

# Ausarbeitung des Assignment

für die Prüfung des Moduls Datenbanken 1 + 2 (W3WI\_105)

Dozent und Prüfer: Herr Markus Trefzer

Duale Hochschule Baden-Württemberg Lörrach

erarbeitet von:

Anna-Lehna Grittke	Matr. 8322504
Chiara Facciola	Matr. 4320222
Moritz Höft	Matr. 2441453
Jacob Ruhnau	Matr. 9732319

*Kurs: WWI22A*

Vorgelegt am 15. Dezember 2023

## Inhalt

Abbildungsverzeichnis .....	3
Bearbeitung der Aufgaben .....	4
Aufgabe 1 .....	4
Aufgabe 2 .....	5
Aufgabe 3 .....	7
Aufgabe 4 .....	8
Aufgabe 5 .....	9
Aufgabe 6 .....	12
Anhang .....	14
Quellen .....	14
Analgenverzeichnis .....	14
Aufgabenstellungen .....	15

# Abbildungsverzeichnis

Abbildung 1: Angepasstes Schema der Tabellen Struktur

Abbildung 2: Error im exemplarischen Trigger für INSERT auf SUPPORT

Abbildung 3: SUPPORT Tabelle gefüllt mit Testdatensätzen

Abbildung 4: SSIS Paket anlegen

Abbildung 5: SSIS Paket sichern

Abbildung 6: Zeitplan für SSIS Paket anlegen

Abbildung 7: Darstellung der räumlichen Trennung von Backups

## Bearbeitung der Aufgaben

### Aufgabe 1

#### Problemstellung

Damit alle 3 Bereiche in einer Datenbank abgebildet werden können, ohne dass Tickets von anderen Bereichen aufgerufen werden können, müssen einige Veränderungen auf Datensatzebene vorgenommen werden.

#### Problemlösung

Das nachfolgende Bild zeigt das bereits angepasste Schema der Tabellen Struktur.

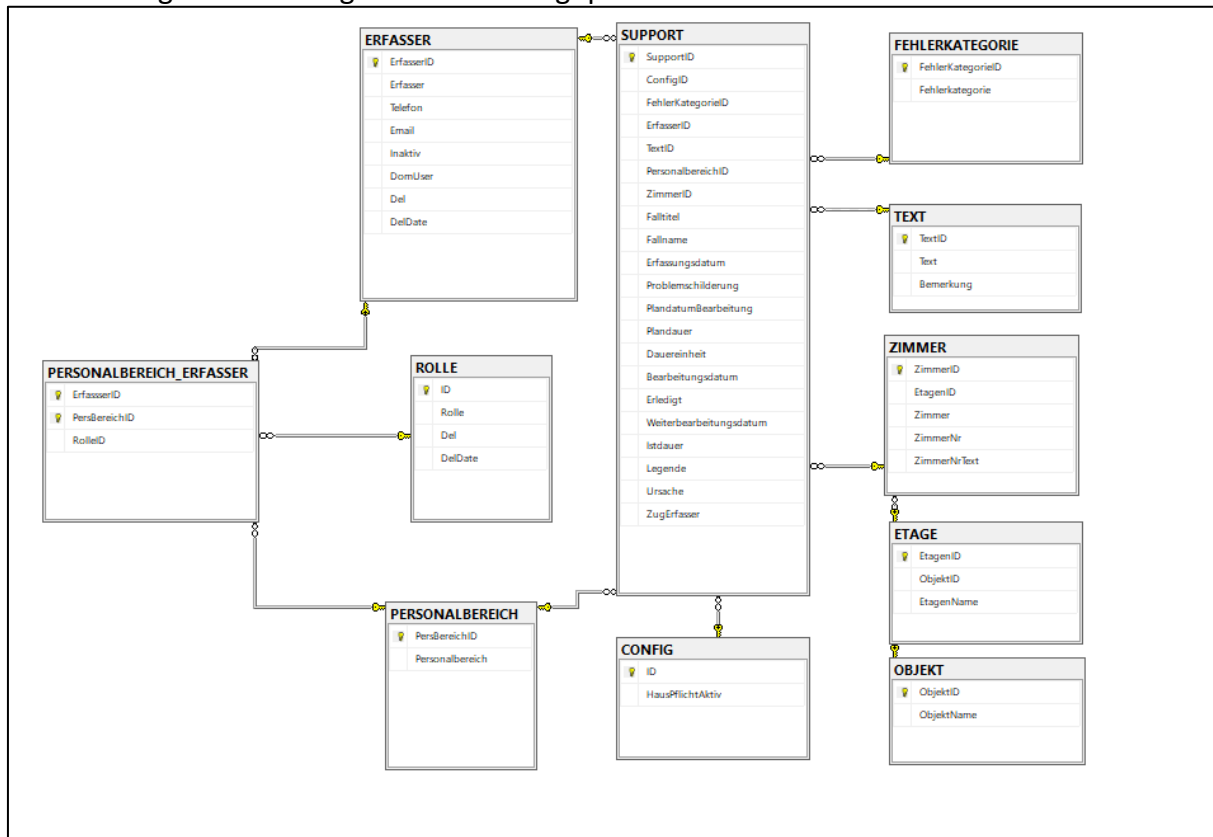


Abbildung 1: Angepasstes Schema der Tabellen Struktur

Die Tabelle **CONFIG** wurde übernommen, jedoch mit der zusätzlichen Anforderung, dass die Spalte **HausPflichtAktiv** nun als Pflichtfeld deklariert ist und nun als Fremdschlüssel für **SUPPORT**.

