|  |  |
| --- | --- |
|  | Network Management  Datenmodell-Dokumentation |
|  | C:\Users\Ruel\AppData\Local\Microsoft\Windows\INetCache\Content.Word\diagram-printscreen.png |
| 7/30/2015 | Erweiterung einer Inventarisierung |
|  | Hinweis: Die Definition ‚Semesterarbeit 1‘ stellt einen integralen Bestandteil dieser vorliegender Arbeit dar. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Autoren** | Ruel Holderegger Gottfried Mayer Maurus Kühne Elias Schwarz | Abgabedatum: | 18. September 2015 |
| Version: | 1.0 |

Inhalt

[Grundlagen 4](#_Toc430278470)

[1. Ausgangslage 4](#_Toc430278471)

[2. Funktionale Anforderungen 4](#_Toc430278472)

[2.1. Finden von freien Netzwerkinterfaces im Device 4](#_Toc430278473)

[2.2. Stored Procedure zur Instanzierung eines Devices 4](#_Toc430278474)

[2.3. Automatisierte Fakturierung 4](#_Toc430278475)

[2.4. Manuelle Fakturierung 5](#_Toc430278476)

[2.5. POD-Statistik 5](#_Toc430278477)

[2.6. Produktkatalog 5](#_Toc430278478)

[2.7. Schnittstelle für Management-Meldungen 5](#_Toc430278479)

[2.8. Schnittstelle zu bestehenden Gerätemanagement-Werkzeug 5](#_Toc430278480)

[2.9. Berechtigungskonzept 6](#_Toc430278481)

[2.10. Client 6](#_Toc430278482)

[Überarbeitetes Datenmodell (ERM) 6](#_Toc430278483)

[Umsetzung der Anforderungen 7](#_Toc430278484)

[1. Finden von freien Netzwerkinterfaces im Device 7](#_Toc430278485)

[2. Stored Procedure zur Instanzierung eines Devices 7](#_Toc430278486)

[3. Automatische Fakturierung 7](#_Toc430278487)

[4. Manuelle Fakturierung 7](#_Toc430278488)

[5. POD-Statistik 7](#_Toc430278489)

[6. Produktkatalog 7](#_Toc430278490)

[7. Schnittstelle für Management-Meldungen 7](#_Toc430278491)

[8. Schnittstelle zu bestehendem Gerätemanagement-Werkzeug 7](#_Toc430278492)

[9. Berechtigungskonzept 8](#_Toc430278493)

[10. Client 8](#_Toc430278494)

[Lieferergebnisse 9](#_Toc430278495)

[Tabellenreferenz 9](#_Toc430278496)

[11. Device 9](#_Toc430278497)

[12. DeviceType 9](#_Toc430278498)

[13. DeviceCategory 10](#_Toc430278499)

[14. DeviceSwitch 10](#_Toc430278500)

[15. DeviceRouter 10](#_Toc430278501)

[16. DeviceMachine 11](#_Toc430278502)

[17. MediumType 11](#_Toc430278503)

[18. Rate 11](#_Toc430278504)

[19. NetworkInterface 12](#_Toc430278505)

[20. RelNetworkInterface 12](#_Toc430278506)

[21. RelNetworkInterfaceVLAN 12](#_Toc430278507)

[22. Invoice 13](#_Toc430278508)

[23. InvoicePosition 13](#_Toc430278509)

[24. Location 14](#_Toc430278510)

[25. POD 14](#_Toc430278511)

[26. Address 14](#_Toc430278512)

[27. Credential 15](#_Toc430278513)

[28. Customer 15](#_Toc430278514)

[ProzedurenReferenz 15](#_Toc430278515)

[1. P\_DeviceAdd 15](#_Toc430278516)

[2. P\_LogMessageAdd 16](#_Toc430278517)

[3. P\_NightlyInvoicing 16](#_Toc430278518)

[4. P\_PodBill 16](#_Toc430278519)

[SichtenReferenz 16](#_Toc430278520)

[1. V\_UsagePerLocation 16](#_Toc430278521)

[2. V\_UsagePerPod 17](#_Toc430278522)

[3. V\_DeviceCatalog 17](#_Toc430278523)

[4. V\_FreeNetworkInterfaces 17](#_Toc430278524)

[Konfigurationen 18](#_Toc430278525)

[Literatur 18](#_Toc430278526)

[Abkürzungsverzeichnis 18](#_Toc430278527)

[Abbildungsverzeichnis 18](#_Toc430278528)

[Revisionsübersicht 18](#_Toc430278529)

[Arbeitsjournal 18](#_Toc430278530)

Datenmodell-Dokumentation

Erweiterung einer Inventarisierung

# Grundlagen

Die in diesem Kapitel dargelegte3n Grundlagen stützen sich auf die im Unterricht verteilten Aufgabenstellung ‚Semesterarbeit Part 2: Umsetzung einer Inventarisierung‘.

