

Datenbanken und Informationssysteme

Semesterarbeit Part II

Umsetzung einer Inventarisierungslösung

Ausgangslage

Zwischenzeitlich wurden im Unterricht weitere theoretische Elemente der Datenbankentwicklung erarbeitet. Diese Theorie wird laufend mit gezielten Praxisübungen und Workshops vertieft. Das so erworbene Wissen soll nun auf Basis der Semesterarbeit des ersten Semesters durch die Umsetzung einiger funktionaler Anforderungen einer Datenbankapplikation weiter vertieft und dokumentiert werden.

Im Zuge der „Semesterarbeit Part I“ wurde für die Firma Muster AG ein Datenbankmodell erarbeitet, das die Speicherung aller von der Muster AG geforderten Daten ermöglichen soll.

In einem weiteren Schritt wurden nun vom Kunden diverse funktionale Anforderungen definiert, welche in der Datenbank abgebildet bzw. ermöglicht werden sollen.

Die Theorie für die Umsetzung dieser Anforderungen, wird im Unterricht laufend erarbeitet. Es besteht jedoch kein zwingender Zusammenhang zwischen der Reihenfolge der Anforderungen und der im Unterricht vermittelten Inhalte.

Hinweis: Der Aspekt, dass es grundsätzlich vermieden werden sollte, zu viel Logik in die Datenbank zu packen (mittels Stored Functions bzw. Stored Procedures), wird für diese Arbeit komplett ignoriert.

Auftrag

Die Definition „Semesterarbeit Part I“ stellt einen integralen Bestandteil der Definition vorliegender „Semesterarbeit Part II“ dar. Die Beschreibung der Daten, die der Kunde verwalten möchte wird daher in diesem Dokument nicht wiederholt.

Sie haben in der „Semesterarbeit Part I“ das Datenmodell für die Datenbankapplikation der Muster AG erarbeitet. Dieses Datenmodell soll nun mit den gewünschten funktionalen Anforderungen erweitert werden.

Erstellen Sie dazu Abfragen (evt. unter Verwendung von Views) damit die Daten, die der Kunde zu verwalten wünscht, sinnvoll ausgegeben werden können. Es wäre z.B. denkbar, dass diese Abfragen von einer Softwareapplikation verwendet werden, um die Daten entsprechend anzuzeigen. Diese ist jedoch, ausgenommen den Bestandteilen unter dem Kapitel ‚Client‘, nicht Inhalt dieser Arbeit. Überlegen Sie sich, wo es evtl. Sinn macht, Stored Functions bzw. Stored Procedures zu verwenden.

Als Grundlage für diese Semesterarbeit entwickeln Sie die von Ihnen konstruierte Lösung weiter. Es kann dabei vorkommen, dass Sie die Eine oder Andere Schwäche Ihres bisherigen Modells korrigieren müssen. Dies kann zur Folge haben, dass Sie das bereits entwickelte Modell gegebenenfalls anpassen müssen.

Funktionale Anforderungen

Nachfolgend sind die funktionalen Anforderungen, welche durch die Datenbanklösung abgedeckt werden müssen, beschrieben:

PoD-Statistik

Für den Planer soll eine Übersicht erzeugt werden, welche die durchschnittliche Auslastung der Netzwerkinterfaces pro Gerätekategorie (Switch, PC, etc.) wiedergibt. (Beispiel: 24 Port-Switch ist mit 16 belegten Ports zu 66% ausgelastet)

Hierfür ist eine Serie von VIEWS zu erzeugen:

- *v_UsagePerLocation*
Zeigt die Auslastung für jede Gerätekategorie pro Location an
- *v_UsagePerPod*
Zeigt die Auslastung für jede Gerätekategorie pro PoD an

Finden von freien Netzwerkinterfaces im Device

Für die Planung soll eine VIEW angelegt werden, welche die freien Netzwerkinterfaces pro Gerät auflistet. Die Ausgabe soll dabei die *Location-Name, Hostname, Portnummer, Medium und Geschwindigkeitsoptionen* des freien Ports enthalten.

Logging

Der Kunde hat in einigen PoD eine SNMP-Monitoring-Lösung installiert, welche SNMP-Ereignisse automatisch an die zentrale Geräteverwaltung senden kann. Die SQL-Scripts für die Anbindung an diese Lösung sollen so einfach wie möglich gestaltet werden. Konkret bedeutet dies, dass eine Stored-Procedure *LogMessageAdd* zur Verfügung stehen muss, welche unter Angabe von PoD- und Hostname des Geräts, dem Level und der effektiven Nachricht, einen entsprechenden Logeintrag in der Datenbank erzeugt.