Die Referenzen aus **SUPPORT** zu **ETAGE** und **OBJEKT** wurden entfernt, um Redundanzen zu vermeiden. Die Information über die Etage und das Objekt kann nun über die Tabelle **ZIMMER** abgeleitet werden, was zu einer optimierten Struktur führt.

Neu ist die Tabelle **PERSONALBEREICH\_ERFASSER**, um Relationen zwischen Erfassern, Personalbereichen und dortigen Rollen zu speichern. Dies ermöglicht so die dynamische Berechtigungszuweisung der User für verschiedene Personalbereiche mit unterschiedlichen Rollen.

Außerdem wurden neue Relationen und Fremdschlüssel zu den Tabellen und **TEXT** **PERSONALBEREICH\_ERFASSER** eingeführt.

Die erstellte Tabellwertfunktion **GetSupportByPersonalbereichID** (vgl. Anlage 1) ermöglicht es, Supportfälle basierend auf einer bestimmten Personalbereichs-ID abzurufen.

Die Funktion akzeptiert einen Parameter, nämlich die Personalbereichs-ID (@PersonalbereichID), und gibt eine Ergebnistabelle zurück.

Die Resultate umfassen relevante Informationen zu den Supportfällen, diese Werte könnten darüber hinaus um weitere ergänzt werden.

Um diese Daten zu erhalten, werden verschiedene Joins zwischen den Tabellen durchgeführt. Die JOIN-Operationen verknüpfen die **SUPPORT**-Tabelle mit anderen relevanten Tabellen wie **CONFIG**, **FEHLERKATEGORIE**, **ERFASSER**, **TEXT**, **PERSONALBEREICH**, **ZIMMER**, **ETAGE** und **OBJEKT** über die Fremdschlüssel, um eine umfassende Sicht auf die Supportfälle in Bezug auf den spezifizierten Personalbereich zu gewährleisten.

Durch die Erstellung dieser Tabellenwertfunktion wird die Abfrage und Analyse von Supportfällen erleichtert, insbesondere wenn der Fokus auf einem bestimmten Personalbereich liegt.

### Anpassung im Bestandssystem

Die Datensätze der jeweiligen Bereiche können über die Tabellenwertfunktion abgerufen werden. Dies kann über `SELECT * FROM dbo.GetSupportByPersonalbereichID(1);` erfolgen.

### Ergänzung

Es könnten alternativ hierzu Views generiert werden, welche den aktuellen Entwicklungen der Bestandssysteme spiegeln und so auf Applikationsebene vermutlich keine Anpassung bedürften.

## Aufgabe 2

### Problemstellung

Da teilweise die gleichen Mitarbeiter in den einzelnen Bereichen sind musste ein Trigger implementiert werden, damit der einzelne User in der Tabelle `Erfasser` nicht mehrmals abgebildet werden muss.

### Problemlösung

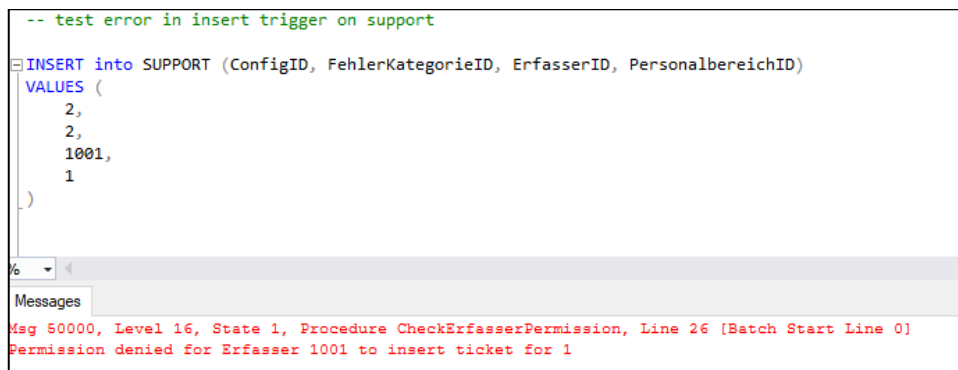
Dieser Trigger (vgl. Anlage 2) wird vor dem Einfügen von Daten in die Tabelle **SUPPORT** ausgelöst. Sein Hauptzweck besteht darin sicherzustellen, dass der Erfasser, der einen Supportfall einreicht, die entsprechende Berechtigung für den angegebenen Personalbereich hat. Dieser Trigger berücksichtigt die Anforderung, dass Benutzer in verschiedenen Bereichen tätig sein können, und stellt sicher, dass ein Benutzer nur Tickets für die Bereiche einreichen kann, für die er autorisiert ist.

Der Trigger ist wie folgt implementiert

Zuerst wird geprüft, ob der Erfasser die Berechtigung für den angegebenen Personalbereich hat in **PERSONALBEREICH\_ERFASSER** besitzt.