## 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 wird nun auf Basis der Semesterarbeit des ersten Semesters durch die Umsetzung einiger funktionaler Anforderungen ein Datenbankapplikation weiter vertieft und dokumentiert.

## Funktionale Anforderungen

Die hier dargestellten Anforderungen sind direkte Zitate aus der Aufgabenstellung.

### 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.

### Stored Procedure zur Instanzierung eines Devices

Das Hinzufügen eines neuen Geräts im Verlauf des Planungsprozesses soll durch eine Stored Procedure *DeviceAdd* abstrahiert werden. Beim Aufruf der Stored Procedure werden lediglich Location ID, ein Verweis in die Device-Bibliothek und die Anzahl hinzuzufügender Geräte angegeben. Die übrigen Schritte werden automatisiert durch die Stored Procedure ausgeführt.

### Automatisierte Fakturierung

Die Fakturierungsprozesse sollen automatisiert werden. Kunden verfügen über ein laufendes Konto mit welchen 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 vom CHF 1000.--.
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, 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 gerechnet, so soll dies explizit als Position auf der Abrechnung ausgewiesen werden.

### 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 Geräteklasse pro Location an.
* **V\_UsagePerPod**: Zeigt die Auslastung für jede Geräteklasse pro POD an.

### Produktkatalog

Die Marketingabteilung möchte auf der Website einen interaktiven Katalog herausgeben, in welchem registrierte Kunden stöbern können. Dabei sollen folgende Eckdaten zur Verfügung stehen:

* Gerätedaten (Hersteller, Typenbezeichnung, Beschreibung, etc.)
* Netzwerkinterfaces mit jeweiliger Nummer sowie den technischen Möglichkeiten wie Geschwindigkeit, Technologie(z.B. 1000BASE-T, 100Base-T, 10BASE-T, 10GBASE-LW4, etc.), Übertragungsmodus (Voll-/Halbduplex), …
* Preis

Um Webentwickler eine einfache Schnittstelle bieten zu können, wird als Liste für weitere Abfragen eine VIEW mit der Bezeichnung *V\_DeviceCatalog* angelegt, auf der die notwendigen Abfragen ausgeführt werden können. Diese View soll Geräte welche nicht mehr verfügbar sind aus dem Bestand herausfiltern um zu vermeiden, dass die Kunden diese finden und eventuell anfordern.

### Schnittstelle für Management-Meldungen

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 soll 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 Schweregrad un der effektiven Nachricht einen entsprechenden Logeintrag in der Datenbank macht.

### Schnittstelle zu bestehenden 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 |
| 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.

### 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.

### Client

Es soll ein Softwareclient erstellt werden, mit dem folgendes möglich ist.

* Auswählen eines POD
* Fakturierung eines POD auslösen
* Rechnung auswählen und anzeigen (tabellarisch ist ausreichend)
* Konfiguration für ein Gerät im POD erzeugen (Das Format der Konfiguration finden Sie Anhang „Konfigurationsformat“)

Die Technologie, die für die Erstellung des Clients zum Einsatz kommt, ist Ihnen freigestellt (z.B. Microsoft C #.Net, Java o.ä. – Stichworte: ADO.NET, JDBC usw.). Der Knowhow-Fokus des Dozenten bzgl. Unterstützung liegt dabei auf Java/JDBC – was nicht heisst, dass zwingen diese Technologie verwendet werden muss.

# Überarbeitetes Datenmodell (ERM)

# Umsetzung der Anforderungen

### Finden von freien Netzwerkinterfaces im Device

Für die Ermittlung der freien Netzwerkinterfaces wurde die View v\_freenetworkinterfaces angelegt. Diese selektiert folgende Felder:

* Description
* Hostname
* PortNr
* MediumType
* MaxSpeed

Freie Netzwerkinterfaces werden ermittelt, in dem geprüft wird, ob keine Beziehung zweier Networkinterfaces (in der Tabelle RelNetworkInterface) vorhanden ist.

