



Datenabfragen ohne SQL

Datenabfragen ohne SQL – Frag deine Daten wie einen Kollegen



Case Study: Datenabfragen ohne SQL – Frag deine Daten wie einen Kollegen

KPI-Box

- **Time-to-Insight:** von „Ticket & warten“ auf **Sekunden** (typisch < 30–90 s pro Anfrage)
 - **IT-Entlastung:** **–75%** weniger Ad-hoc-Reporting-Tickets (Pilot)
 - **Self-Service-Quote:** **70%+** der Standardabfragen direkt durch Fachbereiche
 - **Produktivitätsgewinn:** ca. **27 Std./Woche** weniger Such- & Abstimmungsaufwand (≈ 20 User)
 - **ROI: Amortisation in ~4–6 Monaten**, Zielbild **< 12 Monate** (konservativ gerechnet)
-

Über den Kunden

Ein mittelständisches Unternehmen (anonymisiert) mit mehreren Fachbereichen (Vertrieb, Projektmanagement, Produktion/Service). Daten liegen strukturiert in **SQL-Datenbanken** (z. B. ERP-/Projekt- und operative Trackingdaten wie Barcode-/Teilehistorien), sind aber im Alltag nur über Spezialisten abrufbar.

1) Ausgangssituation

„Kannst du mir mal schnell eine Auswertung ziehen?“ – dieser Satz fiel im Unternehmen mehrfach pro Woche. Vor dem Kundentermin brauchte der Vertrieb aktuelle Zahlen, das Management ein Projekt-Update, die Fachabteilung eine spontane Analyse.



Die Daten waren vorhanden – aber der Zugriff war indirekt:

- Fachbereich formuliert Bedarf (oft unpräzise)
- Ticket an IT/BI
- Rückfragen, Wartezeit, manuelle SQL-Abfrage
- Export/CSV, Interpretation, nächste Rückfrage

Besonders typisch: operative Auswertungen wie eine **Barcode-Historie** mit genau definierten Spalten und Filterkriterien (Zeitraum, Kundennummer, Träger, Artikelinformationen, Ein-/Ausgänge, Waschzyklen).

Konsequenz: Zeitverlust, Medienbrüche – und ein Reporting, das „zu spät“ kommt, um Entscheidungen wirklich zu unterstützen.

2) Herausforderung

Die Idee „Einfach ChatGPT auf die Datenbank lassen“ scheitert in der Praxis an vier Punkten:

1. **Sicherheit & Rechte:** Wer darf welche Tabellen/Spalten sehen? (Rollen, PII, Mandanten)
2. **Verlässlichkeit:** SQL muss korrekt sein – keine Halluzinationen, keine falschen Joins
3. **Governance:** Nachvollziehbarkeit, Logging, Audit-Trail, Freigaben
4. **Praxisfit:** Fachbereich fragt in Alltagssprache – Datenmodell spricht Tabellenlogik

Das Ziel war daher nicht „nice to have“, sondern: **Self-Service ohne Kontrollverlust.**

3) Lösung

„SQL-Abfragen ohne SQL“ als kontrollierter Daten-Assistent



Statt klassischer Dashboards (vordefiniert, langsam, oft unflexibel) entstand ein Assistent, der **natürliche Sprache** in **sichere, geprüfte SQL-Abfragen** übersetzt – passend zum Nutzerkontext.

Kernprinzip: *Frag deine Daten wie einen Kollegen – aber mit Leitplanken.*

So funktioniert's (Blick unter die Haube, ohne Nerd-Sprache)

- **Schema- & Glossar-Kontext:** Der Assistent kennt zugelassene Tabellen/Views und Begriffe (z. B. „Träger“, „Garment ID“, „Waschzyklen“).
- **SQL-Generierung:** Das Modell erzeugt SQL aus der Frage – ähnlich wie es IT heute manuell schreibt.
- **Guardrails / Query-Checks:**
 - nur **Read-only**
 - nur **freigegebene Views** (z. B. „Reporting_BarcodeHistory“)
 - Blocklisten (DELETE/UPDATE, Cross-Tenant, PII-Spalten)
 - Limitierung (Row-Limits), Timeouts
- **Rollen & Rechte:** Zugriff via DB-Rollen/SSO, Ergebnis wird automatisch gefiltert/maskiert
- **Transparenz:** SQL wird optional angezeigt + Logging für Audit/Fehleranalyse

Live-Demo-Use-Case (typisch aus dem Alltag)

Beispielhafte Anfrage aus dem Fachbereich:

„Ich brauche eine Auswertung der Barcode-Historie vom 01.01.2024–04.08.2024 für Kundennr. 350901 und die Träger 164/208/139/230/249 mit Spalten ...“

Früher schrieb die IT dafür ein Join-Statement über Historie, Teilestamm und Trägerdaten, inkl. Filter auf Kundennummer, Trägerliste und Datumsbereich.