- Wenn die Berechtigung nicht vorhanden ist, wird die **INSERT**-Operation abgebrochen, und es wird eine Fehlermeldung generiert. Die Fehlermeldung gibt an, welcher Erfasser versucht hat, ein Ticket für einen bestimmten Personalbereich einzufügen, für den er keine Berechtigung hat (s. Abbildung).

- Wenn die Berechtigung vorhanden ist, wird die **INSERT**-Operation durchgeführt. Es werden alle relevanten Spalten aus der temporären **inserted**-Tabelle verwendet, welche die Daten für die einzufügenden Zeilen enthält.



```
-- test error in insert trigger on support
INSERT into SUPPORT (ConfigID, FehlerKategorieID, ErfasserID, PersonalbereichID)
VALUES (
    2,
    2,
    1001,
    1
)
```

Messages

Msg 50000, Level 16, State 1, Procedure CheckErfasserPermission, Line 26 [Batch Start Line 0]  
Permission denied for Erfasser 1001 to insert ticket for 1

Abbildung 2: Error im exemplarischen Trigger für INSERT auf SUPPORT

Diese Trigger-Logik hilft sicherzustellen, dass nur autorisierte Erfasser Supporttickets für bestimmte Personalbereiche einreichen können. Dadurch wird die Einhaltung von Berechtigungsregeln gewährleistet und unerwünschte Einfügungen von Daten in die **SUPPORT**-Tabelle verhindert. Diese Funktionalität kann auf weitere Statements der DML zugeschnitten werden.

### Anpassungen im Bestandssystem

In den Bestandssystemen muss nun sichergestellt sein, dass die User in der **PERSONALBEREICH\_ERFASSER** Tabelle pro User und Personalbereich ein Datensatz angelegt werden, welcher die entsprechende Rolle in dem System vergibt. Hierbei sind außerdem die möglichen Rollen in der **ROLLEN** Tabelle zu beachten.

## Aufgabe 3

Der Code zu der Aufgabe befindet sich in Anlage 4: "TestData.sql"

### Erklärung des SQL-Skripts

Das Ziel des SQL-Skriptes ist es, eine große Menge von Testdatensätzen in die `SUPPORT`-Tabelle der Datenbank einzufügen und somit die Logik aus Aufgabe 2 zu überprüfen. Dies beginnt mit der Initialisierung und Konfiguration, bei der die Anzeige der Anzahl betroffener Zeilen durch SET NOCOUNT ON; unterdrückt wird, um die Performance und Übersichtlichkeit bei der Ausführung zu verbessern. Ein Zähler @Counter wird eingeführt und auf 1 gesetzt, um eine Schleife zu steuern, die 25.000-mal durchlaufen wird.

Innerhalb der Schleife wird das INSERT INTO-Statement verwendet, um jeweils einen Datensatz in die Tabelle einzufügen. Die Werte für die verschiedenen Spalten der Tabelle werden teils statisch, teils dynamisch generiert. Die dynamische Generierung erfolgt durch Funktionen wie ABS, CHECKSUM, NEWID(), und DATEADD, um Zufälligkeit und Vielfalt zu garantieren. So werden beispielsweise zufällige Werte für ConfigID zwischen 1 und 2 oder zufällige Erfassungsdaten innerhalb des letzten Jahres erstellt. Zusätzlich wird der Wert von @Counter in einige Felder integriert, um Bezeichnungen wie Falltitel1, Falltitel2 usw. zu erzeugen, was dazu beiträgt, dass jeder Datensatz in der Tabelle einzigartig ist.

Nach jedem Durchlauf der Schleife wird @Counter um 1 erhöht, bis die Schleife 25.000 Mal durchlaufen wurde und somit 25.000 einzigartige Datensätze in die Tabelle eingefügt wurden.

Das nachfolgende Bild zeigt den Aufbau der SUPPORT Tabelle, welche mit Testdatensätzen gefüllt wird:

