

SSRS-Training 70-448

Im Februar 2021 gemäß Microsoft-Schulungsunterlagen für die Vorbereitung auf die Zertifizierungs-Prüfung 70-448 vorgesehene, von mir erfolgreich implementierte Beispiel-Projekte.

Übersicht

Übersicht.....	1
Setup Server, Teil 1: Quell-Datenbanken.....	2
Setup IDE.....	2
448-Ch10-Purchasing (Matrix-Bericht, DrillDown, Säulendiagramm, MDX, Action).....	2
448-Ch11-CustomAssembly48Deps	2
448-Ch11-Logistics (Parametrisierung Dataset & Bericht, eigener Code & Assembly).....	3
Setup Server, Teil 2: SSRS	3
448-Ch11-Deployment (via Visual Studio vs. Portal vs. Report Builder).....	3
448-Ch11-ReportViewer-WinForms (eigene .NET-Client-Anwendung entwickeln)	4
448-Ch11-ReportViewer-WebForms (eigene ASP.NET-WebApp entwickeln)	4
448-Ch12-L1-Ex1-Security-item-level-role	4
448-Ch12-L1-Ex2-Security-system-level-role.....	4
448-Ch12-L1-Ex3-Security-authentication-methods	4
448-Ch12-L2-Ex1-Automation-Security.....	5
448-Ch12-L2-Ex2-Automation-shared-Schedule	5
448-Ch12-L2-Ex3-Automation-Subscription.....	5
448-Ch12-L3-Ex1-linked-Report.....	6
448-Ch13-L1-Ex1-Admin-Catalog-DB	6
448-Ch13-L1-Ex2-Admin-Portal-Url.....	6
448-Ch13-L2-Ex1-Admin-Encryption-Key	6

Setup Server, Teil 1: Quell-Datenbanken

- Bestehende Microsoft SQL Server 2017-Installation, Nachinstallation von 3 Quell-Datenbanken:
- SQL-DB: Adventure Works 2008 (OLTP)
- SQL-DB: Adventure Works DW 2008 (OLAP, Star Schema)
- SSAS-DB: TK 448 Mining Models Project (MultiDim)

Setup IDE

- Bestehende Microsoft Visual Studio 2019-Installation, Nachinstallation SSRS-Projektvorlagen.

448-Ch10-Purchasing (Matrix-Bericht, DrillDown, Säulendiagramm, MDX, Action)

- In Visual Studio neues Projekt erstellen mit der Projektvorlage "Berichtsserverprojekt-Assistent".
- Mit dem Assistenten neuen Bericht "**PurchasingSummary**" und neue eingebettete Datenquelle mit Verbindung zur OLTP-Datenbank erstellen.
- Die Authentifizierungsmethode umstellen auf Windows integrierte Sicherheit.
- Eingebettetes Dataset erstellen mit Aggregationsabfrage. SQL-Abfrage-Designer verwenden für Anzeige als relationales Diagramm, als Datenblatt.
- Layouttyp "Matrix" wählen und Felder zuordnen (Seite, Spalten, Zeilen, Details).
- Ergebnis in der Vorschauansicht, insbesondere Paginierung, überprüfen.
- Layoutoptimierungen: Seitenorientierung ändern in Querformat, Spaltenbreiten anpaßen, Zahlenformate setzen, Fußzeile mit User-Loginname und Seite X von Y.
- Gestapeltes Säulendiagramm einfügen. Datenquellen X- und Y-Achse setzen. Haupt- und Achsentitel setzen. Paginierung in Vorschau überprüfen.
- 2 freigegebene Datenquellen neu erstellen, mit Quellen SQL-OLAP-Datenbank und SSAS-Datenbank.
- Neuen Bericht "**TargetMail**" erstellen. Neues eingebettetes Dataset erstellen, das auf freigegebener SQL-OLAP-Datenbank basiert. Die Dataset-Abfrage im SQL-Abfrage-Designer erstellen und Ergebnisdaten anzeigen lassen.
- Zweites eingebettetes Dataset erstellen, das auf freigegebener SSAS-Datenquelle basiert. Im MDX-Abfrage-Designer Abfrage mit drag & drop erstellen und Ergebnisdaten anzeigen lassen.
- Dem bestehenden Bericht "PurchasingSummary" eine berechnete Spalte hinzufügen. Spalten zu 2-stufiger Gruppierung Jahr/ Monat erweitern. Ebene Monat initial ausblenden.
- User-seitigen interaktiven Spalten-Drilldown zulassen.
- Dynamische Liste von Hyperlinks zu Unterberichten erzeugen (Action "Go To Report").
- Im bestehenden Bericht "TargetMail" in einer Matrixzelle die horizontal-lineare Meßinstrument-Visualisierung einfügen und die Formatierungsmöglichkeiten anwenden.
- Unterhalb der bestehenden Matrix eine neue Tabelle mit hardcoded Spaltenanzahl anlegen.
- Diesmal die Zeilenköpfe 2-stufig gruppieren mit 2. Ebene initial ausgeblendet, aber interaktiver User-Drilldown zugelassen.
- Neue berechnete Spalte mit laufendem Saldo mit Hilfe der RunningValue()-Funktion anlegen.