### Stored Procedure zur Instanzierung eines Devices

Mit der Prozedur P\_DeviceAdd können neue Geräte hinzugefügt werden.

|  |  |  |
| --- | --- | --- |
| **Inputparameter** | **Typ** | **Beschreibung** |
| ID\_Location | INT | ID der Location, an der sich das Gerät befindet |
| ID\_DeviceType | INT | Typ des Geräts. Siehe Tabelle DeviceType |
| NumDevices | INT | Anzahl zu erstellende Geräte |

### Automatische Fakturierung

Die automatische Fakturierung wurde mittels MySQL-Event **E\_NightlyInvoicing** vorgenommen: Das Event wird jede Nacht um 23:00 Uhr ausgeführt. Die Prozedur prüft pro Kunde, ob das Erstellen einer neuen Rechnung nötig ist oder nicht.

### Manuelle Fakturierung

Die Manuelle Fakturierung wird über die Anwendung über die Schaltfläche ‚Fakturierung‘ gestartet. Weitere Details siehe Client-Anleitung.

### POD-Statistik

Für die POD-Statistik wurden die Views v\_usagePerLoaction und v\_usagePerPod angelegt.   
Weitere Infos zu den Feldern siehe Kapitel Sichtenreferenz.

### Produktkatalog

Für den Produktkatalog wurde die View **v\_deviceCatalog** erstellt. Es werden nur die verfügbaren Geräte ausgelistet, d.h. solche bei denen die Gerätekategorie nicht inaktiv gesetzt worden ist.

Details zu den Feldern, die selektiert werden sie Kapitel **Sichtenreferenz**.

### Schnittstelle für Management-Meldungen

Für die Schnittstelle für Managmentmeldungen wurde die Prozedur P\_LogMessageAdd erstellt. Die gemachten Log-Einträge in die Tabelle Log gespeichert.

### Schnittstelle zu bestehendem Gerätemanagement-Werkzeug

Hierfür wurde die View v\_logentries erstellt. Diese View gibt nur nicht quittierte Logeinträge zurück. Nachrichten können über die Prozedur P\_LogClear quittiert werden.

Details zu den Felder die selektiert werden siehe Kapitel Sichtenreferenz.

### Berechtigungskonzept

Das vorhergesehene Berechtigungskonzept wurde wie folgt umgesetzt:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Benutzername** | **Passwort** | **Bemerkungen** |
| 1. | gf | GSgsEgHN5e | Geschäftsführer: Hat Zugang zu sämtlichen Daten im System |
| 2. | al | fQf2EY7bb9 | Abteilungsleiter: kann neue PODs oder Locations anlegen. |
| 3. | sb | WnynxpgaNu | Sachbearbeiter: kann bis auf vorgängig definierte Ausnahmen alles einsehen und bearbeiten |
| 4. | log | NrjP6mmX4J | Logger: hat ausschliesslich die Berechtigung Logeinträge zu erzeugen. |
| 5. | tool | gv2EnGJjZc | Können ausschliesslich Log-Einträge abfragen und ggf. quittieren |

### Client

|  |  |
| --- | --- |
| **Anforderung** | **Umschreibung der Umsetzung** |
| Auswählen eines POD | Entweder über *Stammdaten, POD* oder *Stammdaten, Kunden* zuerst anwählen (Doppelklick): Danach sind alle PODs des Kunden ersichtlich |
| Fakturierung eines POD auslösen | Voraussetzung ist, dass zuerst ein Kunde geöffnet worden ist. Danach muss zuerst ein POD angewählt werden  Sofern fakturierende Positionen vorhanden sind, können diese fakturiert werden mit der Schaltfläche „Fakturieren“ |
| Rechnung auswählen und anzeigen | Rechnungen können über den *Stammdaten, Rechnungen* angezeigt werden. Die Positionen werden unter *Stammdaten - Rechnungspositionen* aufgelistet.  Ist ein Kunde geöffnet, werden im unteren Bereich die Rechnungen angezeigt. Mit einem Doppelklick auf eine Rechnung wird ein neues Register mit der Rechnung geöffnet. Darin sind die einzelnen Rechnungspositionen ersichtlich. |
| Konfiguration für ein Gerät im POD erzeugen | Unter *Geschäftsfälle – Konfiguration erzeugen*  Nach dem Anwählen kann die Nr. des Gerätes, welche eine Konfiguration erhält eingegeben werden.  Mit GO bestätigen und die Konfiguration wird nach ein paar Sekunden erstellt. |

# Lieferergebnisse

Im folgenden Abschnitt werden die Lieferergebnisse detailliert beschrieben.

