

idesis.datenarchiv.01

Fallstudie

Aufbau eines Datenarchivs

Projekt Aufbau eines Datenarchivs
Dokument Fallstudie
idesis.datenarchiv.01
Datum 2026-03-13
Kunde idesis GmbH
Empfänger Softwareentwicklung
Max-Keith-Straße 66 (E11)
45136 Essen
Autor Jörg Walter, et. al.
E-Mail vertrieb@idesis.de

© Copyright 2026 – Alle Inhalte, insbesondere Texte, sind urheberrechtlich geschützt. Alle Rechte, einschließlich der Vervielfältigung, Veröffentlichung, Bearbeitung und Übersetzung, bleiben vorbehalten, idesis GmbH.

Inhaltsverzeichnis

1 Migration von Oracle/Db2 auf PostgreSQL zum Aufbau eines Datenarchivs bei einer mittelständischen Finanzinstitution: Vier unterschätzte Erfolgsfaktoren aus der Praxis	4
1.1 Ausgangssituation & Herausforderung	4
1.2 Besondere Anforderungen & Risiken	5
1.3 Vorgehensweise & Lösung	9
1.4 Ergebnisse	10

1 Migration von Oracle/Db2 auf PostgreSQL zum Aufbau eines Datenarchivs bei einer mittelständischen Finanzinstitution: Vier unterschätzte Erfolgsfaktoren aus der Praxis

1.1 Ausgangssituation & Herausforderung

Unser Auftraggeber, ein mittelständisches deutsches Finanzinstitut, modernisierte sein Kernsystem. Die über Jahrzehnte gewachsenen Datenbestände der zentralen Db2- und benachbarter Oracle-Systeme konnten in ihrer ursprünglichen Form nicht in das neue System übernommen werden. Damit stellte sich eine grundsätzliche Frage: Die Altsysteme über die regulatorisch erforderlichen Aufbewahrungsfristen weiterzubetreiben — oder die Datenbestände in ein zentrales Archiv zu überführen. Der Auftraggeber entschied sich für den Aufbau eines Datenarchivs auf Basis von PostgreSQL, um langfristig Lizenz- und Betriebskosten zu senken und die Datenlandschaft zu konsolidieren.

Die Ausgangslage war dabei typisch für Unternehmen dieser Art: rund 10 Schemata mit tausenden von Tabellen, historisch gewachsen, heterogen dokumentiert und mit Abhängigkeiten, die sich erst im Projektverlauf vollstän-

dig erschlossen. Verschärft wurde die Situation durch feste Abschalttermine für die Altsysteme — der Zeitdruck war real und nicht verhandelbar.

Von Anfang an bekannt und regulatorisch nicht verhandelbar war zudem die DSGVO-Dimension: Als Finanzinstitut unterliegt der Auftraggeber strengen Vorgaben zur Datenaufbewahrung — aber ebenso zur nachweisbaren Löschung personenbezogener Daten. Das Archiv musste also nicht nur Daten zuverlässig speichern, sondern auch deren gezielten Zugriff und Löschung reversionssicher ermöglichen.

1.2 Besondere Anforderungen & Risiken

Auf den ersten Blick erscheint eine Datenmigration als technisch überschaubare Aufgabe: Daten lesen, transformieren, schreiben, fertig. Die Realität dieses Projekts war eine andere — und die Herausforderungen lagen oft dort, wo man sie zunächst nicht vermutet hätte.

Metadaten-Transformation: Maßgeschneidert statt Standard-Tool

Eine zentrale Anforderung war, die Originaldaten möglichst unverändert zu übernehmen — also keine inhaltliche Transformation, sondern eine strukturtreue Überführung. Was zunächst nach einem klar umrissenen Problem klingt, erwies sich in der Praxis als erheblich komplexer.

Db2, Oracle und PostgreSQL unterscheiden sich in ihren DDL-Dialekten teils grundlegend: in der Behandlung von Datentypen, Constraints, Zeichenkodierungen, Defaultwerten und weiteren Schemadetails. Eine direkte 1:1-Übernahme der Metadaten war daher nicht möglich — jede Differenz musste analysiert, bewertet und in eine PostgreSQL-konforme Entsprechung überführt werden, ohne dabei die inhaltliche Integrität der Daten zu gefährden.

