

Kontrollierte Anbindung eines KI-Agenten an Unternehmensdaten über MCP

Architektur, Datenzugriff und Sicherheitsaspekte am Beispiel von Claude Code und einem MCP-Server

Christian Drapatz

29. Mai 2026

Vorwort

Künstliche Intelligenz hält zunehmend Einzug in betriebliche Abläufe. Mit KI-Agenten wie Claude Code entsteht der Wunsch, interne Unternehmensdaten nicht nur anzuzeigen, sondern gezielt abzufragen, fachlich einzuordnen und mit weiteren Informationen zu verknüpfen.

Gerade bei besonders schützenswerten Daten – etwa Gesundheits-, Sozial- oder Versicherungsdaten – darf ein solcher Zugriff jedoch niemals unkontrolliert erfolgen. Dieses Dokument zeigt am Beispiel eines MCP-Servers, wie sich ein KI-Agent kontrolliert an eine Unternehmensdatenbank anbinden lässt, welche Komponenten dabei zusammenwirken und welche Sicherheits- und Architekturüberlegungen maßgeblich sind.

Es richtet sich an Entwickler, Softwarearchitekten und technisch interessierte Leser, die einen praxisnahen Einstieg in die kontrollierte Datenanbindung von KI-Agenten suchen.

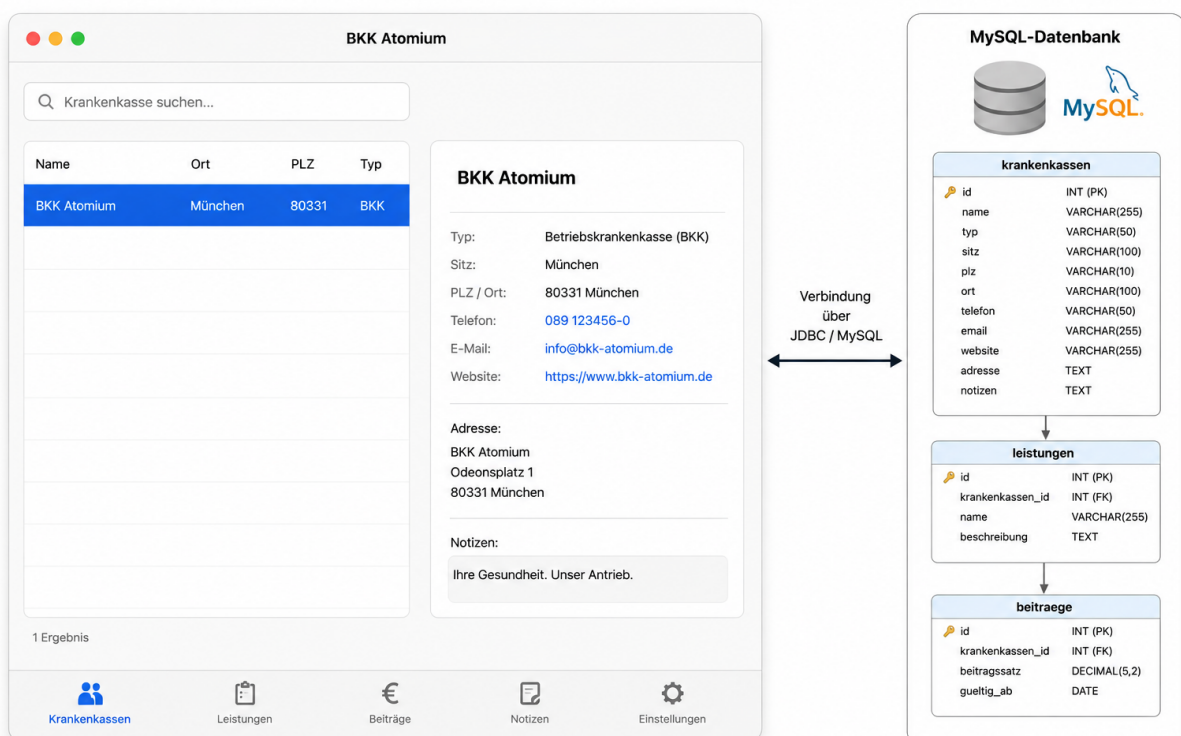
Inhaltsverzeichnis

1	Ausgangssituation	4
2	Herausforderungen und Risiken	5
3	Anbindung eines KI-Agenten an Unternehmensdatenbanken über MCP	6
4	Aktueller technischer Stand (Test-Szenario)	7
5	Beispiel für eine erweiterte KI-Abfrage	8
6	Beispielergebnis	8
7	Fachliche Einordnung	9
8	Fazit.....	9

1 Ausgangssituation

Für den Zugriff auf die Daten aus der Unternehmensdatenbank existiert bereits eine Softwarelösung. Diese Anwendung ermöglicht es, Versicherte anhand bestimmter Suchkriterien zu finden und die dazugehörigen Daten strukturiert anzuzeigen.