SupportID	ConfigID	Fehler...	ErfasserID	TextID	Person...	Zm...	Falltitel	Falldate	Erfassungs...	Problemschil...	PlandatumBear...	Pl...	Dauer...	Bearbeitungs...	Erf...	Weiterbear...	Ist...	Legende	Ursache	ZugErfasser
95018	2	12	1004	1	1	3	Falltitel1	Falldate1	2023-05-29...	Problemschil...	2023-12-27 21...	6	Stun...	2024-03-30 2...	1	2024-01-0...	7	Legende 1	Ursache 1	3
95019	1	11	1006	2	7	1	Falltitel2	Falldate2	2023-01-07...	Problemschil...	2023-12-28 21...	2	Stun...	2024-04-03 2...	1	2023-12-1...	3	Legende 2	Ursache 2	1
95020	2	17	1007	2	1	1	Falltitel3	Falldate3	2023-05-02...	Problemschil...	2024-01-09 21...	7	Stun...	2024-02-07 2...	0	2024-01-0...	9	Legende 3	Ursache 3	4
95021	1	17	1006	2	1	5	Falltitel4	Falldate4	2023-09-15...	Problemschil...	2023-12-29 21...	1	Stun...	2024-11-12 2...	0	2024-01-0...	0	Legende 4	Ursache 4	1
95022	2	6	1006	1	1	7	Falltitel5	Falldate5	2023-04-01...	Problemschil...	2024-01-03 21...	6	Stun...	2024-08-08 2...	1	2023-12-1...	0	Legende 5	Ursache 5	5
95023	2	9	1004	1	8	2	Falltitel6	Falldate6	2023-03-01...	Problemschil...	2024-01-02 21...	9	Stun...	2024-06-24 2...	0	2024-01-1...	2	Legende 6	Ursache 6	4
95024	2	5	1005	1	8	4	Falltitel7	Falldate7	2023-04-12...	Problemschil...	2023-12-17 21...	4	Stun...	2024-04-19 2...	0	2023-12-1...	8	Legende 7	Ursache 7	2
95025	2	7	1004	1	8	3	Falltitel8	Falldate8	2023-12-03...	Problemschil...	2023-12-19 21...	5	Stun...	2024-10-26 2...	1	2023-12-2...	5	Legende 8	Ursache 8	3
95026	2	17	1004	2	8	2	Falltitel9	Falldate9	2023-04-18...	Problemschil...	2023-12-20 21...	2	Stun...	2023-12-28 2...	1	2023-12-2...	7	Legende 9	Ursache 9	1
95027	1	9	1005	1	8	1	Falltitel10	Falldate10	2022-12-27...	Problemschil...	2024-01-11 21...	10	Stun...	2024-10-23 2...	1	2024-01-0...	1	Legende 10	Ursache 10	2
95028	1	4	1004	1	8	4	Falltitel11	Falldate11	2023-03-18...	Problemschil...	2024-01-06 21...	2	Stun...	2024-02-06 2...	1	2023-12-1...	4	Legende 11	Ursache 11	5
95029	1	19	1005	1	8	2	Falltitel12	Falldate12	2023-03-01...	Problemschil...	2023-12-19 21...	5	Stun...	2024-06-20 2...	0	2024-01-0...	0	Legende 12	Ursache 12	2
95030	1	17	1004	1	8	6	Falltitel13	Falldate13	2023-09-14...	Problemschil...	2023-12-27 21...	3	Stun...	2024-03-28 2...	1	2023-12-2...	9	Legende 13	Ursache 13	2
95031	2	10	1005	1	1	2	Falltitel14	Falldate14	2023-10-21...	Problemschil...	2024-01-05 21...	7	Stun...	2024-04-03 2...	1	2023-12-1...	6	Legende 14	Ursache 14	2
95032	1	3	1006	1	8	5	Falltitel15	Falldate15	2023-03-25...	Problemschil...	2024-01-06 21...	3	Stun...	2024-06-30 2...	0	2024-01-1...	1	Legende 15	Ursache 15	3
95033	2	17	1006	1	1	3	Falltitel16	Falldate16	2023-02-07...	Problemschil...	2023-12-19 21...	5	Stun...	2024-03-26 2...	0	2023-12-1...	2	Legende 16	Ursache 16	1
95034	2	4	1004	2	1	6	Falltitel17	Falldate17	2023-10-30...	Problemschil...	2024-01-02 21...	5	Stun...	2024-10-25 2...	1	2023-12-2...	6	Legende 17	Ursache 17	1
95035	1	13	1005	1	7	4	Falltitel18	Falldate18	2023-06-08...	Problemschil...	2024-01-01 21...	10	Stun...	2024-10-03 2...	0	2024-01-0...	4	Legende 18	Ursache 18	1
95036	1	17	1007	2	8	3	Falltitel19	Falldate19	2023-10-17...	Problemschil...	2023-12-22 21...	1	Stun...	2024-04-07 2...	1	2023-12-3...	9	Legende 19	Ursache 19	5
95037	1	4	1007	1	7	2	Falltitel20	Falldate20	2023-03-31...	Problemschil...	2024-01-02 21...	8	Stun...	2024-05-09 2...	1	2023-12-3...	5	Legende 20	Ursache 20	1
95038	2	12	1006	1	1	4	Falltitel21	Falldate21	2023-08-01...	Problemschil...	2023-12-31 21...	8	Stun...	2024-10-04 2...	1	2024-01-0...	8	Legende 21	Ursache 21	1
95039	1	7	1006	2	8	3	Falltitel22	Falldate22	2023-01-14...	Problemschil...	2024-01-04 21...	6	Stun...	2024-11-28 2...	1	2024-01-0...	7	Legende 22	Ursache 22	5
95040	1	20	1007	1	8	6	Falltitel23	Falldate23	2023-02-23...	Problemschil...	2023-12-24 21...	6	Stun...	2024-02-26 2...	0	2023-12-1...	3	Legende 23	Ursache 23	1
95041	1	1	1006	1	8	3	Falltitel24	Falldate24	2023-09-26...	Problemschil...	2024-01-01 21...	8	Stun...	2024-10-01 2...	1	2023-12-2...	5	Legende 24	Ursache 24	1
95042	2	1	1004	1	7	4	Falltitel25	Falldate25	2023-08-07...	Problemschil...	2023-12-21 21...	4	Stun...	2024-04-15 2...	0	2024-01-0...	2	Legende 25	Ursache 25	2
95043	1	1	1004	1	8	3	Falltitel26	Falldate26	2023-03-26...	Problemschil...	2024-01-12 21...	8	Stun...	2024-05-05 2...	0	2024-01-0...	9	Legende 26	Ursache 26	5

Abbildung 3: SUPPORT Tabelle gefüllt mit Testdatensätzen

### Ergänzung:

Um die Applikation vollständig zu testen, dürften sich die Testdaten nicht nur auf die SUPPORT Tabelle beschränken. Jegliche Fremdschlüssel in der Tabelle wie u.a. ErfasserID müssten nach der implementierten Logik auch beachtet werden, um die Datenqualität zu überprüfen.