Schnittstelle zu bestehendem Gerätemanagement-Werkzeug

Die Firma verfügt über ein beliebtes Monitoring-Tool, das es erlaubt, Log-Einträge für Geräte im System zu visualisieren und ggf. zu quittieren. Das Tool erwartet eine Tabelle mit dem Namen *v_logentries* und folgender Struktur:

Attribut	Datentyp	Beschreibung
id	INT	Eindeutige Identifikation des Logeintrags
pod	TEXT	Name des POD in welchem die Nachricht aufgetreten ist
location	TEXT	Name der Location in welcher die Nachricht aufgetreten ist
hostname	TEXT	Hostname des Geräts dem die Nachricht zugeordnet ist
severity	INT	Schwere des Logeintrags (größer = schwerwiegender)
timestamp	TIMESTAMP	Zeitpunkt des Logeintrags
message	TEXT	Lognachricht

Das Tool geht davon aus, dass diese Tabelle lediglich Einträge enthält, welche noch nicht quittiert wurden. Für das Quittieren erwartet das Tool, dass eine Stored Procedure *LogClear(IN Id INT)* zur Verfügung steht, welche die ID der zu quittierenden Logmessage erwartet.

Ein Testtool steht Ihnen zur Verfügung (siehe MonitorClient.zip).

Automatisierte Fakturierung

Die Fakturierungsprozesse sollen automatisiert werden. Kunden verfügen über ein laufendes Konto mit welchem Leistungen gegengerechnet werden können. Trifft eines der nachfolgenden Kriterien zu, werden die offenen Positionen automatisch in Rechnung gestellt:

1. Der aktuell offene (noch nicht fakturierte) Betrag überschreitet eine vom Kunden festgelegte Grenze
2. Am 28. des Monats überschreitet der offene Betrag eine Mindestgrenze von 1000 CHF
3. Mindestens eine offene Position ist älter als 3 Monate

Bei der Erstellung der Abrechnung soll ein allfälliges Guthaben des Kunden automatisch berücksichtigt werden. Wird Guthaben des Kunden gegengerechnet, so soll dies explizit als Position auf der Abrechnung ausgewiesen werden.

Manuelle Fakturierung

Neben der automatisierten Fakturierung kann ein Sachbearbeiter auch jederzeit auf Wunsch des Kunden eine Abrechnung für einen bestimmten PoD erstellen. Hierfür ist eine Stored Procedure *PodBill* vorzusehen.

Bei der Erstellung der Abrechnung soll ein allfälliges Guthaben des Kunden automatisch berücksichtigt werden. Wird Guthaben des Kunden gegengerechnet, so soll dies explizit als Position auf der Abrechnung ausgewiesen werden.

Berechtigungskonzept

Folgende Berechtigungen sind auf Datenbankebene vorzusehen:

- Der Geschäftsführer hat Zugang zu sämtlichen Daten im System.
- Nur Abteilungsleiter können neue PODs oder Locations anlegen.
- Nur der Geschäftsführer kann finanzielle Aspekte im System einsehen.
- Sachbearbeiter können bis auf vorgängig definierte Ausnahmen alles einsehen und bearbeiten.
- Logger haben ausschliesslich die Berechtigung Logeinträge zu erzeugen.
- Geräte-Monitoring-Tools können ausschliesslich Logeinträge abfragen und ggf. quittieren.