Die bestehende Software steht für den klassischen Ansatz: Daten werden fachlich aufbereitet und in einer Oberfläche für Anwender bereitgestellt. In einer produktiven Umgebung greift eine solche Anwendung üblicherweise nicht direkt auf die Datenbank zu, sondern über ein Backend-System, das Authentifizierung, Berechtigungsprüfung und Fachlogik kapselt. Für diese Demonstration wird der Zugriff bewusst vereinfacht und direkt auf der Datenbankebene betrachtet, um den Fokus auf die wesentlichen Funktionen und Abläufe zu legen.



Der neue Ansatz mit einem KI-Agenten und einem MCP-Server ersetzt diese Software nicht, sondern erweitert das bestehende Prinzip um eine zusätzliche Zugriffsmöglichkeit: Daten lassen sich gezielt abfragen, fachlich einordnen und mit weiteren Informationen kombinieren. Der Zugriff bleibt dabei kontrolliert und erfolgt ausschließlich über definierte Schnittstellen.

Für die Demonstration wurde eine vollständige Beispielumgebung aufgebaut, die den Zugriff eines KI-Agenten auf Unternehmensdaten über das Model Context Protocol (MCP) zeigt.

Folgende Komponenten wurden umgesetzt:

1. Aufbau einer Krankenkassen-Datenbank

- Erstellung eines Datenmodells mit Beispieltabellen und Testdaten.
- Ablage von Versicherten-, Vertrags- und Stammdaten zur Demonstration typischer Geschäftsprozesse.

2. Entwicklung eines Krankenkassensystems für macOS

- Programmierung einer nativen macOS-Anwendung mit direkter Anbindung an die Datenbank.
- Umsetzung einer grafischen Benutzeroberfläche zur Anzeige und Verwaltung der Daten.

3. Implementierung typischer Fachfunktionen

- Datensätze suchen
- Datensätze anlegen
- Datensätze bearbeiten
- Datensätze löschen
- Stammdaten verwalten
- Datenbankinhalte anzeigen und aktualisieren

4. Entwicklung eines MCP-Servers

- Implementierung eines eigenen MCP-Servers.
- Anbindung des MCP-Servers an die Krankenkassen-Datenbank.
- Bereitstellung definierter Werkzeuge (Tools) für Datenabfragen und Datenänderungen.
- Kapselung der Datenbankzugriffe über eine kontrollierte Schnittstelle.

5. Anbindung eines KI-Agenten (Claude Code)

- Verbindung von Claude Code mit dem MCP-Server.
- Nutzung der vom MCP-Server bereitgestellten Werkzeuge durch den KI-Agenten.
- Durchführung von Datenabfragen und Geschäftsprozessen über natürliche Sprache.
- Demonstration, wie Unternehmensdaten kontrolliert und sicher für KI-Systeme verfügbar gemacht werden können.

Ziel der Demonstration ist es zu zeigen, wie bestehende Unternehmensanwendungen, Datenbanken und KI-Agenten über das Model Context Protocol miteinander verbunden werden können, ohne der KI einen direkten Zugriff auf die Datenbank gewähren zu müssen.

2 Herausforderungen und Risiken

Die größte Herausforderung besteht darin, einem KI-Agenten Zugriff auf Unternehmensdaten zu geben, ohne daraus einen unkontrollierten Datenzugang entstehen zu lassen. Gerade bei besonders schützenswerten Daten, zum Beispiel Gesundheits-, Sozial- oder Versicherungsdaten, reicht eine rein technische Anbindung nicht aus. Der Zugriff muss fachlich, rechtlich und technisch klar begrenzt werden.

Ein zentrales Risiko ist, dass der KI-Agent mehr Daten erhält, als für die konkrete Aufgabe notwendig sind. Deshalb muss strikt geregelt werden, welche Daten überhaupt verfügbar sind, welche Felder ausgeblendet oder maskiert werden und welche Abfragen erlaubt sind. Der Grundsatz sollte immer lauten: so wenig Daten wie möglich, so viele wie nötig.