## Aufgabe 4

Eine Möglichkeit eine CSV-Datei automatisiert zu exportieren bietet die SQL Server Integration Services (SSIS) von Microsoft. Um den Anforderungen zu entsprechen, wird zunächst ein SSIS Paket erstellt und daraufhin die Ausführung geplant.

### 1. Erstellung eines SSIS Pakets

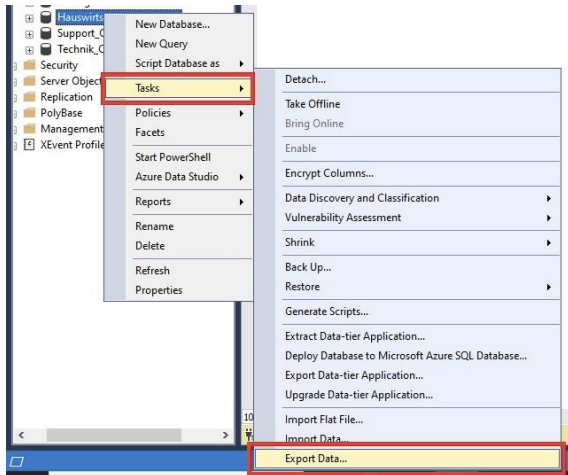


Abbildung 4: SSIS Paket anlegen

Das Paket kann erstellt werden, mit einem Rechtsklick auf der gewünschten Datenbank. Hier wird „Tasks“ ausgewählt und danach zu „Export Data...“ navigiert. Hier wird ein Pop-Up „SQL Server Import and Export Wizard“ geöffnet.

In den folgenden Pop-Ups werden die gewünschte Datenquelle, Zielort der Daten und die zu exportierenden Daten ausgewählt und jeweils mit „Weiter“ bestätigt.

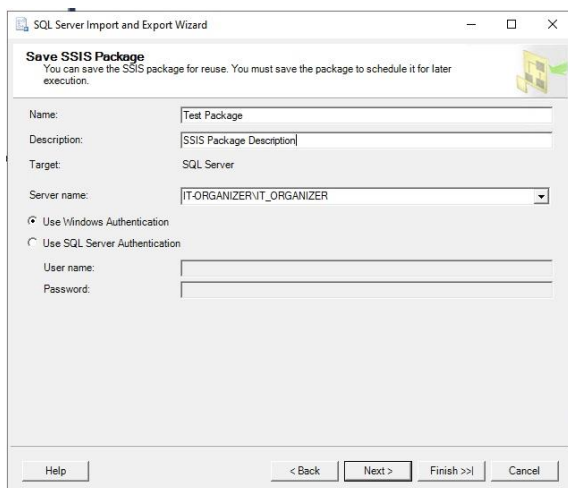


Abbildung 5: SSIS Paket sichern



Im nächsten Schritt kommen die Auswahlmöglichkeiten „Run immediately“ und „Save SSIS Package“. Hier wird die zweite Option „Save SSIS Package“ ausgewählt, um ein Paket zu erstellen. Mit einem Klick auf „Finish“ im unteren Bereich des Pop-Ups ist das Paket erstellt und gespeichert.

## 2. Erstellung eines Zeitplans

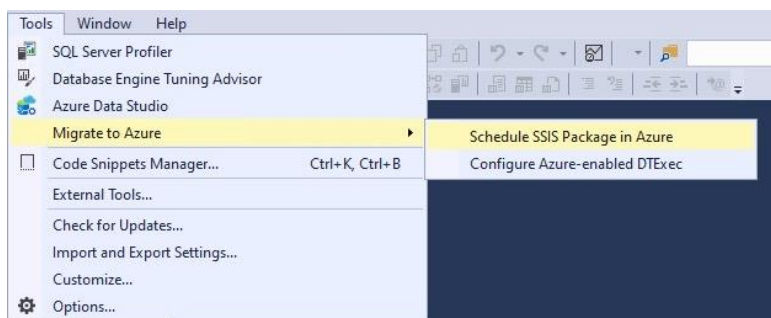


Abbildung 6: Zeitplan für SSIS Paket anlegen

Als nächstes wird die Ausführung des Paketes geplant. Dies wird ebenfalls im SQL Server Management Studio (SSMS) erstellt. Das erstellte Paket kann im Object-Explorer gefunden werden. Mit einem Klick auf die rechte Maustaste wird hier „Schedule...“ ausgewählt. Nun öffnet sich ein neues Pop-Up, in welchem die Ausführung geplant werden kann. Auf der linken Seite des Feldes, gibt es drei verschiedenen Seiten: „General“, „Package“ und „Schedule“.

