

Lageranwendung LVS mit SQL – Stationär und Mobil

Heute stelle ich euch eine kleine App vor, mit der ihr online Artikelbewegungen und Bestände buchen und verwalten könnt.

Die Tabellen verwaltet der SQL Server.

Basis ist unsere Lagerverwaltung AE WWSLite. Diese recht umfangreiche Software habt ihr auf irgendeinem Windows PC installiert und sie bringt eine Datenbank mit, die zahlreiche Felder zur freien Verwendung aufweist. So könnt ihr Stammdaten mit beliebigem Inhalt hinterlegen und alles Wesentliche zu einem Produkt speichern.

<https://www.terminal-systems.de/wp/lager1/lagerverwaltung-aktuell/>

In der SQL Version von AE WWSLite wird diese Datenbank auf einem SQL Server gespeichert. In unserem heutigen Fall nehmen wir dazu den SQL Server von Microsoft SQL Server, oder, wenn es kostenfrei sein soll, den Microsoft SQL Server Express.

Der SQL Server verwaltet dabei drei Tabellen. Die Artikeltabelle. Die Buchungshistorie in der alle Buchungen dokumentiert sind. Eine Benutzertabelle, die für den Betrieb notwendig ist.

Die Version auf dem Windows PC arbeitet daher aus technischer Sicht wie ein SQL Client. Alles, was der Benutzer eingibt, wird an den SQL Server übertragen, dieser liefert Antworten. Beispiel: Ein Benutzer führt für den Artikel 4711 eine Zugangsbuchung von 5 Stück durch. Im Detail geschieht folgendes:

Schritt 1: Der Benutzer wählt auf seinem PC Artikel 4711 aus.

Schritt 2: Der SQL Server sendet die Stammdaten für Artikel 4711 an den Benutzer PC.

Schritt 3: Der Benutzer PC zeigt die Stammdaten für 4711 an.

Schritt 4: Der Benutzer wählt ZUGANG aus und gibt die Menge 5 an.

Schritt 6: Der PC addiert die eingegebenen Menge zum aktuellen Bestand für 4711 und sendet den Datensatz (oder die geänderten Felder) zurück an den SQL Server.

Schritt 7: Der SQL Server speichert die neuen Werte für 4711. Jede neue Anfrage nach diesem Artikel erhält jetzt sofort die geänderten Werte!

In der Praxis geht es noch etwas weiter:

Schritt 8: Der PC erstellt einen Datensatz Historiendaten in der die Buchung für 4711 dokumentiert wird und sendet diesen an den SQL Server.

Schritt 9: Der SQL Server speichert den Datensatz Historiendaten in der Buchungstabelle HISTORIE. Wer jetzt die Buchungstabelle Historie anfordert, bekommt sofort die Buchungsinformation für diese Warenbewegung präsentiert.

Nach diesem Verfahren finden alle Buchungen auf dem oder den PCs der Benutzer statt. Dank SQL Server können verschiedene Benutzer gleichzeitig Buchungen durchführen. Der SQL Server sorgt dafür, dass sie alle in richtiger Reihenfolge durchgeführt und gespeichert werden.

Doch wie sieht es mit mobilen Buchungen aus?

Mobil Buchen im Batch Betrieb

Für die mobile Datenerfassung hat sich oftmals der Batch Betrieb bewährt. Batch bedeutet, die mobile Erfassung und die Datenbank sind voneinander getrennt. Im extremen Fall kann die mobile Erfassung im Hamburger Hafen stattfinden, der Server mit der Datenbank wird in München verwaltet. Der Benutzer gibt seine Buchungen direkt vor Ort in ein mobiles Gerät ein, ohne dass er just-in-time Zugriff auf seine Datenbank hat. Wenn er fertig ist, am Tages- oder Schichtende, werden alle Daten in den Computer eingelesen und mit der Datenbank synchronisiert.

