

• Triggers

Three Triggers are created as below.

1. Automatically update the status column in the Case table to 'Evaluation' upon getting a follow-up for an existing case.

```

CREATE TRIGGER UpdateCaseStatusOnFollowup
ON Follow_up
AFTER INSERT
AS
BEGIN
    UPDATE [Case]
    SET [Status] = 'Evaluation'
    FROM [Case]
    INNER JOIN Inserted i ON [Case].CaseID = i.CaseID;
END;

```

Messages

Command completed successfully.

Completion time: 2024-11-20T13:43:44.7853793-05:00

Query executed successfully.

```

| AS
| BEGIN
| UPDATE [Case]
| SET [Status] = 'Evaluation'
| FROM [Case]
| INNER JOIN Inserted i ON [Case].CaseID = i.CaseID;
| END;
| select * from [Case]

```

Results

CaseID	PatientID	ReporterID	CaseDescription	CaseDate	Status	Severity
2	112	2	Patient undergoing clinical trial	2024-01-05	Submitted for MR	3
3	113	3	Severe adverse reaction to medication	2024-02-01	MR In-Progress	5
4	114	4	Patient suffering from chronic illness	2024-02-10	Submitted for Quality Check	4
5	115	5	Case of drug overdose	2024-03-01	Quality Check In-Progress	5
6	116	6	Long-term side effects under observation	2024-03-15	Case Locked	3
7	117	7	Adverse reaction after drug administration	2024-04-01	Closed	2
8	118	8	Patient with multiple drug interactions	2024-04-05	Evaluation	4
9	119	9	Clinical trial results and analysis	2024-05-01	Submitted for MR	1
10	120	10	Patient on experimental drug regimen	2024-05-10	Closed	2

Query executed successfully.

SQLQuery1.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (66)) - Microsoft SQL Server Management Studio

```

10    END;
11
12    select * from [Case]
13
14    --Inserting data into follow-up table to verify the working of trigger
15    INSERT INTO Follow_up (CaseID, FollowupDate, FollowupNotes, ReportedBy)
16    VALUES (120, GETDATE(), 'Follow-up to check case status change trigger.', 'Medical Staff Personnel');
17

```

(1 row affected)
(1 row affected)

Completion time: 2024-11-20T13:59:25.7419764-05:00

Query executed successfully.

SQLQuery1.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (66)) - Microsoft SQL Server Management Studio

```

10    END;
11
12    select * from [Case]
13
14    --Inserting data into follow-up table to verify the working of trigger
15    INSERT INTO Follow_up (CaseID, FollowupDate, FollowupNotes, ReportedBy)
16    VALUES (120, GETDATE(), 'Follow-up to check case status change trigger.', 'Medical Staff Personnel');
17
18

```

CaseID	PatientID	ReporterID	CaseDescription	CaseDate	Status	Severity
2	112	2	Patient undergoing clinical trial	2024-01-05	Submitted for MR	3
3	113	3	Severe adverse reaction to medication	2024-02-01	MR In-Progress	5
4	114	4	Patient suffering from chronic illness	2024-02-10	Submitted for Quality Check	4
5	115	5	Case of drug overdose	2024-03-01	Quality Check In-Progress	5
6	116	6	Long-term side effects under observation	2024-03-15	Case Locked	3
7	117	7	Adverse reaction after drug administration	2024-04-01	Closed	2
8	118	8	Patient with multiple drug interactions	2024-04-05	Evaluation	4
9	119	9	Clinical trial results and analysis	2024-05-01	Submitted for MR	1
10	120	10	Patient on experimental drug regimen	2024-05-10	Evaluation	2

Query executed successfully.