Auf der Seite „General“ wird der Name und die Beschreibung des Auftrages definiert. Auf der nächsten Seite „Package“ kann die optionale-Laufzeiteinstellung und die Laufzeit ausgewählt werden. Auf der letzten Seite „Schedule“ kann nun die Häufigkeit, die Tageszeit und die Dauer des Auftrages eingestellt werden. Im nächsten Schritt muss in einem neuen Pop-Up bestätigt werden, dass eine neue Data Factory-pipeline in Azure Data erstellt wird. Hier kann der Zeitplan überprüft und angepasst werden.

## Aufgabe 5

Im sensiblen Kontext eines Krankenhausumfelds nimmt die Sicherheit und Integrität von Daten eine entscheidende Rolle ein. Um die ständige Verfügbarkeit zu gewährleisten und potenziellen Datenverlust zu vermeiden, wird die Implementierung eines effizienten externen Backupsystems unerlässlich. Dabei sind zahlreiche Faktoren zu berücksichtigen.

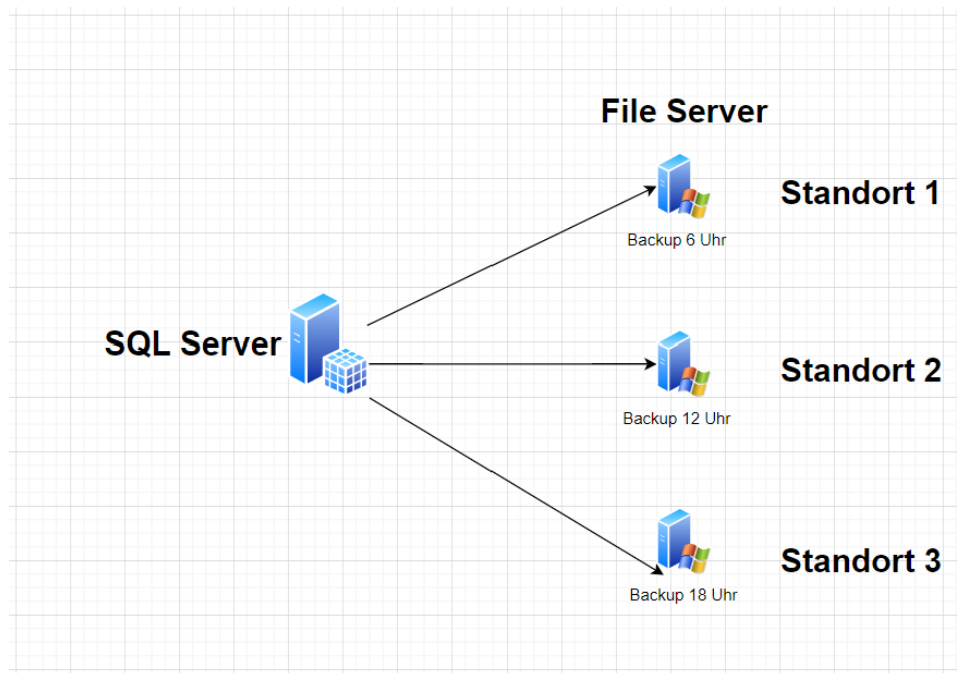


Abbildung 7: Darstellung der räumlichen Trennung von Backups

Zunächst muss ein Backup-Plan erstellt werden, welcher den Zeitpunkt und Ort der Speicherung festlegt. Die Backup-Frequenz ist auf drei Mal täglich festgelegt, und zwar auf 6 Uhr morgens, 12 Uhr mittags und 18 Uhr abends. Diese zeitlichen Intervalle ermöglichen eine regelmäßige Sicherung der Daten und gewährleisten, dass im Falle eines Zwischenfalls stets aktuelle Backups verfügbar sind. Um eine räumliche Trennung und damit höchstmögliche Sicherheit zu gewährleisten, könnten die Backups zu unterschiedlichen Zeiten auf Server mit verschiedenen Standorten gespeichert werden. Wie in der Darstellung zu sehen ist, wären somit 3 unterschiedliche Server Standorte benötigt. Hierbei können sowohl Cloud-Speicherdienste als auch physische Rechenzentren an unterschiedlichen Orten in Betracht gezogen werden. Die zeitlich gestaffelte und räumlich differenzierte Speicherung der Backups dient nicht nur der erhöhten Datensicherheit, sondern stellt auch eine wirksame Absicherung gegen potenzielle externe Bedrohungen wie Ransomware-Angriffe dar. Die Bedrohung durch Ransomware-Angriffe wird durch regelmäßige Backups und unterschiedliche Standorte deutlich minimieren.

Um menschliche Fehler zu minimieren, ist es entscheidend, dass Backups automatisch durch den MSSQL Server-Agenten programmiert werden. Die Überwachung dieses Prozesses spielt eine zentrale Rolle, insbesondere im Falle von Fehlern während eines Backups. Im Agenten können Benachrichtigungen so konfiguriert werden, dass bestimmte Personen über den

Verlauf des Backups informiert werden. Dies ermöglicht eine schnelle Reaktion im Falle von Fehlern und trägt dazu bei, potenzielle Probleme frühzeitig zu identifizieren und zu beheben. Um die Sicherheit von Backups zu gewährleisten, ist es entscheidend, diese zu verschlüsseln. Hierfür bietet sich Transparent Data Encryption (TDE) als effektive Methode an.