|  |  |
| --- | --- |
| ERM | Das komplette ERM wird als PNG separat ausgeliefert. |
| DDL-Script | Mit dem Script „DDL-Script.sql“ werden sämtliche Tabellen angelegt. Zudem beinhaltet das Script auch alle erstellten Procedures und Views und Functions, die in der Aufgabenstellung verlangt wurden. |
| DML-Script | Mit dem Script „DML-Script.sql“ werden sämtliche Beispieldaten in den definierten Tabellen eingefügt. |
| Client | Der Client wird sowohl als Source Code als auch als kompilierte Anwendung ausgeliefert. Damit der Source Code kompiliert werden kann, sind folgende Voraussetzungen notwendig:   * .Net in der Version 4.5.2 oder höher * Visual Studio 2013 oder höher * Installierter MySQL Entity Framework Driver (bei kompilierter Version mitgeliefert) * DevExpress 14.2 (bei kompilierter Version mitgeliefert) |

# Tabellenreferenz

Die Tabellenreferenz dient der detaillierten Dokumentation der einzelnen Tabellen. Jede Tabelle wird hierbei bis auf Feldebene dokumentiert.

### Device

Diese Tabelle umfasst alle Geräte, die zukünftig verkauft werden sollen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Device | INT | Nein | Eindeutiger ID der Tabelle |
|  | ID\_DeviceType | INT | Nein |  |
|  | ID\_Location | INT |  |  |
|  | Hostname | VARCHAR(500) |  |  |
|  | IPv4\_Adresse | VARCHAR(500) |  |  |
|  | IPv6\_Adresse | VARCHAR(500) |  |  |
|  | SerialNumber | VARCHAR(500) |  |  |
|  | IsPhysical | TINYINT(1) |  |  |

### DeviceType

Sämtliche Gerätetypen werden in dieser Tabelle gehalten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_DeviceType | INT(11) | Nein | Eindeutiger ID der Tabelle |
|  | ID\_DeviceCategory | INT(11) | Nein | Id der zugehörigen Gerätekategorie |
|  | Manufacturer | VARCHAR(100) | Ja | Bezeichnung des Herstellers |
|  | ManufacturerNumber | VARCHAR(100) | Ja | Hersteller-Nummer |
|  | NumInterface | INT(11) | Nein | Anzahl Ports diese Gerätetyps |
|  | ID\_Rate | INT(11) | Ja | Id der zugehörigen Preislisten-Position |
|  | ID\_MediumType | INT(11) | Ja | Id des zugehörigen Mediums |

### DeviceCategory

Sämtliche verfügbare Gerätekategorien sind in dieser Tabelle verzeichnet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_DeviceCategory | INT | Nein | Eindeutiger ID der Tabelle |
|  | Description | VARCH(100) |  | Beschreibung der Gerätekategorie |
|  | InActive | TINYINT(1) | Nein | Nicht mehr bereitgestellte Gerätekategorien können mit diesem Flag inaktiv gesetzt werden. |

### DeviceSwitch

Handelt es sich beim Gerät um ein Switch, werden die speziefischen Daten zum Switch hier gespeichert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Device | INT | Nein | Eindeutiger ID der Tabelle |
|  | isPoESupported | TINYINT |  |  |

### DeviceRouter

Handelt es sich beim Gerät um einen Router, werden die spezifischen Daten zum Router hier gespeichert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Device | INT | Nein | Eindeutiger ID der Tabelle |
|  | Description | VARCHAR(100) | Ja | Bezeichnung des Routers |

### DeviceMachine

Handelt sich bei einem Gerät um eine Maschine, werden diese spezifischen Daten hier gespeichert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Device | INT | Nein | Eindeutiger ID der Tabelle |
|  | Processors | VARCHAR(100) |  |  |
|  | RAM | Varchar(100) |  |  |
|  | StorageSize | Varchar(100) |  |  |
|  | GPU | Varchar(100) |  |  |

### MediumType

In dieser Tabelle sind alle verfügbaren Medientypen verzeichnet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_MediumType | INT | Nein | Eindeutiger ID der Tabelle |
|  | Description | VARCHAR(100) |  | z.B. 100 BaseTX, 100 BaseTX |
|  | Fullduplex | TINYINT(1) |  | 1 = Ja = Fullduplex 0 = Nein = Halfduplex |
|  | Speed | INT |  | Geschwindgkeit des Mediums in Mbit/s |

### Rate

Verrechnung der Arbeitsstunden, kann aber auch für die Preisdefinierung von Gerätekategorien DeviceCategory verwendet werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Rate | INT | Nein | Eindeutiger ID der Tabelle |
|  | ID\_ServiceType | INT | Ja |  |
|  | ID\_Unit | INT | Ja |  |
|  | Price | DECIMAL(11) |  | Preis in CHF |
|  | ValidFrom | DATE |  |  |
|  | ValidTo | DATE |  |  |