2. Capture deleted cases in a new table 'DeletedCasesAudit' upon deletion from Case table.

SQLQuery2.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (51)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

Object Explorer

Connect

DESKTOP-UEGKJ04 (SQL Server)

Databases

System Databases

Database Snapshots

Adventure

DrugManagementSystem

DrugManagementSystem_

DrugManagementSystem_

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.Case

dbo.Case_Report

dbo.Case_Report

dbo.Device

dbo.Dose_Regimen

dbo.Drug

dbo.Follow_up

dbo.Patient

dbo.Product

dbo.Regulatory_Age

dbo.Regulatory_Age

dbo.Regulatory_Case

dbo.User

dbo.User_Case_Assig

Dropped Ledger Tab

Views

External Resources

Synonyms

Programmability

Query Store

Service Broker

SQLQuery2.sql - DESKTOP-UEGKJ04\Microsoft (51)* - SQLQuery1.sql - not connected*

```
1 --Audit table to log deleted cases
2 CREATE TABLE DeletedCasesAudit (
3     CaseID INT,
4     PatientID INT,
5     ReporterID INT,
6     CaseDescription TEXT,
7     CaseDate DATE,
8     [Status] VARCHAR(50),
9     Severity INT,
10    DeletedOn DATETIME DEFAULT GETDATE()
11 );
```

131 %

Messages

Commands completed successfully.

Completion time: 2024-11-20T14:38:55.0191021-05:00

Query executed successfully.

DESKTOP-UEGKJ04 (16.0 RTM) | DESKTOP-UEGKJ04\Micros.. | DrugManagementSystem2 | 00:00:00 | 0 rows

Ready

Type here to search

LN 2 Col 1 Ch 1 INS

NASDAQ -0.88% 14:38 20-11-2024

SQLQuery2.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (51)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

Object Explorer

Connect

DESKTOP-UEGKJ04 (SQL Server)

Databases

System Databases

Database Snapshots

Adventure

DrugManagementSystem

DrugManagementSystem_

DrugManagementSystem_

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.Case

dbo.Case_Report

dbo.Case_Report

dbo.Device

dbo.Dose_Regimen

dbo.Drug

dbo.Follow_up

dbo.Patient

dbo.Product

dbo.Regulatory_Age

dbo.Regulatory_Age

dbo.Regulatory_Case

dbo.User

dbo.User_Case_Assig

Dropped Ledger Tab

Views

External Resources

Synonyms

Programmability

Query Store

Service Broker

SQLQuery2.sql - DESKTOP-UEGKJ04\Microsoft (51)* - SQLQuery1.sql - not connected*

```
13 --Creating the trigger to log deleted cases
14 CREATE TRIGGER LogDeletedCases
15 ON [Case]
16 AFTER DELETE
17 AS
18 BEGIN
19     -- Insert deleted rows into DeletedCases table
20     INSERT INTO DeletedCasesAudit (CaseID, PatientID, ReporterID, CaseDescription, CaseDate, [Status], Severity, DeletedOn)
21     SELECT
22         CaseID,
23         PatientID,
24         ReporterID,
25         'N/A Case has been Deleted',
26         CaseDate,
27         [Status],
28         Severity,
29         GETDATE() AS DeletedOn
30     FROM Deleted;
31 END;
```

131 %

Messages

Commands completed successfully.

Completion time: 2024-11-20T14:54:04.3708410-05:00

Query executed successfully.

DESKTOP-UEGKJ04 (16.0 RTM) | DESKTOP-UEGKJ04\Micros.. | DrugManagementSystem2 | 00:00:00 | 0 rows

Ready

Type here to search

LN 14 Col 1 Ch 1 INS

52°F Sunny 14:54 20-11-2024

SQLQuery2.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (51)) - Microsoft SQL Server Management Studio

```
-- Insert deleted rows into DeletedCases table
INSERT INTO DeletedCasesAudit (CaseID, PatientID, ReporterID, CaseDescription, CaseDate, [Status], Severity, DeletedOn)
SELECT
    CaseID,
    PatientID,
    ReporterID,
    'N/A Case has been Deleted',
    CaseDate,
    [Status],
    Severity,
    GETDATE() AS DeletedOn
FROM Deleted;
END;

select * from [Case]
```

131% Results Messages

CaseID	PatientID	ReporterID	CaseDescription	CaseDate	Status	Severity
6	116	6	Long-term side effects under observation	2024-03-15	Case Locked	3
7	117	7	Adverse reaction after drug administration	2024-04-01	Closed	2
8	118	8	Patient with multiple drug interactions	2024-04-05	Evaluation	4
9	119	9	Clinical trial results and analysis	2024-05-01	Submitted for MR	1
10	120	10	Patient on experimental drug regimen	2024-05-10	Evaluation	2
11	1112	1	Test case description.	2024-11-20	Evaluation	2

Query executed successfully.