Zudem sollte immer ein Vollbackup durchgeführt werden, da Vollbackups alle Daten in der Datenbank einschließlich aller Tabellen, Indizes und gespeicherter Prozeduren umfassen. Dies gewährleistet, dass im Falle eines Datenverlusts oder einer Beschädigung der Datenbank sämtliche Informationen aus einem einzigen Backup wiederhergestellt werden können. Zudem sollten stets unterschiedliche Versionen der Backups aufbewahrt werden, um die Möglichkeit zu erhalten, auf frühere Datenstände zurückzugreifen. Es ist ratsam, die Backup-Versionen entsprechend der individuellen Anforderungen und Compliance-Richtlinien der Organisation für einen definierten Zeitraum aufzubewahren, um historische Datenintegrität zu gewährleisten.

## Aufgabe 6

Die Optimierung des SQL-Servers ist entscheidend, um die Leistung zu steigern und die Effizienz der Datenbankverwaltung zu verbessern. Die bereits in Aufgabe 1 durchgeführte Konsolidierung von drei Datenbanken zu einer einzigen vereinfacht die Datenbankverwaltung erheblich und ermöglicht eine effizientere Handhabung von Informationen, was wiederum die Gesamtleistung des Datenbanksystems verbessert.

Der Einsatz des Optimizers, einer Funktion der Datenbank, ist ein zentraler Aspekt bei der Steuerung der Verarbeitung von Data Manipulation Language (DML)-Anweisungen. Der Optimizer beeinflusst die Verarbeitung von SQL-Anweisungen, insbesondere die Zugriffsreihenfolge auf Tabellen oder Indizes, und trägt somit wesentlich zur Geschwindigkeit der Datenbankoperationen bei.

Die effiziente Planung der Hardware und Datenbankstruktur ist für die Performance des SQL-Servers entscheidend. Faktoren wie leistungsfähige Festplatten, vorzugsweise SSDs, und eine sinnvolle RAID-Konfiguration beschleunigen die Datenspeicherung und -abfrage, verbessern die Datenzugriffsgeschwindigkeit und gewährleisten Ausfallsicherheit. Die Anpassung der maximalen Datenbankgröße trägt dazu bei, Ressourcen effizient zu nutzen und unnötigen Speicherplatzverbrauch zu vermeiden, was insgesamt zu einer gesteigerten Leistungsfähigkeit der Datenbank führt.

Die logische oder algebraische Optimierung in einer Datenbank erfolgt durch die Ersetzung von Relationenalgebra-Ausdrücken mithilfe äquivalenter algebraischer Formen, unter Anwendung von Ersetzungsregeln oder Heuristiken. Dies schließt die Umformung von Anfragen durch die Verschiebung von Operationen zur Generierung kleinerer Zwischenergebnisse sowie die Identifizierung und Entfernung von Redundanzen ein. Praktische Empfehlungen für die Entwicklung einer effizienten Datenbank umfassen das sorgfältige Design von Tabellen, um eine optimale Struktur sicherzustellen. Es ist ratsam, versteckte Bremsen wie Trigger und komplexe Berechnungen zu vermeiden oder zu optimieren. Die Entscheidung zwischen Ad-hoc-Statements und gespeicherten Procedures kann die Datenbankleistung beeinflussen. Die Auswahl zwischen Ad-hoc-Statements und gespeicherten Procedures (Stored Procedures) kann die Datenbankleistung erheblich beeinflussen, da gespeicherte Procedures dazu neigen, besser optimiert und wiederverwendbar zu sein, was zu einer insgesamt effizienteren Datenbanknutzung führen

kann. Darüber hinaus spielt eine effektive Verwaltung von Tabellen und Indizes eine entscheidende Rolle bei der Optimierung der Datenbankperformance.

Die Abfrageoptimierung ist entscheidend, um die Leistung unserer SQL-Datenbank zu maximieren. Hierbei liegt der Fokus darauf, ineffiziente Strukturen in häufig genutzten Abfragen zu erkennen und zu verbessern. Durch gezielten Einsatz von Indexen, Vermeidung überflüssiger JOIN-Operationen und Vereinfachung komplexer Abfragen wird angestrebt, Antwortzeiten zu verkürzen und die Gesamtleistung der Datenbank zu steigern.

Zusätzlich steigern Views die Effizienz von Datenabfragen, indem sie vordefinierte komplexe Join-Operationen ermöglichen und den Zugriff auf spezifische Daten beschränken. Diese logische Abstraktion erleichtert die Datenzugriffsschicht und fördert die Wiederverwendbarkeit von Abfragen.

Selbstverständlich spielen auch die in Aufgabe 5 bereits genannten Aspekte eine zentrale Rolle, darunter effiziente Backup-Strategien, Verschlüsselung und ein zuverlässiges Monitoring.

