Patientenverwaltungssystem

 |

Datenbankentwurf

Sultan & Mahmoud

2023

**1 Aufgabenstellung**

Entwickeln Sie eine Datenbank für eine beliebige Hausarztpraxis, welche die Daten der Patienten verwaltet und die Abrechnung mit der Krankenkasse ermöglicht.

**2 Grobstruktur der Miniwelt**

*Einleitung:*

Warum ein Patientenverwaltungssystem wichtig ist: Die steigende Anzahl von Patienten und die Notwendigkeit einer besseren Organisation von Patientendaten in Krankenhäusern und Arztpraxen haben zur Entwicklung eines Patientenverwaltungssystems geführt.

*Was wir mit dem Patientenverwaltungssystem erreichen wollen*:

Unser Ziel ist es, die Erfassung und Verwaltung von Patientendaten zu optimieren, um die medizinische Versorgung zu verbessern und den Arbeitsablauf in medizinischen Einrichtungen zu erleichtern.

*Welche Anforderungen haben wir?*

Welche Funktionen das System haben soll: Wir haben die Anforderungen an das Patientenverwaltungssystem definiert, z.B. die Erfassung und Aktualisierung von Patientendaten, die Verwaltung von Behandlungsverläufe und die Abrechnung.

Wer das System nutzen wird: Wir haben die Hauptbenutzergruppen identifiziert, z.B. Ärzte, medizinisches Personal und Verwaltungspersonal, und ihre spezifischen Anforderungen berücksichtigt.

**3 Feinstruktur der Miniwelt**

**Was ist ein Patientenverwaltungssystem und wozu dient es?**

Ein Patientenverwaltungssystem ist eine spezielle Software, die Krankenhäusern und Arztpraxen hilft, Informationen über Patienten, deren Behandlung und Abrechnung mit der Krankenkasse zu verwalten. Es funktioniert wie ein digitales Tagebuch für Ärzte und Hausärzte, in dem sie alle wichtigen Informationen über ihre Patienten speichern können. Das System ermöglicht es Ärzten, Diagnosen und Behandlungspläne für jeden Patienten zu erfassen. Es hilft auch bei der Abrechnung, indem es die Daten über die erbrachten Leistungen für die Krankenkasse aufzeichnet. Auf diese Weise hilft das System den Ärzten, den Überblick über ihre Patientenbehandlung zu behalten und sicherzustellen, dass die richtigen Abrechnungen an die Krankenkasse gesendet werden.

**Einleitung:** Die vorliegende Dokumentation beschreibt die Feinstruktur eines Patientenverwaltungssystems, das von uns als Datenbankprojekt entwickelt wurde. Das System ermöglicht die effiziente Verwaltung von Patienteninformationen, Hausärzten und Krankenkassen. Durch die Nutzung einer relationalen Datenbank werden die verschiedenen Entitäten und ihre Beziehungen zueinander strukturiert erfasst und ermöglichen eine zuverlässige und effektive Datenverwaltung.

Die Datenbank basiert auf einem ER-Modell, das die Hauptentitäten "Patient", "Hausarzt" und "Krankenkasse" sowie deren Attribute und Beziehungen definiert. Jede Entität verfügt über spezifische Attribute, die relevante Informationen enthalten, wie z. B. Namen, IDs, Kontaktdaten und weitere spezifische Details. Die Beziehungen zwischen den Entitäten ermöglichen die Verknüpfung der Daten und unterstützen die Durchführung von Aufgaben wie der Zuweisung von Hausärzten zu Patienten und der Abrechnung von Hausärzten durch Krankenkassen.

Im folgenden Abschnitt werden die Tabellen und ihre Struktur im Detail erläutert, um ein besseres Verständnis der Datenbankarchitektur zu ermöglichen.

1. *Tabelle "Patient":*

* Patient\_ID (Primärschlüssel): Eindeutige ID zur Identifizierung jedes Patienten.
* P\_Vorname: Vorname des Patienten.
* P\_Nachname: Nachname des Patienten.
* Geburtsdatum: Das Datum, an dem der Patient geboren wurde.
* Telefonnummer: Die Telefonnummer des Patienten.