SQLQuery2.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (51)) - Microsoft SQL Server Management Studio

```
-- Insert deleted rows into DeletedCases table
INSERT INTO DeletedCasesAudit (CaseID, PatientID, ReporterID, CaseDescription, CaseDate, [Status], Severity, DeletedOn)
SELECT
    CaseID,
    PatientID,
    ReporterID,
    'N/A Case has been Deleted',
    CaseDate,
    [Status],
    Severity,
    GETDATE() AS DeletedOn
FROM Deleted;
END;

select * from [Case]
select * from DeletedCasesAudit
```

131% Results Messages

CaseID	PatientID	ReporterID	CaseDescription	CaseDate	Status	Severity	DeletedOn
--------	-----------	------------	-----------------	----------	--------	----------	-----------

Query executed successfully.

SQLQuery2.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (S1)) - Microsoft SQL Server Management Studio

```

26    Severity,
27    GETDATE() AS DeletedOn
28  FROM Deleted;
29
30
31  select * from [Case];
32
33  select * from DeletedCasesAudit;
34
35
36  DELETE FROM [Case]
37 WHERE CaseID = 1112;
38
39

```

131% 4 Messages

(1 row affected)
(1 row effected)
Completion time: 2024-11-20T14:58:37.5507018-05:00

Query executed successfully.

DESKTOP-UEGKJ04 (16.0 RTM) | DESKTOP-UEGKJ04\Micr... | DrugManagementSystem2 | 00:00:00 | 0 rows

SQLQuery2.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (S1)) - Microsoft SQL Server Management Studio

```

26    Severity,
27    GETDATE() AS DeletedOn
28  FROM Deleted;
29
30
31  select * from [Case];
32
33  select * from DeletedCasesAudit;
34
35
36  DELETE FROM [Case]
37 WHERE CaseID = 1112;
38
39

```

131% 4 Results Messages

CaseID	PatientID	ReporterID	CaseDescription	CaseDate	Status	Severity	DeletedOn
1112	1	3	N/A Case has been Deleted	2024-11-20	Evaluation	2	2024-11-20 14:58:37.540

Query executed successfully.

DESKTOP-UEGKJ04 (16.0 RTM) | DESKTOP-UEGKJ04\Micr... | DrugManagementSystem2 | 00:00:00 | 1 rows

3. To log changes in the status column of Case table, the log is captured in a separate log table that is created 'CaseStatusAudit'

SQLQuery3.sql - DESKTOP-UEGKJ04.master (DESKTOP-UEGKJ04\Microsoft (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

master

Object Explorer

SQLQuery3.sql - DESKTOP-UEGKJ04 (SQL Server) -> Trigger2.sql - DESKTOP-UEGKJ04 (Microsoft (51)) -> Trigger1.sql - not connected

```
1 --Audit table to log changes in the status column of Case table
2 CREATE TABLE CaseStatusAudit (
3     AuditID INT IDENTITY(1,1) PRIMARY KEY,
4     CaseID INT,
5     OldStatus VARCHAR(50),
6     NewStatus VARCHAR(50),
7     ChangedOn DATETIME DEFAULT GETDATE()
8 );
```

131% Messages

Command completed successfully.

Completion time: 2024-11-20T15:02:57.3077712-05:00

Query executed successfully.