## Anhang

### Quellen

<https://learnsql.de/blog/wie-exportiert-man-daten-aus-microsoft-sql-server-in-eine-csv-datei/>

<https://learn.microsoft.com/de-de/sql/integration-services/import-export-data/save-ssis-package-sql-server-import-and-export-wizard?view=sql-server-ver16>

<https://learn.microsoft.com/de-de/sql/integration-services/lift-shift/ssis-azure-schedule-packages-ssms?view=sql-server-ver16>

Folien: Datenbank optimieren

Folien: Datenbank Backups

Folien View Prozeduren und Trigger

### Anlagenverzeichnis

Diese Anlagen befinden sich in Gruppe7.zip

Anlage 1:	TableValueFunction.sql
Anlage 2:	InsertTrigger.sql
Anlage 3:	TestQueryTrigger.sql
Anlage 3:	TestData.sql
Anlage 4:	support.bak

## Aufgabenstellungen

### Szenario

Sie sind Mitarbeiter der IT-Abteilung eines Krankenhauses und haben folgendes Problem. Das Krankenhaus verwendet folgende Softwarepakete mit den Namen

- Support,
- Technik und
- Hauswirtschaft Tool

und erfasst mit diesem Tool alle anlaufenden folgende Support Tickets: Arbeitsbereiche:

- Alle IT-technischen Problemstellungen (Tickets).
- Alle Probleme die im Bereich Hauswirtschaft anläuft.
- Alle Probleme die in den Bereichen Technik anlaufen.

Die Applikationen sind ähnlich aufgebaut und verwenden auch gleiche Datenbanken. Sie erhalten von Ihrem Projektleiter 3 Backup Files, mit dem Sie die Datenbank erstellen sollen. In Zukunft sollen statt drei Datenbanken nur noch eine verwendet werden.

### Aufgabe 1 [14 von 60 Punkten]

Welche Änderungen müssen an der Datenbank vorgenommen werden, damit alle 3 Bereiche in einer Datenbank abgebildet werden können, ohne dass Tickets von anderen Bereichen aufgerufen werden. Lösen Sie, dass auf Datensatz Ebene, so dass der Programmierer wenig Aufwand hat das zu realisieren.

Zu erstellen:

Realisierung innerhalb einer Dokumentation (**Art Lastenheft für den Programmierer der Applikation**).

Realisierung innerhalb der Datenbank (**Programmierung Umsetzung**)

### Aufgabe 2 [14 von 60 Punkten]

Es soll ebenfalls berücksichtigt werden, dass in den einzelnen Bereichen teilweise die gleichen Mitarbeiter sind und teilweise aber das nicht unbedingt der Fall ist. Wie können Sie abbilden, dass in der User Tabelle der einzelne User nur einmal abgebildet werden und die Zugehörigkeit zu den einzelnen Bereichen gewählt werden können und dieser Bereich ausbaufähig bleibt ohne das Sie zukünftig die Tabellenstruktur ändern müssen.

Zu erstellen:

Realisierung innerhalb einer Dokumentation (Art **Lastenheft** für den Programmierer der Applikation).

Realisierung innerhalb der Datenbank (**Programmierung Umsetzung**)

**Aufgabe 3 [8 von 60 Punkten]**

Sie sollen die Tabelle „SUPPORT“ mit 25.000 Testdatensätze abfüllen, damit Sie die Applikation testen können. Wie realisieren Sie das in Ihrem DBMS. Berücksichtigen Sie, dass die FehlerKategorie und PersBereichID mit unterschiedlichen Daten gefüllt werden.

Zu erstellen:

**Realisierung innerhalb der Datenbank (Programmierung Umsetzung)****Aufgabe 4 [8 von 60 Punkten]**

Einmal im Quartal sollen die Tickets nach der Zuordnung der Bereiche und der Kategorien an ein DWH via CSV-File geliefert werden. Wie realisieren Sie die Automation in ein von Ihnen festgelegtes Directory. Die Realisierung über ein SQL Script gibt es in MSSQL nicht.

**Recherchieren Sie wie Sie das umsetzen können.****Aufgabe 5 [8 von 60 Punkten]**

Erstellen Sie ein Konzept und einen Plan für ein externes Backupsystem. Die Daten sollen aus Sicherheitsgründen um 6, 12 und 18 Uhr gesichert werden, ohne dass die Backups im gleichen Netzwerk liegen (Ransomware).

**Aufgabe 6 [8 von 60 Punkten]**

Erstellen Sie ein Konzept, um Ihren SQL-Server performanter zu machen.

**Informationen**

- Bitte auf MSSQL beschränken außer Aufgabe 4
- Bitte kurz aber gezielt halten
- Bilder oder Videos gerne erwünscht
- Bearbeitungszeitraum: 08.12.2023 um 9 Uhr bis 15.12.2023 6:59 Uhr
- Abgabe: 15.12.2023 von 7 bis 9 Uhr
- Präsentation: Am 15.12.2023 ab 10 Uhr
- Termine und Zeitslots für die Präsentation stelle ich ein, sobald ich von Einer Person (Kursprecher) die Liste der Gruppen (mit Namen und Matrikelnummern bitte) erhalten habe.
- Die Präsentation wird am Freitag den 15.12. stattfinden. Sie besteht aus einer LIVE-Demo Eures Systems und eine Übersicht über die Highlights der Ausarbeitung gefolgt von Fragen. Bitte die Kameras einschalten.
- Bei Fragen bitte per E-Mail an [m.trefzer@trefzer-it.de](mailto:m.trefzer@trefzer-it.de) senden. Nicht über Moodle Nachrichten senden