankenhaus neuen Patienten und dem Krankenhauspersonal geben muss. Wie uns das Personal einer medizinischen Einrichtung in Berlin mitteilte, hat das Krankenhauspersonal bereits seine eigene Kleidung, die es während seiner Schicht selbst tragen muss. Jedes Kleidungsstück hat seine eigene eindeutige Kleidung_ID-Nummer, ein Farbattribut, die Größe der Kleidung und Kleidung_Name, der die Kategorie widerspiegelt, für die das Kleidungsstück bestimmt ist: Verwaltungspersonal, medizinisches Personal, Patienten mit entsprechendem Schweregrad, Reinigungspersonal.

Die Geräte-Tabelle ist für die technischen Geräte zuständig, die dem Krankenhaus zur Verfügung stehen. Jedes der Geräte hat eine eindeutige Geräte_ID, ein Hersteller-Attribut, ein Attribut Gerät_Name, das den technischen Zweck widerspiegelt, und ein Attribut Seriennummer, das sehr wichtig ist, wenn ein Gerät defekt ist und ausfällt. In den meisten Fällen kann es im Rahmen der Garantie repariert werden.

Die nächste Tabelle ist die Tabelle Medikamente. Auch hier hat jede Tabelleninstanz ihre eigene eindeutige Medikamente_ID-Nummer, das Attribut Medikamente spiegelt den Namen des Medikaments wieder, das Attribut Wirkstoff gibt ein Arzneistoff an und das Attribut Klassifizierung zeigt an, ob das Medikament zu einer Medikamentengruppe gehört, was bei der Suche nach dem richtigen Rezept sehr wichtig ist.

Die Tabelle für die letzte Entität Einwegartikel schließlich spiegelt genau das Inventar wider, das nach einmaligem Gebrauch sofort entsorgt werden muss. Da ein Krankenhaus eine Einrichtung ist, in der alles streng kontrolliert werden muss, müssen auch Einwegartikel eine eigene Identifikationsnummer haben. Die beiden anderen Attribute sind Einwegartikel_Name und Kategorie, wobei letztere die Zugehörigkeit des Artikels zu Kategorien wie Spritzen, Aderpressen, Verbandsmaterial usw. widerspiegelt.

Die wichtigste relationale Beziehung ist "arbeitet in". Sie spiegelt wider, welcher Mitarbeiter welchem Lager zugeordnet und für dieses verantwortlich ist. Da es sich um ein Krankenhaus handelt und jede Inventarkategorie ihre eigene spezifische Arbeitsweise hat, wird oft beschlossen, einem bestimmten Lager mehrere bestimmte Mitarbeiter zuzuweisen, um die Verwaltungsarbeit zu optimieren. Daher wird das Verhältnis zwischen diesen beiden Einheiten 1 zu N sein.

Eine solche Mitarbeiter-Lager-Beziehung impliziert zwei Arten von Beziehungen in der von uns betrachteten Datenbank: das Auffinden von Artikeln einer bestimmten Kategorie in dem jeweiligen Lager und die Verwendung von Artikeln dieser Kategorie durch die Mitarbeiter des Lagers.

Zur ersten Gruppe gehören die Relation "basiert auf", die die Lagerung von Einwegartikeln im Lager mit dem entsprechenden Namen darstellt, sowie "befindet sich in", "steckt in" und "liegt in". "Befindet sich in" steht für die Lagerung von Arzneimitteln in einem entsprechend benannten Lager, "liegt in" in ähnlicher Weise für Kleidung und "steckt in" für elektronische Geräte. Alle Relationen der obigen Gruppe sind N bis 1. Denn nach den Vorschriften des Krankenhauses darf jede Kategorie von Geräten (Geräte, Kleidungen, Medikamente und Einwegartikel) nur in einem Lager mit entsprechendem Namen gelagert werden.

Die "Holt ab"-Haltung der zweiten Gruppe deutet auf die Interaktion des Personals mit dem Medikamentenlager hin. Natürlich können zumindest einige Mitarbeiter mehrere für das Krankenhaus benötigte Medikamente mitnehmen.

Dasselbe gilt für die Beziehung "nutzt" (besser ein anderes Wort), wenn Mitarbeiter spezielle Ausrüstung mitnehmen, "zieht an" (ersetzt), wenn spezielle Kleidung für neue Mitarbeiter und Patienten ausgegeben wird, und für die Beziehung "nimmt", wenn ein Mitarbeiter oder eine Gruppe von Mitarbeitern Einwegartikel des Inventars mitnimmt.

Natürlich gibt es in all diesen Relationen N zu M.