DESKTOP-UEGKJ04 (16.0 RTM) DESKTOP-UEGKJ04\Microsoft (62) master 0:00:00 0 rows

Ready Type here to search

15:02 20-11-2024

SQLQuery3.sql - DESKTOP-UEGKJ04.DrugManagementSystem2 (DESKTOP-UEGKJ04\Microsoft (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

DrugManagementSystem2

Object Explorer

SQLQuery3.sql - DESKTOP-UEGKJ04 (SQL Server) -> Trigger2.sql - DESKTOP-UEGKJ04 (Microsoft (51)) -> Trigger1.sql - not connected

```
6     NewStatus VARCHAR(50),
7     ChangedOn DATETIME DEFAULT GETDATE()
8 );
9
10
11 --Creating the Trigger
12 CREATE TRIGGER LogCaseStatusChange
13 ON [Case]
14 AFTER UPDATE
15 AS
16 BEGIN
17     INSERT INTO CaseStatusAudit (CaseID, OldStatus, NewStatus)
18     SELECT i.CaseID, d.[Status], i.[Status]
19     FROM Inserted i
20     INNER JOIN Deleted d ON i.CaseID = d.CaseID
21     WHERE i.[Status] <> d.[Status];
22 END;
```

131% Messages

Command completed successfully.

Completion time: 2024-11-20T15:07:46.1394184-05:00

Query executed successfully.

DESKTOP-UEGKJ04 (16.0 RTM) DESKTOP-UEGKJ04\Microsoft (62) DrugManagementSystem2 0:00:00 0 rows

Ready Type here to search

15:07 20-11-2024

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'DESKTOP-UEGKJ04\Microsoft (62)'. The left pane shows the Object Explorer with the 'DrugManagementSystem' database selected. The right pane displays a query window titled 'Trigger2.sql - DESKTOP-UEGKJ04\Microsoft (51)'. The script creates a trigger named 'CaseStatusAudit' on the 'Case' table:

```
16 BEGIN  
17     INSERT INTO CaseStatusAudit (CaseID, OldStatus, NewStatus)  
18     SELECT i.CaseID, d.[Status], i.[Status]  
19     FROM Inserted i  
20     INNER JOIN Deleted d ON i.CaseID = d.CaseID  
21     WHERE i.[Status] <> d.[Status];  
22 END;  
23  
24 select * from [Case]  
25  
26 --Updating Status column of one of the cases from Case table to verify the trigger  
27 UPDATE [Case]  
28 SET [Status] = 'MR In-Progress'  
29 WHERE CaseID = 117;
```

The status bar at the bottom shows 'Query executed successfully.' and the completion time '2024-11-20T15:13:35.2577567-05:00'. The bottom right corner shows the date and time '20-11-2024 15:13'.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure for 'DESKTOP-UEGKJ04'. The central pane displays a script for creating a trigger named 'Trigger2.sql'.

```
22 END;
23
24 select * from [Case]
25
26 --updating Status column of one of the cases from Case table to verify the trigger
27 UPDATE [Case]
28 SET [Status] = 'MR In-Progress'
29 WHERE CaseID = 117;
30
31 select * from CaseStatusAudit.
```

The 'Results' tab at the bottom shows the output of the executed query:

	AuditID	CaseID	OldStatus	NewStatus	ChangedOn
1	1	117	Closed	MR In-Progress	2024-11-20 15:13:35.240

At the bottom status bar, it says 'Query executed successfully.'

Triggers are created as per the script mentioned in psm_script.sql (The DDL for new audit tables required for triggers to work are also present in the script).