Heute erledigt der Assistent das in Sekunden – inklusive sauberer Spaltenstruktur (CSV/Excel-Export auf Knopfdruck).

Weitere typische Fragen, die sofort Self-Service-fähig wurden:

- Lieferscheine & Lieferdatum zu einem bestimmten Teil / einer Garment-ID



- Kundenliste mit bestimmten Artikelgruppen (z. B. Eigenwäsche-Artikel)

Der Durchbruch

Der Wendepunkt kam nicht im IT-Meeting, sondern im Tagesgeschäft:

Kurz vor einem Kundentermin fehlte dem Vertrieb ein aktueller Stand („Welche Teile sind aktuell draußen? Welche wurden zuletzt eingelesen?“).

Früher: Ticket, warten, improvisieren.

Diesmal: Frage in natürlicher Sprache – **Antwort in Sekunden**. Das Meeting startete mit Fakten statt Bauchgefühl.

Ab da war klar: **Das ist kein Reporting-Tool. Das ist ein neuer Arbeitsmodus.**

4) Ergebnis

Im Pilot (konservativ gemessen) ergaben sich drei sichtbare Effekte:

1. **Tempo:** Ad-hoc-Auswertungen sind sofort verfügbar (Sekunden statt Tage)
2. **Entlastung:** IT/BI wird von Routineabfragen befreit, fokussiert sich auf Datenqualität & strategische Themen
3. **Befähigung:** Fachbereiche arbeiten datengetriebener – ohne SQL-Schulung, ohne Tool-Zirkus

Beispielhafte Pilot-KPIs (realistisch & konservativ):

- **-75%** weniger Ad-hoc-Tickets für Datenabfragen (Standardfragen laufen Self-Service)
- **~27 Std./Woche** Zeitgewinn durch weniger Abstimmung/Warten (20 Nutzer, ~10 Abfragen/Woche, Ø 8 Minuten Ersparnis je Abfrage)
- **Bessere Datenklarheit:** weniger Missverständnisse durch einheitliche Spalten-/Begriffsdefinition (Glossar + Views)



5) ROI

Kostenannahme (Pilot → Rollout):

- Setup/Integration/Guardrails/Views: **~25.000 € einmalig**
- Betrieb/Lizenzen/Monitoring: **~1.500 €/Monat** (= 18.000 €/Jahr)

Nutzen (konservativ):

- Zeitgewinn Fachbereiche: ~27 Std./Woche × 65 €/Std. ≈ **~91.000 €/Jahr**
- IT-Entlastung: 45 Std./Monat × 80 €/Std. ≈ **~43.000 €/Jahr**
- **Gesamtnutzen: ≈ 134.000 €/Jahr**

➔ **Payback:** 25.000 € / (134.000 € / 12) ≈ **2–3 Monate** (ohne Betrieb)

➔ **Mit Betrieb konservativ: ~4–6 Monate** realistisch, Zielbild < **12 Monate**.

Visualisierungshinweise für Design

- **Process-Flow „Vorher/Nachher“:** Ticket → Warten → Export **vs.** Frage → Guardrails → Ergebnis
- **KPI-Infografik:** Time-to-Insight, Ticketreduktion, Stundenersparnis, ROI
- **Architektur-Skizze:** Fachbereich → NLP → SQL-Generator → Validator → Read-only Views → DB → Audit-Log
- **Zeitstrahl:** Pilot (2–4 Wochen) → Rollout (6–10 Wochen) → Skalierung (Glossar/Views/Use-Cases)

Call-to-Action

Du willst sehen, wie das live auf einer echten SQL-Datenbank funktioniert – inklusive Datenschutz, Rollen & Rechte?



Webinar „KI zum Mittag – Teil 6: SQL-Datenabfragen neu gedacht“

🎤 Referent: Florian Auer

🖥️ Online via Microsoft Teams

📅 Mittwoch, 10.12.2025 | ⌚ 12:00–13:00 Uhr | 🆓 kostenfrei (Anmeldung erforderlich)

Anmeldung:

www.berchtesgadener-land.de/ki-zum-mittag-teil-6-sql-datenabfragen-neu-gedacht
<https://lnkd.in/dGvgUmhk>

Teaser / Social Preview

SQL-Abfragen ohne SQL – frag deine Daten wie einen Kollegen.

In vielen Unternehmen kosten „mal schnell eine Auswertung“ jede Woche Stunden: Ticket an die IT, Rückfragen, warten. In unserem Webinar zeigen wir live, wie Fachbereiche SQL-Datenbanken in natürlicher Sprache abfragen – sicher, rollenbasiert und in Sekunden. Ergebnis: weniger Tickets, schnellere Entscheidungen, ROI in wenigen Monaten.