Der Vorteil von Batch: Man muss nicht viel tun. Mobile Erfassung und Datenbank können physikalisch unabhängig voneinander betrieben werden. Erst beim Synchronisieren müssen die Daten irgendwie vom PC erreichbar sein. Entweder weil das mobile Gerät in der Zentrale vorliegt oder die mobilen Erfassungsdaten in der Cloud gespeichert wurden.

Batch Betrieb ist daher eine kostengünstige und nahezu jederzeit verfügbare Art und Weise, wie mobile Daten erfasst und gespeichert werden können. Unsere Lagerverwaltung AEWWSLite unterstützt mit dem Modul EXTERNE DATEN die Übernahme der Erfassungsdaten aus verschiedenen mobilen Geräten. Der Datentransfer kann dabei vollständig automatisiert ablaufen, je nachdem, wie die IT das Modul Externe Daten eingerichtet hat.

<https://www.terminal-systems.de/wp/lager1/lagerverwaltung-aktuell/externedaten-doku/>

Doch was ist, wenn ich nicht Batch buchen will – sondern direkt mobil in der Datenbank?

Mobil Buchen im Online Betrieb

Hierfür kommt dann die eingangs erwähnte App ins Spiel. Wir nennen Sie mal AE SQL ONLINE. Letztlich ist sie nichts anderes als ein mobile SQL Client – nur nicht für den Windows PC, sondern für ein Android Gerät. Zum Beispiel für ein Android Smartphone.

<https://www.terminal-systems.de/wp/lager1/mobileaewwsliteapp/>

Ihr könnt die App direkt von der Webseite von AE SYSTEME [www.terminal-systems](http://www.terminal-systems.de) downloaden. Im Google Store dürfte sie nicht zu finden sein. Die Vorgaben und Kriterien für den Google Store sind meistens recht aufwändig – daher veröffentlichen wir spezielle Apps lieber auf unserer Webseite und sparen und den manchmal enormen Zeit- und Kostenaufwand einer Google Play Store Veröffentlichung.

Doch die Installation von Apps nicht aus dem Google Store ist kein Teufelswerk. Wie das geht könnt ihr problemlos überall bei Google nachlesen – oder ich habe das hier in unserem Blog beschrieben:

<https://www.terminal-systems.de/wp/android-app-apk-datei-installieren/>

Sobald ihr die App auf einem Android Gerät installiert habt, könnt ihr sie starten. Was nun als Erstes zu geschehen hat, ist vermutlich leicht verständlich: ihr müsst die Zugangsdaten für Euren SQL Server in die App eintragen. Damit das Smartphone direkt mit dem SQL Server kommunizieren kann.

Das Ganze führt uns in das Menü SETTINGS.

In Settings werden die Zugangsdaten für den SQL Server eingetragen, man nennt es auch SQL Serverstring. Hierzu zählen z.B. IP Adresse unter der der SQL Server erreichbar ist, Instanzname und Name der Datenbank, die für AEWWSLite verwendet werden soll. (Hier müssen dann die Artikeltabellen, Buchungstabelle und Usertabelle enthalten sein.)

Ebenso wichtig sind die Zugangsdaten für den SQL Server, also Benutzername und Passwort mit der das mobile Android Gerät sich auf den SQL Server einloggen soll.

Sobald die Daten eingetragen und (nicht vergessen!!!) gespeichert sind, kann der Vorgang getestet werden. Im Eingabemenü gibt es den Button CHECK SQL. Hiermit wird eine direkte Verbindung zwischen Android Gerät und SQL Server aufgebaut. Wenn das möglich ist erscheint der Text SUCCESS! Wenn es nicht funktioniert, passiert erst mal länger gar nichts (weil das Gerät es einige Male versucht) und dann kommt eine Fehlermeldung.