- **Encryptions**

Before encrypting the columns of the tables, we have altered the data types of few columns from varchar(x)/TEXT to VARCHAR(MAX) that are being encrypted, so that the length of the column is sufficient to store encrypted data and because EncryptByKey function in SQL Server does not

support the TEXT data type.

```
ALTER TABLE Dose_Regimen
ALTER COLUMN Dosage VARCHAR(MAX);
ALTER TABLE Dose_Regimen
ALTER COLUMN Frequency VARCHAR(MAX);

ALTER TABLE Case_Report
ALTER COLUMN ReporterName VARCHAR(MAX);
ALTER TABLE Case_Report
ALTER COLUMN ReporterRole VARCHAR(MAX);
ALTER TABLE Case_Report
ALTER COLUMN ContactNo VARCHAR(MAX);
ALTER TABLE Case_Report
ALTER COLUMN ReporterNotes VARCHAR(MAX);

ALTER TABLE Patient
ALTER COLUMN MedicalHistory VARCHAR(MAX);
ALTER TABLE Patient
ALTER COLUMN ContactNo VARCHAR(MAX);
```

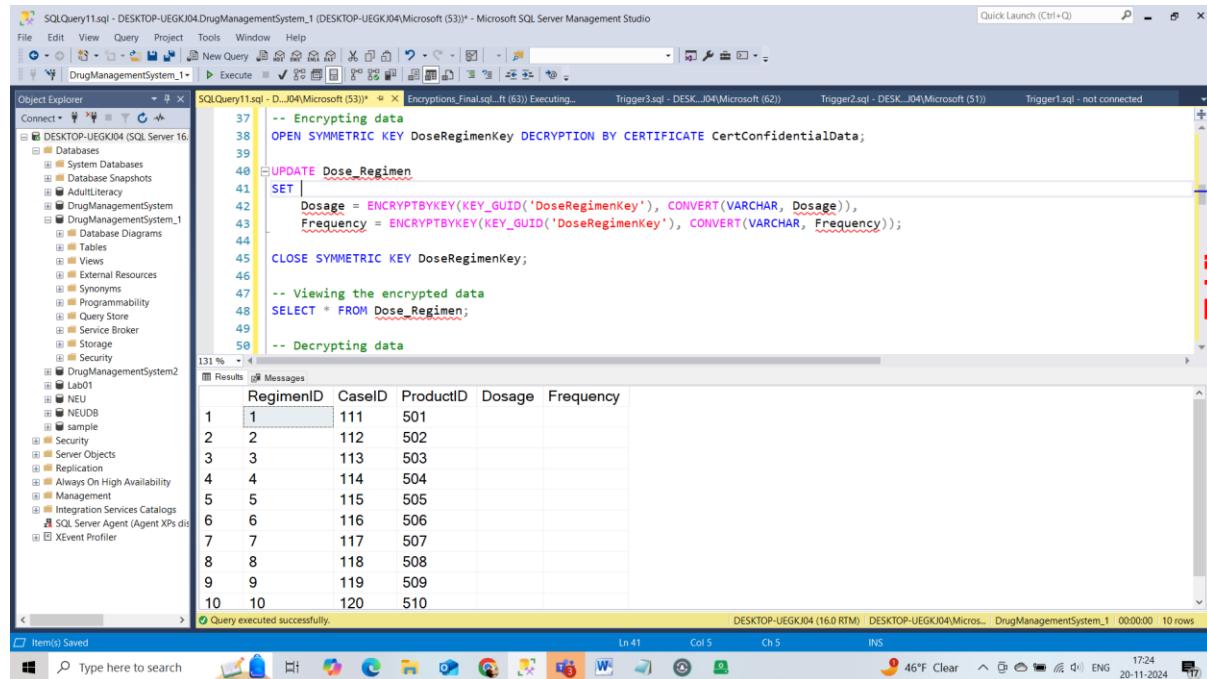
Created encryptions with symmetric keys on columns containing confidential data. Below tables and their respective columns were encrypted. Separate symmetric keys were created for each table encryption.