### NetworkInterface

Sämtliche Netzwerkinterfaces werden in dieser Tabelle gepeichert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_NetworkInterface | INT | Nein | Eindeutiger ID der Tabelle |
|  | Physical | TYNINT | Ja | Beschreibt, ob es sich beim Interface um ein physicalisches oder virtuelles Netzwerkinterface handelt 1 = Ja = physikalisches Interface 0 = Nein = virtuelles Interface |
|  | IPv4\_Address | INT(10) | Ja | IPv4-Adresse |
|  | IPv6\_Address | VARBINARY(16) | Ja | IPv6-Adresse |
|  | ID\_Device | INT(11) | Ja | Zugehöriges Gerät des Netzwerkinterfaces |
|  | PortNr | INT(11) | Ja | Portnummer, auf welchem das Interface läuft |
|  | ID\_MediumType | INT(11) | Ja | Zugehöriges Medium zum Netwerkinterface |
|  | Description | VARCHAR(100) | Ja | Bezeichnung des Netzwerkinterfaces |

### RelNetworkInterface

Beziehungstabelle zwischen einem Netzwerkinterface A und einem Netzwerkinterface B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_RelNetworkInterface | INT | Nein | Eindeutiger ID der Tabelle |
|  | ID\_NetworkInterfaceA | INT | Nein |  |
|  | ID\_NetworkInterfaceB | INT | Nein |  |

### RelNetworkInterfaceVLAN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_RelNetworkInterface | INT | Nein | Eindeutiger ID der Tabelle |
|  | ID\_VLAN | INT | Nein |  |

### Invoice

Rechnungen werden in dieser Tabelle verzeichnet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Invoice | INT(11) | Nein | Eindeutiger ID der Tabelle |
|  | InvoiceDate | DATE | Ja |  |
|  | ID\_Customer | INT(11) | Nein |  |
|  | Closed | TYNINT | Nein | 0 = Fakturiert 1 = Verrechnet |

### InvoicePosition

Beschreibt eine Position der Rechnung.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_InvoicePosition | INT | Nein | Eindeutiger ID der Tabelle |
|  | ID\_Invoice | INT | Nein |  |
|  | ID\_ServiceType | INT | Ja |  |
|  | Amount | DECIMAL(10,0) |  |  |
|  | Price | DECIMAL(10,0) |  |  |
|  | Description | VARCHAR(100) |  |  |
|  | ID\_NetworkInterface | INT |  | So können Arbeiten auf spezifischen Netzwerkinterfaces explizit abgerechnet werden |
|  | ID\_Device | INT |  | Ist in Fällen,wo beispielsweise der Kunde ein Gerät kauft, gefüllt. |
|  | ID\_Location | INT |  |  |

### Location

Sämtliche Adress- und Ortsangaben sind in dieser Tabelle verzeichnet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Location | INT(11) | Nein | Eindeutiger ID der Tabelle |
|  | ID\_POD | INT(11) | Nein | ID der zugehörigen POD |
|  | ID\_Address | INT(11) | Nein | ID der zugehörigen Adresse |
|  | ID\_ParentLocation | INT(11) | Ja |  |

### POD

Sämtliche Felder eines „Point of Delivery“ werden in dieser Tabelle verwaltet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_POD | INT | Nein | Eindeutiger ID der Tabelle |
|  | Description | VARCHAR(100) | Ja | Beschreibung des POD |
|  | ID\_Customer | INT | Ja | ID des zugehörigen Kunden |
|  | ID\_ContactPerson | INT | Ja | ID der zugehörigen Kontaktperson |
|  | TimeZone | VARCHAR(50) | Ja | Zeitzone des POD Standardwert: Europe/Zurich |
|  | Description | VARCHAR(100) | Nein | Bezeichnung des POD |
|  | DomainName | VARCHAR(100) | Ja | Name der Domäne |

### Address

In dieser Tabelle werden die Daten der Adresse gespeichert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Adress | INT(11) | Nein | Eindeutiger ID der Tabelle |
|  | Street | VARCHAR(100) | Ja | Strassenbezeichnung |
|  | StreetNumber | VARCHAR(8) | Ja | Nummer |
|  | PLZ | VARCHAR(100) | Ja | Postleitzahl zum Ort |
|  | Location | VARCHAR(100) | Ja | Bezeichnung des Ortes |
|  | ISOCountry | VARCHAR(8) | Ja | ISO Ländercode |
|  |  |  |  |  |