Ein weiteres Risiko liegt in falschen oder zu weit gefassten Abfragen. Wenn ein Agent beliebige SQL-Abfragen ausführen könnte, wären Massenabfragen oder unbeabsichtigte

Datenabflüsse möglich. Deshalb sollte der Zugriff nicht direkt auf Tabellen erfolgen, sondern nur über freigegebene Views, Stored Procedures oder klar definierte MCP-Funktionen.

Auch die Nachvollziehbarkeit ist eine wichtige Herausforderung. Es muss jederzeit erkennbar sein, welcher Nutzer welche Anfrage gestellt hat, welches Tool verwendet wurde und welche Daten abgerufen wurden. Ohne Audit-Logging lassen sich Fehler, Missbrauch oder Datenschutzvorfälle später kaum sauber bewerten.

Zusätzlich besteht das Risiko, dass sensible Daten in Prompts, Logs oder KI-Antworten landen. Deshalb müssen Eingaben und Ausgaben geprüft, sensible Inhalte maskiert und Protokolle datensparsam geführt werden. Besonders kritisch ist auch die Frage, ob Daten an externe KI-Anbieter übertragen werden oder ob die Verarbeitung vollständig in einer kontrollierten Umgebung erfolgt.

Ein KI-Agent kann sehr hilfreich sein, aber er darf nicht wie ein normaler Datenbanknutzer mit breiten Rechten behandelt werden. Er braucht eine eigene Sicherheitsschicht mit klaren Regeln, begrenzten Funktionen, Rollenprüfung, Protokollierung und technischer Zugriffsbeschränkung.

Kurz gesagt: Der Nutzen liegt in der intelligenten Verarbeitung und Verknüpfung von Unternehmensdaten. Das Risiko liegt darin, dass ohne klare Grenzen besonders schützenswerte Daten ungewollt offengelegt, falsch verarbeitet oder nicht mehr nachvollziehbar genutzt werden.

3 Anbindung eines KI-Agenten an Unternehmensdatenbanken über MCP

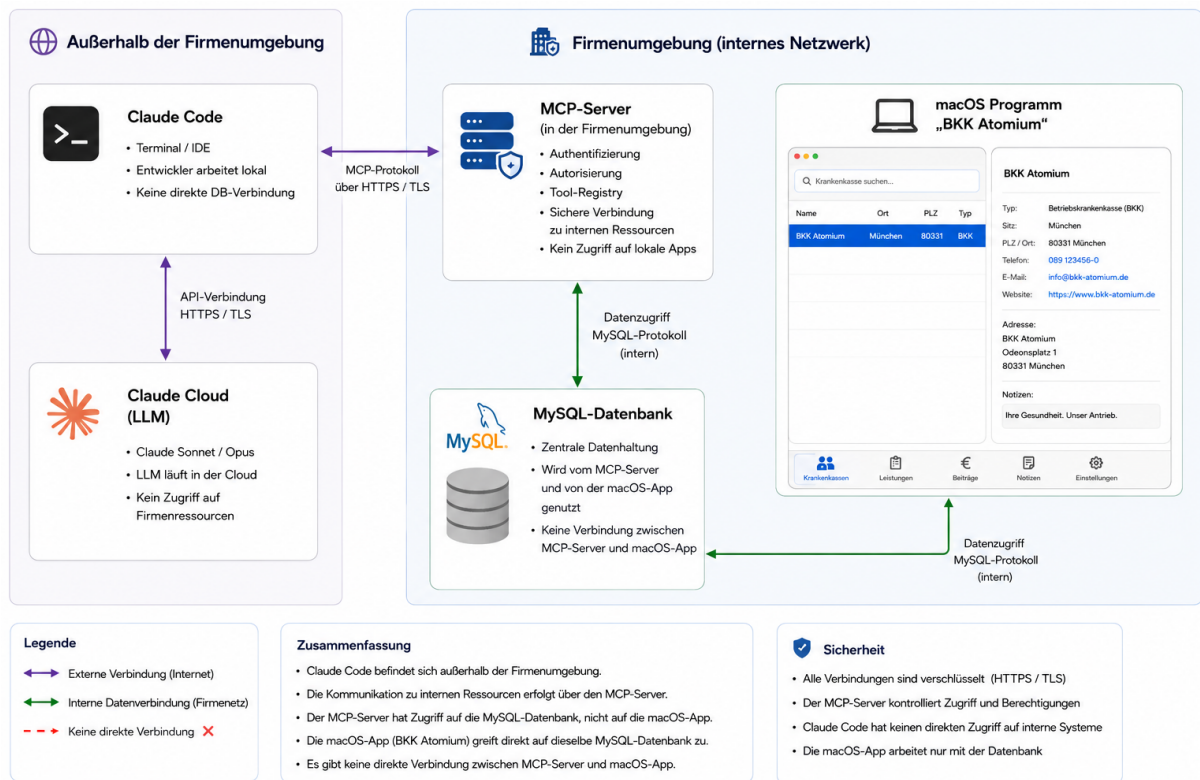
Im Folgenden wird betrachtet, wie ein KI-Agent, zum Beispiel **Claude Code**, kontrolliert mit Unternehmensdatenbanken verbunden werden kann, um interne Daten gezielt abzufragen und weiterzuverarbeiten.