1. *Tabelle "Hausarzt":*

* Arzt\_ID (Primärschlüssel): Eindeutige ID zur Identifizierung jedes Hausarztes.
* Fachgebiet: Das Fachgebiet des Hausarztes.
* A\_Vorname: Vorname des Hausarztes.
* A\_Nachname: Nachname des Hausarztes.
* Adresse: Die Adresse des Hausarztes (mehrwertiges Attribut, bestehend aus Straße, Hausnummer, Postleitzahl).

1. *Tabelle "Krankenkasse":*

* Krankenkasse\_ID (Primärschlüssel): Eindeutige ID zur Identifizierung jeder Krankenkasse.
* K\_Name: Der Name der Krankenkasse.
* Versicherungsart: Die Versicherungsart der Krankenkasse (mehrwertiges Attribut, z. B. "Gesetzlich", "Privat" ).
* Hauptsitz: Der Hauptsitz der Krankenkasse.

*Die Beziehungen zwischen den Entitäten:*

* Die Beziehung "Behandlung" verbindet die Tabelle "Patient" und "Hausarzt" über die Fremdschlüssel "Patient\_ID" und "Arzt\_ID". Dies zeigt an, welcher Patient von welchem Hausarzt behandelt wird.
* Die Beziehung "Abrechnung" verbindet die Tabelle "Hausarzt" und "Krankenkasse" über den Fremdschlüssel "Krankenkasse\_ID". Dies zeigt an, welche Krankenkasse den Hausarzt abrechnet.

3.2 Die relevanten Objekte

|  |  |
| --- | --- |
| Arzt\_ID | Eindeutige Kennzeichnung zur Zuordnung des Arztes |
| Wertebereich | Ganze Zahlen |
| Werte-Einschränkung | Nein |
| Wertevergabe | Automatische Wertevergabe |
| Default-Wert | Nein |
| NULL-Wert erlaubt | Nein |
| Duplikate erlaubt | Nein |
| Primärschlüssel | Ja |
| Sonst |  |
| Beispielwerte |  |

|  |  |
| --- | --- |
| Fachgebiet | Fachrichtung des Mediziners |
| Wertebereich | Text von Maximal 50 Zeichen |
| Werte-Einschränkung | Nein |
| Wertevergabe | Nein |
| Default-Wert | Nein |
| NULL-Wert erlaubt | Nein |
| Duplikate erlaubt | Nein |
| Primärschlüssel | Nein |
| Sonst |  |
| Beispielwerte | Allgemeinmedizin |

|  |  |
| --- | --- |
| A\_Vorname | Gibt den Vornamen des Arztes an |
| Wertebereich | Text von Maximal 50 Zeichen |
| Werte-Einschränkung | Nein |
| Wertevergabe | Nein |
| Default-Wert | Nein |
| NULL-Wert erlaubt | Nein |
| Duplikate erlaubt | Ja |
| Primärschlüssel | Nein |
| Sonst |  |
| Beispielwerte | Joachim |

|  |  |
| --- | --- |
| A\_Nachname | Gibt den Nachnamen des Arztes an |
| Wertebereich | Text von Maximal 50 Zeichen |
| Werte-Einschränkung | Nein |
| Wertevergabe | Nein |
| Default\_Wert | Nein |
| NULL-Wert erlaubt | Nein |
| Duplikate erlaubt | Ja |
| Primärschlüssel | Nein |
| Sonst |  |
| Beispielwerte | Winterscheidt |

|  |  |
| --- | --- |
| Adresse | Arzt\_ID, Straße, Hausnummer, PLZ |
| Wertebereich | Mehrere Text von jeweils maximal 75 Zeichen |
| Werte-Einschränkung | Muss aus Straße, Hausnummer und PLZ bestehen |
| Wertevergabe | Manuelle Eingabe |
| Default\_Wert |  |
| NULL-Wert erlaubt |  |
| Duplikate erlaubt |  |
| Primärschlüssel |  |
| Sonst |  |
| Beispielwerte |  |

3.3 Beziehungen

Aus der Analyse des Patientenverwaltungssystems ergeben sich die nachfolgend beschriebenen relevanten Beziehungen für die Patientenverwaltungssystem-Datenbank.