448-Ch11-CustomAssembly48Deps

- In Visual Studio neues Projekt vom Typ "VB.NET Klassenbibliothek (Framework)" anlegen.
- Als .NET-Framework-Version 4.8 wählen. 7 Standard-Verweise "System" belassen.
- 2 Beispiel-Funktionen deklarieren: GetStockColorStatic() und GetStockColorInstance().
- Compilieren und manuell bereitstellen in den SSRS-spezifischen Verzeichnissen auf dem Entwicklungsrechner und auf dem SSRS-Server.

448-Ch11-Logistics (Parametrisierung Dataset & Bericht, eigener Code & Assembly)

In Visual Studio neues SSRS-Projekt mit freigegebener Datenquelle zur OLTP-Datenbank erstellen.

- Neuem Bericht "**Tracking Numbers by Month**" und eingebettetes Dataset mit SQL-Abfrage erstellen. Tabelle mit 4 Spalten erstellen, Zeilen nach 1. Spalte gruppieren. Ergebnis 2.477 Seiten.
- Im Dataset die SQL-Abfrage um eine WHERE-Klausel mit 2 Parametern erweitern.
- Beide Parameter an an die Felder Jahr und Monat binden.
- In der Vorschau testweise die Parameterwerte 2004 und 01 eingeben. Ergebnis nur noch 3 Seiten.
- 2 eingebettete Datasets mit SQL-Abfragen für die Auswahl Jahr und Monat einfügen, wobei die Auswahl Monat parametrisch nach Jahr gefiltert wird.
- Bestehenden 2 Berichts-Parameter Jahr und Monat ändern auf vordefinierte Auswahllisten.
- Wobei Auswahl Monat auf 2 Spalten Auswahltitel (Monatsname) und Filterwert (MonatsNr) basiert.
- Ergebnis Auswahl Monat ist gesperrt, solange User nicht den Parameter Jahr gewählt hat.
- Neuen Bericht "**Finished Products Stock Level**" erstellen. Auf freigegebener OLTP-Datenquelle basierendes eingebettetes Dataset erstellen, mit Abfrage von 7 Spalten der Produkt-Tabelle, gefiltert nach Status "finished".
- Tabelle mit 4 Spalten einfügen.
- Für Spalte "Name" bedingte Formatierung definieren, basierend auf a) Ausdruck.
- Im Bericht eingebetteten VB.NET-Code anlegen mit eigener Funktion GetStockColor().
- Für Spalte "Stock Level" bedingte Formatierung definieren, basierend auf b) eingebettetem Code.
- Im Bericht Verweis auf eigene Assembly (siehe oben) registrieren. Objekt-Instanz benamen.
- Für 3. und 4. Spalte bedingte Formatierung definieren, basierend auf c) statische und d) dynamische Assembly-Methode.

Setup Server, Teil 2: SSRS

- Bestehende Microsoft Sql Server 2017-Installation, Nachinstallation SSRS. Vorerst nur minimal erforderliche Konfigurationsschritte durchführen:
- Konfiguration Firewall inbound zusätzlichen Port 80 für SSRS-Web-Dienst und -Portal öffnen.
- Im SSRS-Konfigurations-Manager:
- Den Web-Dienst aktivieren.
- Im SQL-Modul neue Katalog-Datenbank anlegen und SSRS-Zugriff konfigurieren.
- Das Web-Portal aktivieren.
- Katalog-Verschlüsselungs-Schlüssel Backupdatei erstellen und auf Entwicklungsrechner sichern.