Ein einfaches Beispiel wäre:

„Zeige mir die Adressen zur KVNR K145145875.“

Damit ein KI-Agent solche Anfragen bearbeiten kann, benötigt er Zugriff auf die passenden Unternehmensdaten. Statt diese fest in den Kontext einzubetten, wird der Agent über eine standardisierte Schnittstelle an die Datenquelle angebunden. Dafür eignet sich das Model Context Protocol (MCP), ein offenes Protokoll, über das ein KI-Agent externe Werkzeuge und Datenquellen in einheitlicher Form ansprechen kann.

Der MCP-Server stellt dem Agenten eine Reihe klar definierter Funktionen (Tools) bereit. Der Agent ruft diese Funktionen auf, erhält aber zu keinem Zeitpunkt direkten Zugriff auf die zugrunde liegende Datenbankstruktur. Welche Daten gelesen werden und wie der Zugriff erfolgt, bestimmt allein der MCP-Server.

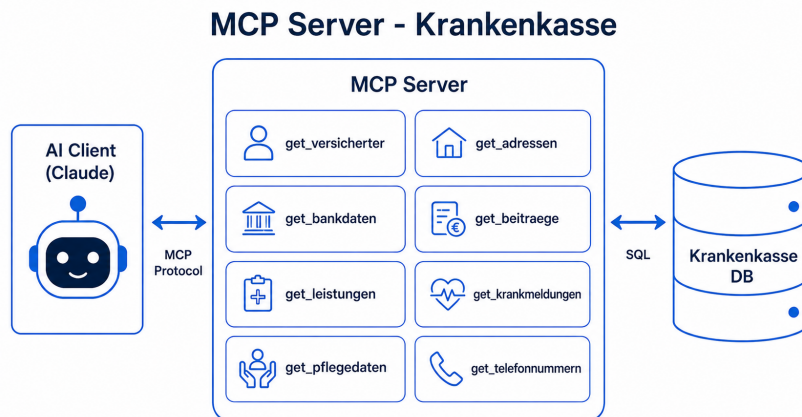


4 Aktueller technischer Stand (Test-Szenario)

Aktuell existiert ein MCP-Server als Testaufbau. Er verbindet sich über einen Datenbankbenutzer mit ausschließlich lesenden Rechten (Read-only) mit der Datenbank. Schreibende Operationen sind damit bereits auf Berechtigungsebene ausgeschlossen.

Die Zugriffe erfolgen nicht direkt auf produktive Tabellen, sondern ausschließlich über freigegebene Datenbank-Views. Über die Definition dieser Views wird gesteuert, welche Spalten und Datensätze überhaupt sichtbar sind; sensible Felder lassen sich so ausblenden oder maskieren. Auf diese Weise bleiben Datenbankstruktur, Datenschutz und Zugriffsrechte kontrollierbar, ohne dass der Agent die darunterliegende Tabellenstruktur kennt.

Die vom MCP-Server bereitgestellten Funktionen sind bewusst eng zugeschnitten. Die Adressabfrage erwartet beispielsweise genau eine konkrete KVNR als Parameter; sie gibt nur die zu dieser einen Versichertennummer gehörenden Adressdaten zurück.



Dadurch ist eine Massenabfrage wie:

„Gib mir alle Adressen zurück.“

technisch nicht vorgesehen.