3.3.1 Abrechnung

Abrechnung ist eine M:N Beziehung zwischen Krankenkasse und Hausarzt.

Es können mehrere Hausärzte bei verschiedenen Krankenkassen abrechnen und umgekehrt können mehrere Krankenkassen mit mehreren Hausärzten in Beziehung stehen, damit abgerechnet werden kann.

|  |  |  |
| --- | --- | --- |
|  | Beteiligung | Kardinalität |
| Krankenkasse | N | M |
| Eine Krankenkasse kann mit mehreren Hausärzten in Beziehung stehen. Es können also mehrere Hausärzte von derselben Krankenkasse abgerechnet werden. |  |
| Hausarzt | M | N |
|  |  |  |

**4 Entity Relationship-Modell**

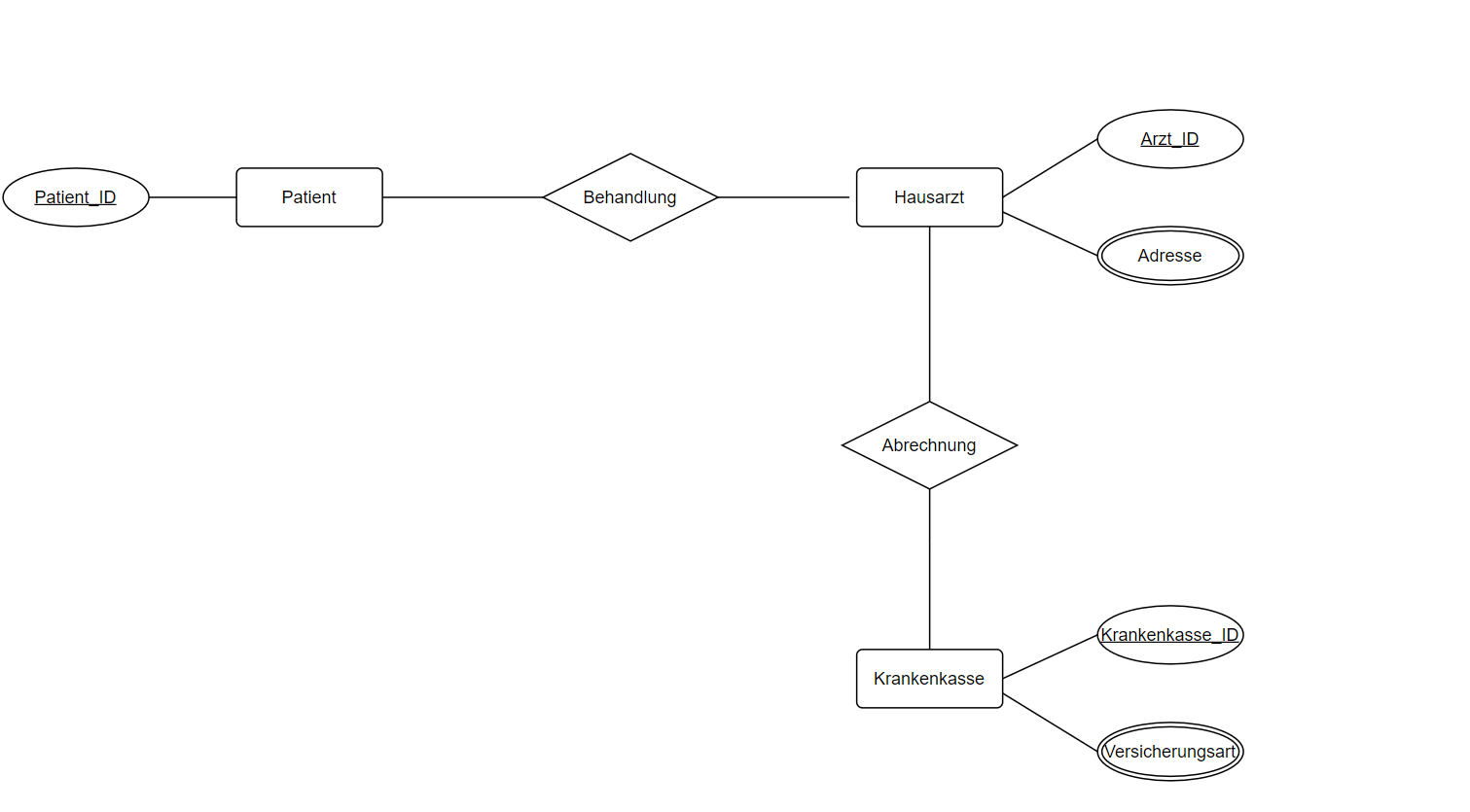
**4.1 Gesamtdarstellung:**

Abbildung 4.1: Entity Relationship-Modell: Gesamtdarstellung

4.2 Detail-Darstellung

Jede Entität wird nachfolgend detailliert mit den dazugehörigen Attributen dargestellt.

4.2.1 Hausarzt

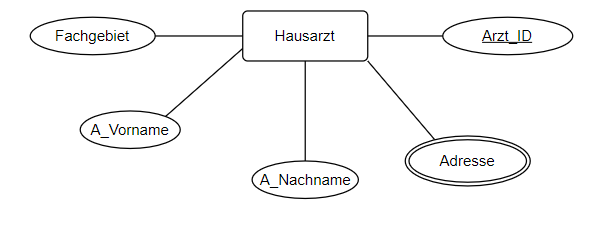
****

Abbildung 4.2: Entity Relationship-Modell: Hausarzt

4.2.2 Patient

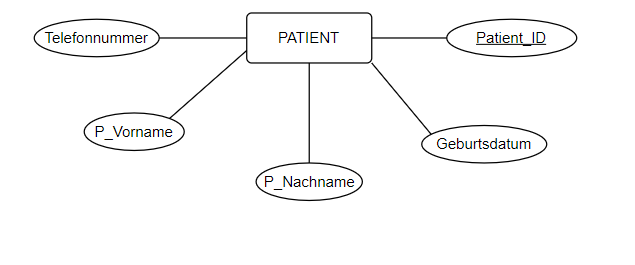


Abbildung 4.3: Entity Relationship-Modell: Patient

4.2.3 Krankenkasse

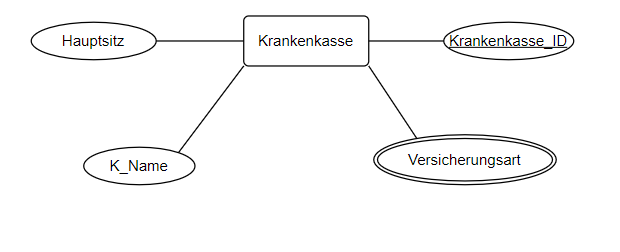


Abbildung 4.4: Entity Relationship-Modell: Glas