Der Auftraggeber hatte [marktgängige Migrationswerkzeuge](#) im Vorfeld evaluiert und deren Einsatz ausgeschlossen: Die Kontrolle über technische Details wäre verloren gegangen, und der Koordinationsaufwand zwischen

Werkzeug-Verhalten, eigenen Anforderungen und der Komplexität der Ausgangssysteme erschien größer als der Nutzen. Die Entscheidung fiel daher bewusst für einen maßgeschneiderten Transformationsprozess, den wir entwickeln und vollständig kontrollieren konnten.

Laufzeit: Warum der naive Ansatz bei Terabyte-Datenmengen scheitert

Bei der Planung einer Datenmigration liegt ein bestimmtes Vorgehen nahe: Daten aus der Quelle lesen, transformieren, ins Ziel schreiben — Zeile für Zeile, Tabelle für Tabelle. Für überschaubare Datenmengen funktioniert das. Bei Terabyte-Datenmengen führt dieser Ansatz zu inakzeptablen Laufzeiten, die einen fristgerechten Abschluss vor den Abschaltterminen der Altsysteme gefährden.

Der Engpass liegt dabei weniger beim Lesen als beim Schreiben: Konventionelle INSERT-basierte Prozesse erzeugen erheblichen Overhead durch Constraint-Prüfungen, Index-Aktualisierungen und Transaktionslogging — bei jeder einzelnen Zeile.

Die Lösung, die sich in diesem Projekt bewährt hat, kombiniert vier Maßnahmen: erstens den Einsatz von CSV als Transferformat, das von Db2, Oracle und PostgreSQL nativ unterstützt wird und deutlich effizientere Bulk-Operationen erlaubt; zweitens den Einsatz der datenbanknativen Bulk-Import-Mechanismen im Ziel statt zeilenweiser INSERTs; drittens die verzögerte Erstellung von Constraints und Indizes — also erst nach dem Datenimport, nicht während; und viertens die Parallelisierung der drei Teilschritte Extraktion, Transfer und Import. Interessanterweise bilden diese Schritte eine klassische ETL-Pipeline — und wie bei jeder Pipeline lassen sie sich parallel ausführen, insbesondere wenn große Tabellen frühzeitig in kleinere Verarbeitungseinheiten zerlegt werden. Diese Kombination reduzierte die Laufzeiten auf ein projektaugliches Maß.

Diese Optimierungen sind nicht neu — aber sie sind in der Planungsphase vergleichbarer Projekte nach unserer Erfahrung systematisch unterschätzt.

Wer Laufzeiten früh realistisch einschätzen will, sollte Lasttests mit repräsentativen Datenmengen zu einem frühen Projektzeitpunkt einplanen.

Die schiere Datenmenge in Kombination mit den festen Abschaltterminen machte die Migrationsgeschwindigkeit zu einem kritischen Parameter. Erste Schätzungen zur Laufzeit erwiesen sich als zu optimistisch — die tatsächlichen Transferzeiten erforderten eine sorgfältige Planung der Migrationsfenster und eine iterative Optimierung des Prozesses. Dieser Aspekt ist in Planungen vergleichbarer Projekte häufig unterschätzt.

Validierung als Kernaufgabe, nicht als Nachgedanke

Validierung: vom Nachgedanken zur Kernaufgabe Zu Projektbeginn war die Validierung der migrierten Daten wenig konkretisiert — im Wesentlichen waren einfache Zählprüfungen vorgesehen. Die inhaltliche Abnahme sollte durch die Fachabteilungen des Auftraggebers erfolgen, die die fachliche Kenntnis der Daten mitbringen.

Im Projektverlauf änderte sich diese Ausgangslage grundlegend: Per Change Request wurde eine möglichst vollständige inhaltliche Validierung aller migrierten Daten gefordert — also nicht nur die Prüfung, ob die richtige Anzahl Zeilen angekommen ist, sondern ob die Inhalte korrekt und vollständig übertragen wurden. Bei tausenden von Tabellen und Terabyte-Datenmengen ist das eine erhebliche Aufgabe.

Für die Umsetzung kam [Google Data Validation Tool \(DVT\)](#) zum Einsatz — ein Open-Source-Werkzeug, das genau für diesen Anwendungsfall entwickelt wurde und Vergleiche zwischen Quell- und Zielsystem auf Spalten- und Zeilenebene ermöglicht. DVT erwies sich als geeignetes Instrument, stieß jedoch bei bestimmten Konstellationen — etwa komplexen Datentypen oder sehr großen Tabellen — an Grenzen, die zusätzlichen manuellen Aufwand erforderten.