448-Ch11-Deployment (via Visual Studio vs. Portal vs. Report Builder)

- In Visual Studio neues SSRS-Projekt mit freigegebener Datenquelle zur SQL-OLAP-Datenbank erstellen.
- Neuen Bericht "**Customer Income in Number of Cars Owned Groups**" mit Assistenten erstellen.
- Eingebettetes Dataset erstellen, das auf der freigegebenen OLAP-Datenquelle basiert, mit Abfrage von 3 Spalten.
- Tabelle erstellen, Zeilen gruppiert, mit eingerückter Formatierung, Zwischensummen einblenden, User-Drilldown erlauben.
- In den Berichts-Eigenschaften, im Profil "Release", die Bereitstellungseigenschaften konfigurieren, insbesondere die Ziel-Url des SSRS-Servers hinterlegen.
- Bereitstellungsmethode a): In Visual Studio den Befehl für die Projekt-Bereitstellung aufrufen.
- Die Protokollierung überprüfen, ob die Bereitstellung fehlerfrei erfolgte.
- Im Web-Portal prüfen, ob im Unterverzeichnis "Datenquellen" die freigegebene OLAP-Datenquelle angezeigt wird. Im Unterverzeichnis des Projektes prüfen, ob der Bericht korrekt angezeigt wird.
- Exkurs: Die Berichts-Datei unter neuem Namen klonen. Hierzu in einem Texteditor low-level die alte RDL-Content-Struktur mit einer neuen Berichts-ID kombinieren.
- Spalte neu einfügen mit Durchschnitt-Aggregatfunktion, Zahlenformatierung Währung.
- Bereitstellungsmethode b): Im Browser das Web-Portal öffnen, zum Projekt-Verzeichnis navigieren und die rdl-Datei aus dem Visual Studio-Projekt-Verzeichnis hochladen.
- Ergebnis ist zunächst keine Fehlermeldung, der hochgeladene Bericht wird im Verzeichnis korrekt angezeigt. Beim Öffnen des Berichtes jedoch erscheint die Fehlermeldung, daß die angegebene Datenquelle nicht gefunden wurde.

- Exkurs: Zur Fehlerlokalisierung im Berichts-Generator ("Report Builder") einen alten funktionierenden Bericht herunterladen. In einem Texteditor die XML-Strukturen low-level vergleichen. Ergebnis: In der heruntergeladenen rdl-Datei wird die Datenquelle qualifiziert angegeben, in der Visual Studio-rdl-Datei dagegen unqualifiziert (Qualifizierung erfolgt vermutlich erst später im Bereitstellungsprozeß).
- Fazit für die Zukunft: Will man eine VS-rdl-Datei über das Web-Portal hochladen, muß man die Angabe der freigegebenen Datenquelle immer erst noch low-level qualifizieren.
- Bereitstellungsmehtode c): Im Web-Portal, Menüpunkt "Hinzufügen - Paginierter Bericht" aufrufen. Im Berichts-Generator die bestehende freigegebene SQL-OLAP-Datenquelle wählen, den Layouttyp "Tabelle" wählen, 3 Felder auswählen und anordnen. Teilergebnisse einblenden, User-Drilldown erlauben.
- Im Berichts-Generator Vorschau prüfen und interaktiven Drilldown testen.
- Im Berichts-Generator erstellten Bericht im Portal speichern bzw. hochladen. Bericht im Portal prüfen.
- Exkurs: Die mit dem Berichts-Generator erstellte rdl-Datei in Visual Studio hinzufügen. Öffnen in Entwurfsansicht ohne Fehlermeldung. Wechsel in Vorschau führt zu Fehlermeldung Datenquelle.
- Fazit: Die rdl-Dateien von Visual Studio und Berichts-Generator sind in beide Richtung inkompatibel und müssen jedesmal erst low-level angepaßt werden.

448-Ch11-ReportViewer-WinForms (eigene .NET-Client-Anwendung entwickeln)

- In Visual Studio neues Projekt vom Typ "VB.NET Windows Forms-App (Framework)" anlegen.
- Als.NET-Framework-Version 4.8 wählen. 7 Standard-Verweise "System" belassen.
- NuGet-Paket für SSRS-ReportViewer für Windows Forms nachinstallieren.
- Haupt-Formular erweitern um Dienst-Verweis, Web-Verweis mit SOAP-Url des SSRS-Servers, Dropdown-Menü mit dynamischer Auswahlliste der Berichte, ReportViewer-Steuerelement.
- CodeBehind ausfüllen für Ereignisse Laden = Menü aufbauen und Menü-Klick = ReportViewer-Inhalt aktualisieren.
- Project compilieren und im Debug-Modus starten. Sämtliche Funktionen des ReportViewers testen (Paging, Drilldown, Parameterfelder, Layoutvorschau, Zoom, Exportformate, Drucken, Suche).