**5 Relationenmodell**

Patient (Patient\_ID, P\_Vorname, P\_Nachname)

Behandlung (Patient\_IDnn, Arzt\_IDnn)

Adresse ( Adresse, Arzt\_ID)

Hausarzt (Arzt\_ID, Fachgebiet, A\_Vorname, A\_Nachname)

Abrechnung ( Arzt\_ID, Krankenkasse\_ID)

Versicherungsart (Versicherungsart, Krankenkasse\_ID)

Krankenkasse (Krankenkasse\_ID, K\_Name, Hauptsitz)

**6 Normalisierung**

An dieser Stelle sollen nur die ersten 3 Normalformen betrachtet werden.

1. Normalform

Alle Attribute weisen nur atomare Werte auf.

2. Normalform:

Da die 1. Normalform erfüllt ist und alle Primärschlüssel aus künstlichen Schlüsseln gebildet werden, liegt die 2. Normalform vor.

3. Normalform:

Die 2. Normalform liegt vor und es existieren keine Nichtschlüsselattribute, die transitiv vom Primärschlüssel abhängen. Somit ist die 3. Normalform ebenfalls gegeben.

**7 Datenbankobjekte**

Zum Erstellen der Datenbankobjekte wird ein Benutzer benötigt, der über ausreichende Rechte verfügt.

7.1 Tabellen

Für die Datenbank: Hausarzt werden sämtliche Tabellen erstellt.

Skript:

Struktur > erstellen

Aufgabe des Skriptes:

Da die Tabellen existieren könnten, werden sie zunächst gelöscht und anschließend erstellt.