Die zentrale Lehre aus diesem Projektverlauf: Validierungsanforderungen sollten in vergleichbaren Projekten frühzeitig und verbindlich zwischen allen

Beteiligten — Auftragnehmer, IT und Fachabteilungen des Auftraggebers — geklärt werden. Was als Abnahmekriterium gilt, bestimmt maßgeblich den Projektaufwand. Ein nachträglicher Scope-Wechsel in diesem Bereich ist kostspielig.

DSGVO: Auskunft und Löschung als technische Anforderung

Die DSGVO verpflichtet Finanzinstitute nicht nur zur sicheren Aufbewahrung personenbezogener Daten, sondern auch zu deren nachweisbarer Auskunft und Löschung auf Anfrage. Diese Anforderung war von Projektbeginn an bekannt — und eine frühe Voruntersuchung zeigte, dass ihre technische Umsetzung besondere Aufmerksamkeit erfordern würde.

Der naheliegende Ansatz — einem Einsprungspunkt folgen und über Primär- und Fremdschlüsselbeziehungen alle zugehörigen personenbezogenen Daten zu identifizieren — setzt voraus, dass diese Beziehungen auch deklariert sind. In historisch gewachsenen Systemen ist das häufig nicht der Fall: Referenzielle Integrität wurde oft in der Anwendungslogik implementiert, nicht als Constraint in der Datenbank. Die Voruntersuchung bestätigte dieses bekannte Muster auch hier.

Eine automatische Rekonstruktion der impliziten Beziehungen ist ein bekanntes Problem der Datenprofilierung. Ansätze auf Basis von Inclusion Dependencies (IND) — zur Erkennung potenzieller Fremdschlüsselbeziehungen — und Unique Column Combinations (UCC) — zur Identifikation natürlicher Schlüssel — sind in der [Forschung](#) gut beschrieben. In der Praxis scheitert ihre Anwendung auf großen, historisch gewachsenen Schemata jedoch regelmäßig: Die Anzahl der Kandidaten wächst kombinatorisch, die Fehlerrate ist hoch, und die manuelle Nachvalidierung der Ergebnisse übersteigt den Aufwand einer direkten fachlichen Klärung. Dieser Weg wurde daher frühzeitig als nicht praktikabel eingestuft.

Die gewählte Lösung war pragmatisch und zielgerichtet: Der Auftraggeber definierte den für die DSGVO-Anforderungen relevanten Beziehungsgraphen selbst — eingeschränkt auf Kunden- und Kontodaten. Damit lag die fachliche

Verantwortung dort, wo sie hingehört: beim Auftraggeber, der die Bedeutung seiner Daten kennt. Die technische Umsetzung konnte auf dieser Grundlage sauber und nachvollziehbar erfolgen.

1.3 Vorgehensweise & Lösung

Das Projekt war in zwei klar getrennte Abschnitte gegliedert: einer initialen Voruntersuchung folgte das eigentliche Migrationsprojekt. Diese Trennung erwies sich als wertvoll — die Voruntersuchung legte die Grundlage für eine realistische Projektplanung und deckte frühzeitig kritische Punkte auf, darunter die fehlenden Fremdschlüsselbeziehungen und die Anforderungen an die DSGVO-Umsetzung.

Iterative Migration quellenweise

Die Migration selbst folgte keinem “Big Bang”-Ansatz, sondern wurde iterativ und quellenweise durchgeführt. Den Schwerpunkt bildete das Db2-System mit rund 80% der Datenbestände, gefolgt vom Oracle-System mit den verbleibenden 20%. Dieser Ansatz erlaubte es, Erfahrungen und Optimierungen aus der Db2-Migration direkt in die Oracle-Migration einfließen zu lassen — insbesondere bei der Metadaten-Transformation und der Laufzeitoptimierung des Transferprozesses.

Die Migrationspipeline wurde in Python entwickelt und umfasste die Extraktion aus den Quellsystemen, die Transformation der Metadaten und den Bulk-Import in PostgreSQL über CSV als Transferformat mit verzögerter Constraint- und Index-Erstellung. Eine formale Orchestrierungsschicht wurde bewusst nicht eingesetzt — der Prozess war skriptbasiert und damit transparent, wartbar und gut nachvollziehbar.

Metadatengetriebene Architektur als Leitprinzip

Ein durchgehendes Gestaltungsprinzip war der metadatengetriebene Ansatz: Statt für jede Tabelle individuelle Migrationskripte zu entwickeln, wurden

die Schemainformationen systematisch erfasst und als Steuerungsgrundlage für den gesamten Prozess genutzt. Dieses Prinzip zog sich von der Migration bis in den späteren Betrieb durch.