### Credential

Diese Tabelle führt die Zugangsdaten für die Inventarisierung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Credential | INT(11) | Nein | Eindeutige ID der Tabelle |
|  | ID\_Customer | INT(11) | Nein | Id des zugeordneten Kunden |
|  | Username | VARCHAR(100) | Ja | Benutzername |
|  | Password | VARCHAR(100) | Ja | Passwort |
|  | SNMP\_Community | VARCHAR(100) | Ja |  |

### Customer

Diese Tabelle führt sämtliche Daten des Kunden

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Attributname** | **Datentyp** | **Optional** | **Bemerkungen** |
| PK | ID\_Customer | INT(11) | Nein | Eindeutige ID der Tabelle |
|  |  |  |  |  |

# ProzedurenReferenz

Sämtliche erstellte Prozeduren sind hier beschrieben. Als Namenskonvention wird ein vorangehendes „P\_“ vor sämtlichen Prozedurbezeichnungen verwendet.

### P\_DeviceAdd

Fügt ein neues Gerät hinzu. Eingabeparameter:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Parametername** | **Datentyp** | **Bemerkungen** |
| 1. | ID\_Location | INT | ID\_Location, an dem sich die neuen Geräte befinden werden |
| 2. | ID\_DeviceType | INT | Type der neuen Geräte |
| 3. | NumDevice | INT | Anzahl Geräte, die hinzugefügt werden sollen |

### P\_LogMessageAdd

Erstellt einen entsprechenden Logeintrag in der Datenbank. Eingabeparameter:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Parametername** | **Datentyp** | **Bemerkungen** |
| 1. | ID\_POD | INT | POD, der geloggt werden soll |
| 2. | Hostname | VARCHAR(100) | Hostname des Gerätes |
| 3. | Severity | INT | Schweregrad der Meldung |
| 4. | Message | VARCHAR(1000) | Meldung, die geschrieben werden soll |

### P\_NightlyInvoicing

Diese Prozedur führt die automatische Fakturierung durch. Eingabeparameter:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Parametername** | **Datentyp** | **Bemerkungen** |
| - | - | - | Keine Eingabeparameter. |

### P\_PodBill

Diese Prozedur wird für die manuelle Fakturierung verwendet. Eingabeparameter:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Parametername** | **Datentyp** | **Bemerkungen** |
| 1. | ID\_POD | INT | POD, welcher explizit fakturiert werden soll |

# SichtenReferenz

Sämtliche Sichten sind hier beschrieben. Als Namenskonvention wird ein vorangehendes „v\_“ vor sämtlichen Sichtenbezeichnungen verwendet.

### V\_UsagePerLocation

Zeigt die Auslastung für jede Geräteklasse pro Location an. Datenfelder:

|  |  |  |
| --- | --- | --- |
| **Attributname** | **Datentyp** | **Bemerkungen** |
| ID\_Location | INT | ID der Location |
| ID\_DeviceType | INT | ID des Gerätetyps |
| Usage | INT | Auslastung in % |

### V\_UsagePerPod

Zeigt die Auslastung für jede Geräteklasse pro POD an. Datenfelder:

|  |  |  |
| --- | --- | --- |
| **Attributname** | **Datentyp** | **Bemerkungen** |
| ID\_Location | INT | ID der Location |
| ID\_DeviceType | INT | ID des Gerätetyps |
| Usage | INT | Auslastung in % |

### V\_DeviceCatalog

Diese View listet alle Geräte, welche verfügbar sind, aus.

|  |  |  |
| --- | --- | --- |
| **Attributname** | **Datentyp** | **Bemerkungen** |
| Manufacturer | VARHAR(100) | Wird noch überarbeitet |
| DeviceTypeDescription | VARCHAR(100) | Beschreibung der Gerätekategorie |
| NumInterfaces | INT(11) | Anzahl Interfaces |
| Speed | INT(11) | Geschwindigkeit in Mbit/s |
| FullDuplex | TINYINT | 1= Ja = Fullduplex 0 = Nein= Halfduplex |
| Technology | VARHCHAR(100) | Technologie aus Tabelle MediumType |
| DevicetypeDescription | VARCHAR(100) | Beschreibung des Gerätetyps |
| Price | Decimal(10,0) | Preis des Gerätes |

### V\_FreeNetworkInterfaces

Ermittelt alle Netzwerk Interfaces, die nicht belegt sind.

|  |  |  |
| --- | --- | --- |
| **Attributname** | **Datentyp** | **Bemerkungen** |
| LocationName | VARCHAR(100) | Nein |
| Hostname | VARCHAR(100) | Ja |
| PortNumber | 5 |  |
| MediumType | VARCHAR(100) |  |
| MaxSpeed | INT | Ja |

# Konfigurationen

Für den MySQL-Zugang wurde der Hoststar-Account von Gottfried Mayer verwendet.