448-Ch11-ReportViewer-WebForms (eigene ASP.NET-WebApp entwickeln)

- In Visual Studio neues Projekt vom Typ "C# ASP.NET Web-App (Framework)" anlegen.
- Als.NET-Framework-Version 4.8 wählen. 7 Standard-Verweise "System" belassen.
- NuGet-Paket für SSRS-ReportViewer für Web Forms nachinstallieren.
- WebForm-Seite neu erstellen, als Startseite setzen, Seitenüberschrift einfügen, ASP-ScriptManager-Steuerelement einfügen, ReportViewer-Steuerelement einfügen und konfigurieren mit SOAP-Url des SSRS-Servers und dem Pfad zu einem ersten Beispielbericht.
- Project compilieren und Debugging starten. Im Browser sämtliche Funktionen des ReportViewers testen (Paging, Drilldown, Parameterfelder, Zoom, Export, Drucken, Suche. Fehlt: Layoutvorschau).

448-Ch12-L1-Ex1-Security-item-level-role

- In Sql Server Management Studio (SSMS) mit SSRS-Instanz verbinden.
- Eine neue item-level-Rollendefinition anlegen, 2 aus 18 Einzelrechten selektieren und speichern.
- Im Web-Portal im Stamm-Verzeichnis der bestehenden Benutzergruppe "VORDEFINIERT\Benutzer" diese neue Rolle zuordnen.
- In einem der Unterverzeichnisse die erfolgreiche automatische Vererbung überprüfen.

448-Ch12-L1-Ex2-Security-system-level-role

- In SSMS eine neue system-level-Rollendefinition anlegen, 2 aus 9 Einzelrechten selektieren und speichern.
- Im Web-Portal in den Site-Einstellungen der bestehenden Benutzergruppe "VORDEFINIERT\ Benutzer" diese neue Rolle zuordnen.

448-Ch12-L1-Ex3-Security-authentication-methods

- Im Web-Portal zu einem bestehenden Bericht die Eigenschaften anzeigen lassen, insbesondere die zugeordnete freigegebene Datenquelle. Die Eigenschaften der Datenquelle anzeigen lassen.
- Die Authentifizierungsmethode ändern von a) "Windows integrierte Sicherheit" auf b) "Benutzereingabe - Windows-Benutzername und -Kennwort".

- ba) Bericht aufrufen und testweise ungültige Zugangsdaten eingeben. Ergebnis: Fehlermeldung "rsLogonFailed".
- bb) Bericht aufrufen und eigene gültige Zugangsdaten eingeben. Ergebnis korrekte Berichtsanzeige.
- Die Authentifizierungsmethode ändern auf c) hardcoded Windows-Zugangsdaten und die eigenen Zugangsdaten eingeben und speichern.
- Bericht aufrufen. Ergebnis: Ohne das Zugangsdaten abgefragt werden, wird der Bericht angezeigt.

448-Ch12-L2-Ex1-Automation-Security

- In der Sicherheitsinfrastruktur die Voraussetzungen für eine Automatisierung schaffen:
- In SSMS zu SQL-Instanz verbinden. Skript ausführen, das 1 neues Datenbank-Login-Account sowie für die 3 Datenbanken OLTP, msdb und die Katalog-Datenbank je ein neues User-Account erstellt, dem je 1 Datenbank-Rolle zugeordnet wird (datareader/ datawriter).
- Auf dem SSRS-Server in Windows ein neues Benutzer-Konto " ReportExecutionSSRS" für die spätere automatisierte Berichtsausführung erstellen. Im Dateisystem ein neues Ziel-Verzeichnis für die späteren pdf-Berichtsdateien erstellen, dieses Verzeichnis freigeben für das eben erstellte neue Benutzer-Konto (read/ write) und im Dateisystem NTFS-Vollzugriff gewähren.
- Im Web-Portal in der bestehenden freigegebenen Datenquelle die Authentifizierungsmethode umstellen auf hardcoded Datenbank-Zugriffsdaten und diese eingeben. Beide werden im SSRS-Katalog verschlüsselt gespeichert.