Sample Data

Als Daten für die Tabelle haben wir beliebige Daten verwendet, die den oben beschriebenen Kategorien entsprechen, um eine reale Datenbanksituation zu simulieren und zu testen

```
create table lager (  
  lager_name varchar(22) not NULL,  
  lager_id integer not NULL,  
  standort varchar(20) not NULL,  
  primary key(lager_id)  
);
```

```
create table Mitarbeiter (  
  mitarbeiter_id integer not NULL,
```

```
vorname varchar(20) not NULL,  
nachname varchar(20) not NULL,  
position varchar(20) not NULL,  
lager_id integer not NULL,  
FOREIGN KEY (lager_id) REFERENCES lager(lager_id),  
primary key(mitarbeiter_id)  
);
```

```
create table Geraete (  
geraet_id integer not NULL,  
geraet_name varchar(20) not NULL,  
hersteller varchar(20) not NULL,  
lager_id integer not NULL,  
mitarbeiter_id integer not NULL,  
FOREIGN KEY (mitarbeiter_id) REFERENCES Mitarbeiter(mitarbeiter_id),  
FOREIGN KEY (lager_id) REFERENCES lager(lager_id),  
primary key(geraet_id)  
);
```

```
create table Kleidung (  
kleidung_id integer not NULL,  
kleidung_name varchar(20) not NULL,  
farbe varchar(20) not NULL,  
groesse varchar(20) not NULL,  
lager_id integer not NULL,  
mitarbeiter_id integer not NULL,  
FOREIGN KEY (mitarbeiter_id) REFERENCES Mitarbeiter(mitarbeiter_id),  
FOREIGN KEY (lager_id) REFERENCES lager(lager_id),  
primary key(kleidung_id)  
);
```

```
create table Medikamente(  
medikament_id integer not NULL,  
medikament_name varchar(20) not NULL,  
wirkstoff varchar(20) not NULL,  
klassifizierung varchar(20) not NULL,  
lager_id integer not NULL,  
mitarbeiter_id integer not NULL,  
FOREIGN KEY (mitarbeiter_id) REFERENCES Mitarbeiter(mitarbeiter_id),  
FOREIGN KEY (lager_id) REFERENCES lager(lager_id),  
primary key(medikament_id)  
);
```

```
create table Einwegartikel(  
artikel_id integer not NULL,  
einwegartikel_name varchar(20) not NULL,  
kategorie varchar(20) not NULL,  
lager_id integer not NULL,  
mitarbeiter_id integer not NULL,  
FOREIGN KEY (mitarbeiter_id) REFERENCES Mitarbeiter(mitarbeiter_id),  
FOREIGN KEY (lager_id) REFERENCES lager(lager_id),
```

```
primary key(artikel_id)  
);
```

Database Usage

Access Rights

Wie bereits erwähnt, ist ein wichtiges Element in der Struktur unserer Datenbank die Entität Mitarbeiter. Darüber hinaus dient unsere Datenbank nicht nur der Speicherung des Lagerbestands, sondern auch der administrativen Kontrolle des Lagers. Wir haben uns daher entschlossen, die Zugriffsrechte in zwei Kategorien zu unterteilen: den normalen Lagermitarbeiter, der nur allgemeine Informationen über das ihm zugewiesene Lager erhält, und die Rolle des Verwaltungsleiters: der Abteilungsleiter, der für die Organisation aller Lager und des Verwaltungspersonals des Krankenhauses verantwortlich ist. Zum Beispiel für die Einkaufsabteilung, deren Mitarbeiter wissen müssen, wie viele Artikel noch in das Lager aufgenommen werden müssen und wie viel Inventar insgesamt in den einzelnen Krankenhauslagern verbraucht wird.

Fazit

Der Umfang und die Größe unserer Datenbank wurden auch durch die zeitlichen und ressourcenbedingten Beschränkungen beeinflusst, die durch die Länge des Semesters und unsere Erfahrung mit Datenbanken im Allgemeinen bedingt waren. Mit mehr Zeit und Ressourcen gäbe es viele Möglichkeiten, unsere auf Kunstmuseen ausgerichtete Datenbank weiter auszubauen.

Da muss man Ideen schreiben.

Quelle

- [1] Nikita Ostrovskii, Hussam Naji, “Lagerdatenbank-krankenhaus,” [Online].
Verfügbar: <https://github.com/throughTheGalaxy/Lagerdatenbank-Krankenhaus> [Acces. July 2023].
- [2] The PostgreSQL Global Development Group, accessed 3rd Jan 2023, Documentation,
<https://www.postgresql.org/docs/>.