Benutzername: inp1316\_usr  
Passwort: xaPn4Qn5FU  
Port: 3306 oder 465

# Literatur

Ralf Adams – SQL: Eine Einführung mit vertiefenden Exkursen

# Abkürzungsverzeichnis

# Abbildungsverzeichnis

# Revisionsübersicht

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Änderung | Autor |
|  |  |  |  |

# Arbeitsjournal

Sämtliche durchgeführte Arbeitsstunden sind hier verzeichnet.

|  |  |  |
| --- | --- | --- |
| **Alle** | | |
| **Datum Uhrzeit von / bis** | **Dauer** | **Tätigkeit** |
| 03. Juli 2015  17:00 – 20:00 | 3 h | Kickoff-Meeting  Arbeitsaufteilung vorgenommen |
| 24.08.2015 18:00 – 20:00 | 2 h | Erstellung Todo-Liste für Projektabschluss:  Folgende Punkte wurden noch als offen gekennzeichnet:  Freie Netzwerkinterfaces finden Produktkatalog Berechtigungskonzept Client – Konfiguration erstellen für ein Gerät aus POD Client – Rechnung anzeigen  Fakturierung auslösen (Check ob Rechn. Wirklich erstellt wird) Installations-/ / Konfigurationsanleitung ERM (generieren) DDL-Script (generieren) Beispieldaten Dokumentation - Umsetzungen -Lieferergebnisse - Arbeitsjournal  Offene Punkte:  Produktkatalog (Elias!) |
| **Total in Stunden** |  |  |

|  |  |  |
| --- | --- | --- |
| **Elias Schwarz** | | |
| **Datum Uhrzeit von / bis** | **Dauer** | **Tätigkeit** |
| 18.08.2015 17:00 – 19:00 | 2 h | Am Produktkatalog weitergearbeitet  v\_DeviceCatalog |
| 20.08.2015 18:00 – 20:30 | 2.5 h | Diverse Korrekturen vorgenommen |
| 25.08.2015 18:00 – 19:45 | 1.75 h | V\_DeviceCatalog  Beispieldaten in Tabellen erfasst  Erfassung Beispieldaten in übrigen Tabellen  Vorbereitung Präsentation |
| 15.09.2015 17:00 – 20:00 | 3h | Berechtigungskonzept getestet und Fehler korrigiert. |
| **Total in Stunden** |  |  |

|  |  |  |
| --- | --- | --- |
| **Gottfried Mayer** | | |
| **Datum Uhrzeit von / bis** | **Dauer** | **Tätigkeit** |
| 30.07.2015 18:00 – 20:45 | 2.75 h | Tabelle Device geändert: FullDuplex, Speed und IsPhysical auf Nullable gesetzt  Neues Feld ‚Closed‘: Beschreibt, dass die Rechnung gestellt worden ist.  Automatische Fakturierung: Proc P\_Nigthly\_Invoicing erstellt. |
| 06.08.2015 19:30 – 21:00 | 1.5 h | Neues Feld InvoiceTrashhold auf Tabelle Customer  Autom. Fakturierung analysiert und disskutiert |
| 11.08.2015 18:30 – 20:30 | 2 h | Hinweis: Version 5.5. für Signal Keyword notwendig! (Für RaisError notwendig)  Automatisierte Fakturierung fertigestellt mit Gottfried  Log-MessageAdd implementiert |
| 17.08.2015 17:30 – 19:30 | 2 h | Am Berechtigungskonzept weitergearbeitet |
| 18.08.2015 17:00 – 18:00 | 1 h | Dito. |
| 20.08.2015 18:00 – 20:30 | 2.5 h | Änderung an Datenmodell vorgenommen  Weitergearbeitet am Berechtigungskonzept |
| 24.08.2015 18:00 – 20:45 | 2.75 h | Weitergearbeitet am Berechtigungskonzept  Script UserAndGrants.sql erstellt  Neue View v\_FreeNetworkInterfaces erstellt |
| 25.08.2015 18:00 – 19:45 | 1.75 h | Erfassung Beispieldaten  Implementierung Berechtigungskonzept  Hinweis:  Wegen des Konfigurationsfiles wird die Erfassung von mehreren VLANs auf ein Networkinterface applikatorisch verhindert. |
| **Total Stunden** |  |  |