448-Ch12-L2-Ex2-Automation-shared-Schedule

- In SSMS eine neue freigegebene Zeitplan-Definition erstellen, Typ "einmalig" mit fixer Start-Uhrzeit.
- Im Web-Portal in den Site-Einstellungen - Zeitpläne die Anzeige des neuen Eintrags prüfen.

448-Ch12-L2-Ex3-Automation-Subscription

- Im Web-Portal zu einem Beispiel-Bericht ein neues Abonnement anlegen, daß den freigegebenen Zeitplan verwendet und als Ziel das oben eingerichtete Freigabeverzeichnis hat, Exportformat Excel.
- Nach Ablauf der Startzeit die produzierte Exceldatei im Zielverzeichnis überprüfen.
- Das Abonnement vorsichtshalber deaktivieren.
- Automatisierte Massenproduktion: Zum gleichen Bericht neues Abonnement anlegen, Typ "datengesteuert", Abfrage für Steuertabelle mit 97 Einträgen hinterlegen, gleiches Zielverzeichnis, Exportformat pdf, einmalig laufen lassen. Ergebnis 97 pdf-Dateien prüfen.

448- Ch12-L3-Ex1-linked-Report

- Im Web-Portal zu einem Beispiel-Bericht den Menüpunkt "Verknüpften Bericht erstellen" aufrufen.
- Im verknüpften Bericht 2 Parameter abweichend setzen und ausblenden. 30-min-Caching aktivieren. Zeitplan-gesteuerte automatische nächtliche Snapshots erstellen lassen.
- Verknüpften Bericht anzeigen lassen. Ergebnis: 2 Parameterfelder sind ausgeblendet, die neuen Parameterwerte sind wirksam.
- Zum Vergleich den Original-Bericht anzeigen lassen. Ergebnis: Parameterfelder und deren Vorbelegung sind unverändert.
- Testweise den Original-Bericht löschen. Ergebnis: Der verknüpfte Bericht bleibt erhalten, beim Anzeigen erscheint jedoch die Fehlermeldung "rsInvalidReportLink".

448-Ch13-L1-Ex1-Admin-Catalog-DB

- Im Konfigurations-Manager testweise eine neue Katalog-Datenbank erstellen.
- Im Web-Portal Ergebnis überprüfen: Stamm-Verzeichnis ist leer.
- Im Konfigurations-Manager wieder auf die alte Katalog-Datenbank umstellen.
- Im Web-Portal Ergebnis überprüfen: Der alte Inhalt ist wieder vollständig zugänglich.

448-Ch13-L1-Ex2-Admin-Portal-Url

- Im Konfigurations-Manager testweise das virtuelle Verzeichnis für das Web-Portal umbenennen.
- Im Browser alte Portal-Url aufrufen. Ergebnis: Fehlermeldung HTTP#404.
- Im Browser neue Portal Url aufrufen. Ergebnis alter Inhalt wieder vollständig zugänglich.
- Im Konfigurations-Manager wieder alten Namen setzen.
- Im Browser neue und alte Portal-Url aufrufen. Ergebnis HTTP#404 bzw. alter Inhalt.

448-Ch13-L2-Ex1-Admin-Encryption-Key

- Auf dem SSRS-Server mit dem CLI-Tool "RSKeyMgmt" eine Sicherungskopie des aktuellen Schlüssels zur Verschlüsselung der Katalog-Datenbank erstellen.
- Bei dem gleichen Tool den substitute-Befehl aufrufen. Das Tool generiert einen neuen Schlüssel und schlüsselt mit diesem den Inhalt der Katalog-Datenbank um. Der neue Schlüssel ist noch nicht im SSRS-Dienst wirksam. Ergebnis: Im Portal kommt es zur Fehlermeldung "Integrität".
- Im Konfigurations-Manager den SSRS-Dienst beenden und neu starten. Ergebnis: Im Portal wird der Inhalt wieder korrekt angezeigt.
- Mit dem CLI-Tool auch von dem neuen Schlüssel eine Sicherungskopie erstellen.
- Mit dem CLI-Tool den alten Schlüssel wiederherstellen und den SSRS-Dienst neu starten. Ergebnis: Im Portal kommt es wieder zur Fehlermeldung "Integrität" (weil der Dienst versucht mit dem alten Schlüssel auf den neu verschlüsselten Katalog zuzugreifen).
- Mit dem CLI-Tool den neuen Schlüssel wiederherstellen und den SSRS-Dienst neu starten. Ergebnis: Im Portal wird der ursprüngliche Inhalt wieder korrekt angezeigt.