Lieferergebnisse	<p>Folgende Ergebnisse werden erwartet:</p> <ul style="list-style-type: none"> • ERM (grafisch) • DDL – Script (gegen MySQL ausführbare Textdatei *.sql) • DML – Script (gegen MySQL ausführbare Textdatei *.sql) • Dokumentation • Arbeitsjournal
<i>ERM</i>	<p>Erstellen Sie ein ERM, welches das Model Ihrer Datenbank aufzeigt. Dabei sollen alle funktionalen Aspekte ersichtlich sein. Dokumentieren Sie die für die Aufgabenstellung bzw. für die funktionalen Anforderungen relevanten Aspekte: „wie/warum wurde welcher Aspekt genauso im Model umgesetzt“. Das ERM soll so umgesetzt werden, dass die Datenstrukturen funktional (d.h. die für den Kunden bestmögliche Funktionalität), stabil, optimiert, wartbar und erweiterbar sind.</p>
<i>DDL-Script Beispieldaten (DML) Import-Script</i>	<p>Erstellen Sie korrespondierend zu Ihrem ERM ein DDL-Script, um die Datenbank entsprechend dem ERM vollständig anzulegen. Das DDL-Script muss in einer Datei abgeliefert werden. Es darf KEIN Forward-Engineering eingesetzt werden!</p> <p>Zusätzlich erstellen Sie ein DML-Script, dass für jede Entität eine, bzgl. Test und Vorführung Ihrer Lösung sinnvolle, Anzahl von Datensätzen anlegt (Hinweis: eine sinnvolle Anzahl ist immer grösser als 1). Das Script für die Datenerfassung soll auch in einer separaten Datei abgeliefert werden.</p>
<i>Dokumentation</i>	<p>Erstellen Sie eine Dokumentation, in der Sie alle relevanten Punkte der Aufgabenstellung sowie der Lieferergebnisse dokumentieren. Insbesondere sollen alle funktional relevanten Aspekte des ERM in die Dokumentation einfließen (siehe ERM). Beschreiben Sie dabei u.a. auch die Handhabung Ihrer Lösung. Zielgruppe für diese Dokumentation sind Entwickler die sich an Ihr entwickeltes Datenmodell ankoppeln sollen.</p> <p>In der Dokumentation soll in Form eines kleinen Arbeitsjournals auch ersichtlich sein, welcher Gruppenteilnehmer welche Aspekte erarbeitet hat sowie wieviel Zeit aufgewendet wurde.</p> <p>Das Dokument soll dabei den gängigen Richtlinien eines Fachdokumentes folgen (z.B. Inhaltsverzeichnis, Glossar usw.). Siehe dazu auch das Dokument „DB - Dokumentationsbeispiel.pdf“.</p>
Hilfsmittel	<p>Grundsätzlich gibt es keine Einschränkungen bzgl. Hilfsmittel. Es steht die komplette Infrastruktur des ZbW zur Verfügung. Die Lieferergebnisse müssen jedoch zwingend selbstständig bzw. im Team erarbeitet werden.</p>
Unterstützung	<p>Der Dozent steht für die Klärung von Verständnisfragen zur Verfügung. Dies im Rahmen der Unterrichtsblöcke am ZbW sowie per E-Mail. Vom Dozenten werden keine Lösungen bzw. Teillösungen angeboten.</p>
Arbeitsform	<p>Die Arbeit wird in den gleichen Gruppenzusammensetzungen wie im letzten Semester durchgeführt. Die Gruppen dürfen während der Arbeit nicht mehr geändert werden. Es wird darauf Wert gelegt, dass die Aufgabenstellungen von allen Gruppen selbstständig erarbeitet werden. D.h. es werden keine gleichen Lösungen akzeptiert. Zudem soll jedes Gruppenmitglied in gleichem Masse involviert werden.</p>
Zeitraum	<p>Für die Umsetzung steht ihnen das verbleibende Semester zur Verfügung. Beachten Sie die Termine auf Moodle. Die Erarbeitung des Semesterprojekts ist Teil des Selbststudiums.</p>
Abgabe	<p>Die komplette Dokumentation (ERM, Dokumentation, Arbeitsjournal) sowie die Scripts (DML-/DDL-Script) sind in digitaler Form per Moodle einzureichen.</p>

Benotung	Diese Arbeit wird nicht im klassischen Sinne benotet – es handelt sich jedoch um ein Testat. Ein Erfüllen der Aufgabe ist zwingend erforderlich, um für die Modulprüfung zugelassen zu werden. Im weiteren Verlauf werden zudem andere Module an den Ergebnissen dieser Arbeit anknüpfen, weshalb es entscheidend ist, eine gute Basis zu schaffen.
Hinweise	<p>Versuchen Sie zuerst, sich einen Überblick über die Thematik bzw. die Aufgabenstellung/den Kundenauftrag zu verschaffen. Dazu verwenden Sie optimal eine grafische Form (z.B. ein Mindmap, eine Handskizze, UseCases o.ä.). Teilen Sie dabei die Aufgabenstellung in Meilensteine ein und identifizieren bzw. definieren Sie Arbeitsschritte, die parallel erledigt werden können und so auf die verschiedenen Gruppenmitglieder aufgeteilt werden können. D.h. zuerst identifizieren, dann planen, dann umsetzen!</p> <p>Vermeiden Sie, sich unnötig mit Tools rumschlagen zu müssen, die zum Ergebnis bzw. zur Qualität der Arbeit nichts beitragen.</p> <p>Beschäftigen Sie sich nicht von Beginn weg mit Detailfragen. Treiben Sie das Projekt schnellst möglich Richtung Lösung voran und behandeln Sie die Detailfragen in weiteren Schritten. Stichwort „80/20-Prinzip bzw. Pareto-Prinzip“ sowie „Vertikale Integration“.</p> <p>Seien Sie grundsätzlich kreativ. Die Aufgabenstellung wurde bewusst offen definiert um Ihre Kreativität zu fördern. Sollte etwas nicht definiert sein, treffen Sie Annahmen im Sinne des Kunden bzw. des Kundenauftrages/der Aufgabenstellung und dokumentieren Sie diese.</p>
Anhang	Vorlage Arbeitsjournal