Was häufig stört: Falls ihr einen Firewall auf Eurem SQL Server betreibt, müsst ihr den so konfigurieren, dass IP Abfragen von außen durchkommen. Sonst kann das arme Android Gerät natürlich niemals eine Antwort bekommen. In vielen Fällen ist der SQL Server unter Port 1433 erreichbar. Ihr müsst also eine Ausnahme für SQL hinterlegen, damit die Kommunikation läuft. Google Suche ist hier oftmals eine gute Lösung!

Wenn Ihr den Text SUCCESS bekommt, ist die Kommunikation möglich!

In der Praxis heißt das nun: wenn ihr eine Artikelnummer in das mobile Gerät eingibt oder mit einem Scanner (oder der Kamera) einen Barcode einscannet, dann wird diese an den SQL Server übertragen und der SQL Server liefert euch den Datensatz für die Buchung zurück.

Wenn ihr nun eine Menge eingibt und das Ganze bestätigt, werden die Bestandsdaten entsprechend geändert und die Werte auf dem SQL Server geupdated. Kurz: Ihr arbeitet nun immer online auf dem SQL Server.

Natürlich wird auch in der Buchungshistorie ein Datensatz mit den Buchungsdaten vermerkt, so dass ihr weiterhin eine genaue Dokumentation über alle Buchungen bekommt.

Platz für Erweiterungen

Allerdings seien auch die Nachteile dieser Lösung nicht verschwiegen:

Die Online App wird von uns als kostenfreies Toolkit angeboten. Das bedeutet auch, wir haben uns nicht die Mühen gemacht, die wir für Bezahlsoftware machen würden. Von irgend etwas müssen wir auch leben – also kosten Änderungen extra.

Zum einen: Die mobilen Geräte brauchen bei dem Tool einen stets funktionierenden Wireless / WLAN Zugang zum Netzwerk mit dem SQL Server. Schon Funklöcher oder andere WLAN Probleme würden dazu führen, dass die mobile Kommunikation mit dem SQL Server nicht möglich ist. Das mobile Gerät kann nicht mehr verwendet werden und ist eigentlich nur noch teurer Ballast. Wenn man Änderungen programmieren möchte, könnte man z.B. die Daten zwischenspeichern, wenn der SQL Server mal für eine Zeitspanne nicht zur Verfügung stehen würde.

Zum anderen: die kleine App führt weder Record Locking durch noch ist sie für Buchungen an gleichen Artikel optimiert. Wenn zwei Clients zur gleichen Zeit Buchungen für den gleichen Datensatz erfassen, kann es daher theoretisch zu Problemen kommen, weil der Datensatz nur einmal ausgetauscht wird: Nach Eingabe der Artikelnummer. Die Speicherung der geänderten Menge findet aber erst nach Eingabe einer Menge statt. Hier kann eine Zeitspanne X dazwischen liegen und in dieser Zeitspanne könnte ein anderer Benutzer bereits den Bestand verändern.

Im Fall des Windows PC als SQL Client ist das Ganze etwas umfangreicher gelöst. Hier wird ein Datensatz gelocked, wenn er bearbeitet wird und unmittelbar vor dem Speichern neuer Bestandswerte findet noch mal ein wiederholter SQL Lesevorgang auf den nun im exakten Moment vorliegenden aktuellen Bestand statt. Hiermit wird erreicht, dass der Fall „zwei Anwender buchen zur gleichen Zeit den gleichen Artikel“ auf eine absolut kleine Zeitspanne minimiert ist. Diese Noch-Mal-Kurz-Vor-Dem-Schreiben-Lesen Logik wurde in das mobile Tool standardmäßig nicht hinein programmiert – lässt sich aber kundenindividuell natürlich problemlos nachrüsten.

Dafür ist der Online Client für SQL unter Android kostenfrei erhältlich. Ihr braucht allerdings AE WWSLite in der SQL Version und natürlich die Datenbanken auf dem SQL Server, um es zu verwenden.

Informationen und das Benutzerhandbuch findet ihr auch auf der Webseite zum Produkt.

Text und Idee: Hans-Jochen Walter, © AE SYSTEME 2025