Datenbankviewer und DSGVO-Werkzeuge

Parallel zur Migration entstand eine webbasierte Applikation zum Betrieb des Archivs — ein metadatengetriebener Datenbankbrowser, der auf Vorentwicklungen und -erfahrungen aufbaute. Die Anwendung ermöglicht den strukturierten Lesezugriff auf das Archiv ohne direkten Datenbankzugriff — ein wesentlicher Aspekt für den sicheren und kontrollierten Betrieb.

Die DSGVO-Funktionalität — Auskunft und Löschung personenbezogener Daten — wurde als eigenständiges .NET-CLI-Werkzeug realisiert, ebenfalls metadatengetrieben. Grundlage war der vom Auftraggeber definierte Beziehungsgraph auf Kunden- und Kontodaten. Die Löschung folgt diesem Graph deterministisch und nachvollziehbar — eine wichtige Voraussetzung für die revisionssichere Dokumentation gegenüber Aufsichtsbehörden.

Projektsteuerung

Das Projekt wurde zu einem Festpreis und weitgehend eigenverantwortlich durchgeführt. Ein wöchentlicher Jour fixe mit dem Auftraggeber stellte die kontinuierliche Abstimmung sicher, ohne den Arbeitsfluss zu unterbrechen. Änderungen im Scope — wie die erweiterten Validierungsanforderungen — wurden als Change Requests formal behandelt.

1.4 Ergebnisse

Das Projekt wurde in seinem Kernziel vollständig erreicht: Die Datenbestände von Db2 und Oracle wurden erfolgreich in ein zentrales PostgreSQL-basiertes Datenarchiv überführt und bilden damit die Grundlage für die Abschaltung der Altsysteme.

Umfang und Laufzeit

Migriert wurden mehrere tausend Tabellen — sowohl strukturell als auch inhaltlich — mit einem Gesamtdatenvolumen im Terabyte-Bereich. Die Projektlaufzeit betrug 9 Monate, was angesichts der Komplexität der Ausgangssysteme und der festen Abschalttermine einen straffen Zeitplan darstellte.

Der Abschalttermin für das Db2-System wurde eingehalten. Für das Oracle-System war nach unserem Kenntnisstand eine Verschiebung von einem Monat erforderlich — eine angesichts der Projektgröße und der Komplexität der Metadaten-Transformation vertretbare Abweichung.

Validierung: Aufwand, der sich gelohnt hat

Alle migrierten Tabellen wurden einer vollständigen inhaltlichen Validierung unterzogen — ein erheblicher, aber letztlich unverzichtbarer Aufwand. Die Validierung deckte reale Probleme auf, die sonst unbemerkt geblieben wären: Insbesondere Zeichenkodierungen erwiesen sich als fehleranfällig, da die Quellsysteme unterschiedliche Kodierungen verwendeten — EBCDIC auf dem Mainframe, CP1252 und UTF-8 in den weiteren Systemen. Konvertierungsfehler in diesem Bereich sind schwer zu erkennen, da betroffene Daten auf den ersten Blick plausibel wirken können.

Eine unerwartete Validierungsquelle waren zudem die Audit-Logs des Zielsystems: Da Finanzinstitute auch Lesezugriffe auditieren, fanden sich dort erwartete Abweichungen, die als zusätzlicher Indikator für Migrationsprobleme genutzt werden konnten — ein Nebeneffekt der regulatorischen Anforderungen, der sich als nützlich erwies.

Betrieb: Datenbankbrowser und DSGVO-Werkzeuge

Der metadatengetriebene Datenbankbrowser und die DSGVO-Werkzeuge für Auskunft und Löschung befinden sich aktuell in der Abnahme. Die Zielgruppe sind Spezialisten — ein kleiner, aber kritischer Nutzerkreis, der gezielte Zugriffe auf das Archiv benötigt. Parallel löst der Auftraggeber Abfragefunktionen des abgelösten Kernsystems schrittweise ab.

Das übergeordnete Ergebnis

Der vielleicht bedeutsamste Effekt des Projekts lässt sich mit einem Satz zusammenfassen: Der Auftraggeber konnte den Betrieb seines z/OS-Rechenzentrums zurückbauen. Was das in Lizenz-, Wartungs- und Betriebskosten bedeutet, lässt sich ohne Insiderwissen erahnen — und bedarf keiner weiteren Erläuterung.