7.2 Test-Datensätze

Zum Testen der Datenbank werden einige typische Datensätze in die entsprechenden Tabellen eingefügt.

Skript:

Struktur > einfuegen.sql

Aufgabe des Skriptes:

Um immer von denselben Testdatensätzen auszugehen, werden aus den Tabellen zunächst die eventuell vorhandenen Datensätze gelöscht und anschließend eingefügt.

**8 Datenmanipulation**

8.1 Patienten einfügen

Ein neuen Patienten aufnehmen.

Skript:

Manipulation > Patienten\_hinzufuegen.sql

Aufgabe des Skriptes:

Der Benutzer wird zur Eingabe des Vornamen für den Patienten und deren Nachnamen aufgefordert. Die Daten werden in die Tabelle t\_patient eingefügt. Der benötigte Primärschlüssel wird basierend auf den vorhandenen Primärschlüsselwerten der Tabelle t\_patient berechnet

8.2 Krankenkasse hinzufuegen

Einen neue Krankenkasse aufnehmen.

Skript:

Manipulation > Krankenkasse\_hinzufuegen.sql

Aufgabe des Skriptes:

Der Benutzer wird zur Eingabe der Krankenkasse für den Patienten aufgefordert.

8.3 Behandlung hinzufuegen

Eine Behandlung wird dem Patienten hinzugefuegt.

Skript:

Manipulation > Behandlung\_hinzufuegen.sql

Aufgabe des Skriptes:

Ein Patient kann sich z.B. bei Rückenschmerzen zusätzliche Behandlungen wünschen. Dies wird dann der gängigen Behandlung hinzugefügt.

Zur Laufzeit wird die entsprechende ID für den Hausarzt bzw. des Patienten ermittelt und dann in die Tabelle t\_Behandlung eingefügt.

**9 Datenabfrage**

Eine Auflistung der derzeitigen Datenabfragen sollen nachfolgend vorgestellt werden.

9.1 Patientendaten anzeigen

Welche Patienten haben wir?

Skript:

Lesen > Patientendaten\_anzeigen.sql

Aufgabe des Skriptes:

Ausgabe aller Patientendaten, wie z.B. Vorname, Nachname, Geburtsdatum und Telefonnummer

9.2 Behandlungsdaten anzeigen

Welche Behandlungsdaten des jeweiligen Patienten liegen vor?

Skript:

Lesen > Behandlungsdaten\_anzeigen.sql

Aufgabe des Skriptes:

Es wird angezeigt, was beim Patienten behandelt wurde. Selbstverständlich mit Verweis auf den Vor- und Nachnamen, sowie der Patient\_ID.

9.3 Krankenkassendaten Anzeigen

Welche Krankenkassendaten liegen zum jeweiligen Patienten vor?

Skript:

Lesen > Krankenkassendaten\_anzeigen.sql

Aufgabe des Skriptes:

Es wird der Krankenkassen-Name und Hauptsitz der Krankenkasse angezeigt.

**10 Menügestaltung**

Die Verwaltung der Datenbankobjekte, die Datenmanipulation und die Datenabfrage kann menügesteuert ausgeführt werden. Die Menügestaltung wird mit den Möglichkeiten, die SQL\*Plus bietet, realisiert.

Start.bat

-**Menue**

0.sql

1.sql

2.sql

3.sql

beenden.sql

erstellen.sql

hauptmenue.sql

lesen.sql

manipulieren.sql

start.sql

**Lesen**

0.sql

1.sql

2.sql

3.sql

Behandlungsdaten\_anzeigen.sql

Krankenkassendaten\_anzeigen.sql

Menue\_lesen.sql

Patientendaten\_anzeigen.sql

Zurueck.sql

**Manipulation**

0.sql

1.sql

2.sql

3.sql

Behandlung\_hinzufuegen.sql

Krankenkasse\_hinzufuegen.sql

menue\_manipulation.sql

Patient\_hinzufuegen.sql

zurueck.sql

**Struktur**

0.sql

1.sql

2.sql

einfuegen.sql

erstellen.sql

menue\_struktur.sql

zurueck.sql

10.2 Menü-Übersicht