Stattdessen sind nur gezielte Einzelabfragen möglich, zum Beispiel:

„Lade alle Adressen zur KVNR B000003310.“

5 Beispiel für eine erweiterte KI-Abfrage

Durch die Kombination aus KI-Agent und MCP-Server können Unternehmensdaten mit weiteren Informationen verknüpft und fachlich bewertet werden.

Beispiel:

Lade alle Adressen zur KVNR **B000003310**. Prüfe für jede Postleitzahl, ob dort am gestrigen Tag starke Windereignisse aufgetreten sind, und gib das Ergebnis je Adresse kurz aus.

6 Beispielergebnis

Für die KVNR **B000003310** wurde die versicherte Person **Karin Bauer** gefunden. Zur Person ist eine Hauptadresse hinterlegt: **Lindenallee 142, 45879 Gelsenkirchen**. Diese Adresse ist seit dem **24.10.2023** gültig.

Für die Postleitzahl **45879** in **Gelsenkirchen, NRW** wurde geprüft, ob am **13.05.2026** starke Windereignisse aufgetreten sind. Die Prüfung ergab Hinweise auf Windböen in der Region.

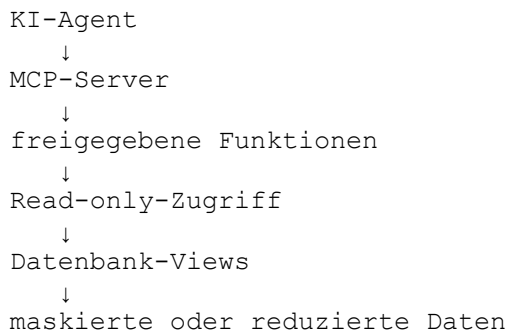
Bewertung:

- Am 13.05.2026 gab es Hinweise auf stärkere Windereignisse in der Region.
- Die Adresse liegt in einem potenziell betroffenen Gebiet.
- Das Ergebnis wurde je Adresse bewertet.

7 Fachliche Einordnung

Das Beispiel zeigt, dass ein KI-Agent nicht auf das reine Beantworten von Datenbankabfragen beschränkt ist. Er kann die über den MCP-Server abgerufenen Unternehmensdaten mit externen oder fachlichen Informationen verknüpfen und das Ergebnis aufbereiten.

Wichtig ist dabei, dass der Zugriff technisch begrenzt bleibt:



Jede Stufe dieser Kette schränkt den Zugriff weiter ein. Dadurch entsteht keine freie Datenbankschnittstelle für die KI, sondern eine kontrollierte Zugriffsschicht, deren Grenzen durch den MCP-Server und die Datenbankberechtigungen definiert sind.

8 Fazit

Der aktuelle Prototyp zeigt, dass sich ein KI-Agent über einen MCP-Server kontrolliert an Unternehmensdaten anbinden lässt. Der Agent kann ausschließlich die definierten MCP-Funktionen aufrufen und die zurückgegebenen Ergebnisse fachlich weiterverarbeiten.

Gleichzeitig bleiben wichtige Schutzmechanismen erhalten:

- kein direkter Tabellenzugriff
- nur Read-only-Verbindung
- Zugriff ausschließlich über Views
- Maskierung sensibler Felder
- keine Massenabfragen
- nur definierte MCP-Funktionen
- Abfragen nur über konkrete KVNR

Damit lässt sich ein praxisnahes Szenario abbilden, bei dem KI Unternehmensdaten unterstützt verarbeitet, ohne der KI unbegrenzten Zugriff auf die Datenbank zu geben.

Ist keine direkte Datenbankanbindung vorgesehen, kann der MCP-Server stattdessen an bestehende Backend-Schnittstellen angebunden werden. Der KI-Agent spricht dann keine Datenbank an, sondern ruft MCP-Funktionen auf, die intern freigegebene Backend-Endpunkte (etwa REST-APIs) nutzen. Dieser Aufbau ist für den produktiven Einsatz in der Regel vorzuziehen, da die im Backend bereits vorhandenen Sicherheitsmechanismen – Authentifizierung, Rollenprüfung, Fachlogik, Logging und Maskierung – erhalten bleiben und