|  |  |  |
| --- | --- | --- |
| **Maurus Kühne** | | |
| **Datum Uhrzeit von / bis** | **Dauer** | **Tätigkeit** |
| 30.07.2015 18:00 – 20:45 | 2.75 h | Tabelle Device geändert: FullDuplex, Speed und IsPhysical auf Nullable gesetzt.  Neues Feld ‚Closed‘: Beschreibt, dass die Rechnung gestellt worden ist.  Automatische Fakturierung: Prozedur P\_Nigthly\_Invoicing erstellt. |
| 06.08.2015 19:30 – 21:00 | 1.5 h | Neues Feld InvoiceTreshold auf Tabelle Customer  Automatische Fakturierung analysiert und diskutiert |
| 11.08.2015 17:00 – 20:00 | 3 h | Automatisierte Fakturierung fertiggestellt mit Gottfried  Manuelle Fakturierung implementiert  POD-Statistik implementiert  Analyse Implementation Produktkatalog begonnen |
| 17.08.2015 17:30 – 19:30 | 2 h | Am Client weitergearbeitet |
| 18.08.2015 17:00 – 18:00 | 1h | Am Client weitergearbeitet |
| 20.08.2015 18:00 – 20:30 | 2.5 h | Am Client weitergearbeitet |
| 24.08.2015 18:00 – 20:45 | 2.75 h | Am Client weitergearbeitet  Bugfix Automatische Fakturierung abgeschlossen  Bugfix Manuelle Fakturierung – Verrechnung von Gutschriften  Anzeige Rechnung in Client implementiert |
| 25.08.2015 18:00 – 19:45 | 1.75 h | Geräte-Konfiguration aus Client am implementieren |
| 03.09.2015 18:00 – 19:15 | 1.25 h | Umsetzung Login-Fenster für MySQL-DB Anmeldung |
| 15.09.2015 17:00 – 20:00 | 2 h | Client-Releasedistribution erstellt  DML-Script erstellt. |
| 17.09.2015 17:30 – 20:00 | 1.5 h | ERM erstellt, Lieferobjekte weiter beschrieben, Korrekturen vorgenommen |

|  |  |  |
| --- | --- | --- |
| **Ruel Holderegger** | | |
| **Datum Uhrzeit von / bis** | **Dauer** | **Tätigkeit** |
| 30.07.2015 18:00 – 20:45 | 2.75 h | Initiale Erstellung Dokumentation  Ausgangslage, Funktionale Anforderungen und Arbeitsjournal erweitert. |
| 06.08.2015 19:30 – 21:30 | 2 h |  |
| 11.08.2015 17:00 – 20:00 | 3 h | Dokumentation erweitert  View v\_UsagePerLocation und View v\_UsagePerPOD erweitert mit Gruppierung nach Geräteart |
| 17.08.2015 17:30 – 19:00 | 1.5 h | Sichtenreferenz in Dokumentation erstellt |
| 18.08.2015 17:00 – 19:00 | 2 h | Rapportierung 17.08.2015  Dokumentation Tabellenreferenz weitergearbeitet |
| 20.08.2015 18:00 – 20:30 | 2.5 h | Modulare Devices werden in unserem Datenmodell absichtlich nicht berücksichtigt.  Auf Tabelle DeviceType wird ein neues Feld ID\_MediumType bereitgestellt.  Einem DeviceType kann genau ein MediumType zugewiesen werden. DeviceType muss mindestens ein MediumType umfassen.  Neues InActive in Tabelle DeviceCategory |
| 21.08.2015 18:00 – 20:30 | 1.5 h |  |
| 24.08.2015 18:00 – 20:45 | 2.75 h | Erstellung Todo-Liste für Projektabschluss  Probleme beim Verbinden auf MySQL Testserver: ab ZbW: Port 465, ab InnoSolv: 3306  **Hinweise:**  Nach dem Laden des des Datenbank-Dumps muss das Berechtigungs-Script manuell ausgeführt werden. |
| 25.08.2015 17:00 – 19:45 | 2.75h | Dokumentation v\_DeviceCatalog angepasst  Dokumentation sämtlicher Tabellen abgeschlossen |
| 03.09.2015 18:00 – 19:15 | 1.25 h | Dokumentation über Umsetzung des Clients vorgenommen |
| 15.09.2015 17:00 – 20:00 | 3 h | DDL-Script erstellt Todo: restliche Views auch noch aufnehmen! |
| 17.09.2015 17:30 – 20:00 |  | Korrekturen an DDL-Script: restliche Views in DDL-Script aufgenommen (inkl. Procs und Functions)  Doku fertigstellt. |
| **Total in Stunden** |  |  |