wiederverwendet werden. Der MCP-Server wird in diesem Fall zu einer kontrollierten Brücke zwischen KI-Agent und bestehendem Backend.

```
> claude mcp list
claude.ai Google Drive: https://drivemcp.googleapis.com/mcp/v1 - ! Needs authentication
krankenkasse: /Users/drapatz/GIT-Home/XCode-Developer-Toolbox-Light/krankenkasse/mcp/.build/arm64-apple-macosx/release/KrankenKasseMCP - ✓ Connected
xcode-toolbox: /Users/drapatz/GIT-Home/XCode-Developer-Toolbox-Light/mcpserver/.build/release/XcodeMCPServer - ✓ Connected
```

```
Sources:
- Wilder Wetterwechsel im Mai: Erst fröhlich, dann nass, stürmisch und kalt (https://www.wetterprognose-wettervorhersage.de/wetter/mai/wetter-mai-2026/13553-wilder-
- Wetterwarnung Gelsenkirchen - DWD warnt vor Windböen (https://www.news.de/amp/lokales/
- Unwetterwarnungen für Gelsenkirchen (http://www.unwetterzentrale.de/wwz/getwarning.de)

* Worked for 30s
Lade alle Adressen zur KVNR B000003310. Prüfe für jede Postleitzahl, ob dort am gestrige
* Die Adressdaten für B000003310 liegen bereits aus der vorherigen Abfrage vor - ich verme
Ergebnis:
---
B000003310 - Bauer, Karzin
---


| # | Adresstyp    | Anschrift       | PLZ   | Ort           | Gültig             | Wi |
|---|--------------|-----------------|-------|---------------|--------------------|----|
| 1 | Hauptadresse | Lindenallee 142 | 45879 | Gelsenkirchen | 24.10.2023 - heute | A  |


---
Bewertung
PLZ 45879 (Gelsenkirchen, NRW)
-> Tiefdrucktrog über Deutschland am 13.05.2026
-> Südwestliche Strömung, stürmische Böen
-> DWD-Windwarnung für die Region aktiv
-> Ergebnis: BETROFFEN
Nur eine Adresse vorhanden - kein weiterer Vergleich möglich.
---
Hinweis: Hätte die Person historische Adressen (z.B. frühere Wohnorte), würde dieselbe P
ist oder nicht. Soll die Prüfung auf aktuelle Adressen beschränkt bleiben, muss das als
* Brewed for 10s
recap: Wir bauen ein KI-gestütztes System zur Sachbearbeiter-Unterstützung einer Kranken
Datenbankzugriffs gegen unkontrollierte Massenabfragen an. (disable recaps in /config)
> Wie kann man das Ergebnis automatisch im Sachbearbeiterfenster anzeigen?
abbruch befehl
```

KrankenKasse

Verbinden Verbindung

Suche: K*

Bauer, Karzin B000003310

Zur Trefferliste

Personen 1 Adressen 1 Telefon 1 Bank 1 Beschäftigung 1 Leistungen 0

KVNR	Nachname	Vorname	Geburtsdatum	Geschle...	Ver
B000003310	Bauer	Karzin	---	W	pfli

MySQL - Dienst starten

MySQL läuft bereits - kein Neustart notwendig.
Version: mysql Ver 9.6.0 for macos26.3 on arm64 (Homebrew)

Database Explorer

- krankenkasse@127.0.0.1
 - information_schema
 - views 84
 - krankenkasse
 - tables 13
 - adresse
 - arbeitgeber
 - bankverbindung
 - beitrag
 - beitragsart
 - beschaeftigung
 - krankmeldung
 - leistung
 - leistungsart
 - person
 - pflge
 - pflgegrad
 - telefonnummer
 - views 16
 - mcp_adresse_view
 - mcp_bankverbindung_view
 - mcp_beitrag_view

Run

Database > krankenkasse@127.0.0.1 > krankenkasse > tables > bankverbindung

bankverbindung

WHERE

ORDER BY

Data Sources and Drivers

Name: krankenkasse@127.0.0.1

Driver: MySQL supports since 5.2

Connection type: default Unix Socket URL only

Host: 127.0.0.1 Port: 3306

Authentication: User & Password

User: kasse

Password: <hidden> Save: Forever

Database: krankenkasse

URL: jdbc:mysql://127.0.0.1:3306/krankenkasse

Test Connection MySQL 9.6.0

Cancel Apply OK

KrankenKasseMCP

Downloads

Ordner	Ordner	Ordner	Ordner
.claude	.build	KrankenKasseMCP	Infrastructure
.idea	.claude		Models
images	images		Repositories
macos	Sources		Services
mcp			Tools
Dokumente	Dokumente		Entwickler
DATENBANK.md	DOKUMENTATION.md		main.swift