1. Dose_Regimen: Dosage, Frequency
2. Case_Report: ReporterName, ReporterRole, ContactNo, ReporterNotes
3. Patient: PatientName, MedicalHistory, Allergies, ContactNo

Encryption is performed as per the script: Encryption_Script.sql

Screenshots:

1. Dose_Regimen:



```
-- Encrypting data
OPEN SYMMETRIC KEY DoseRegimenKey DECRYPTION BY CERTIFICATE CertConfidentialData;
UPDATE Dose_Regimen
SET
    Dosage = ENCRYPTBYKEY(KEY_GUID('DoseRegimenKey'), CONVERT(VARCHAR, Dosage)),
    Frequency = ENCRYPTBYKEY(KEY_GUID('DoseRegimenKey'), CONVERT(VARCHAR, Frequency));
CLOSE SYMMETRIC KEY DoseRegimenKey;

-- Viewing the encrypted data
SELECT * FROM Dose_Regimen;

-- Decrypting data
```

RegimenID	CasID	ProductID	Dosage	Frequency
1	1	111	501	
2	2	112	502	
3	3	113	503	
4	4	114	504	
5	5	115	505	
6	6	116	506	
7	7	117	507	
8	8	118	508	
9	9	119	509	
10	10	120	510	

SQLQuery11.sql - DESKTOP-UEGKJ04\DrugManagementSystem_1 (DESKTOP-UEGKJ04\Microsoft (53)) - Microsoft SQL Server Management Studio

```

49
50    -- Decrypting data
51    OPEN SYMMETRIC KEY DoseRegimenKey DECRYPTION BY CERTIFICATE CertConfidentialData;
52
53    SELECT
54        RegimenID,
55        CaseID,
56        ProductID,
57        CAST(DECRYPTBYKEY(Dosage) AS VARCHAR(MAX)) AS DecryptedDosage,
58        CAST(DECRYPTBYKEY(Frequency) AS VARCHAR(MAX)) AS DecryptedFrequency
59    FROM Dose_Regimen;
60
61    CLOSE SYMMETRIC KEY DoseRegimenKey;
62

```

Results

	RegimenID	CaseID	ProductID	DecryptedDosage	DecryptedFrequency
1	1	111	501	500 mg	Once Daily
2	2	112	502	5 ml	Twice a Day
3	3	113	503	1 g	Once Weekly
4	4	114	504	200 mg	Every 12 Hours
5	5	115	505	5 mg	Once Daily
6	6	116	506	1.5 ml	Twice a Day
7	7	117	507	750 mg	Once Daily
8	8	118	508	2 ml	Twice a Day
9	9	119	509	10 mg	Once Daily
10	10	120	510	5 mg	Once Daily

Query executed successfully.

2. Case_Reportert:

SQLQuery11.sql - DESKTOP-UEGKJ04\DrugManagementSystem_1 (DESKTOP-UEGKJ04\Microsoft (53)) - Microsoft SQL Server Management Studio

```

70
71    -- Encrypting data
72    OPEN SYMMETRIC KEY CaseReporterKey DECRYPTION BY CERTIFICATE CertConfidentialData;
73
74    UPDATE Case_Reportert
75    SET
76        ReporterName = ENCRYPTBYKEY(KEY_GUID('CaseReporterKey'), CONVERT(VARCHAR, ReporterName)),
77        ReporterRole = ENCRYPTBYKEY(KEY_GUID('CaseReporterKey'), CONVERT(VARCHAR, ReporterRole)),
78        ContactNo = ENCRYPTBYKEY(KEY_GUID('CaseReporterKey'), CONVERT(VARCHAR, ContactNo)),
79        ReporterNotes = ENCRYPTBYKEY(KEY_GUID('CaseReporterKey'), CONVERT(VARCHAR, ReporterNotes));
80
81    CLOSE SYMMETRIC KEY CaseReporterKey;
82
83    -- Viewing the encrypted data
84    SELECT * FROM Case_Reportert;
85

```

Results

	ReporterID	ReporterName	ReporterRole	ContactNo	ReporterNotes
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8	8				
9	9				
10	10				

Query executed successfully.

SQLQuery11.sql - DESKTOP-UEGKJ04.DrugManagementSystem_1 (DESKTOP-UEGKJ04\Microsoft (53)) - Microsoft SQL Server Management Studio

```

-- Decrypting data
OPEN SYMMETRIC KEY CaseReporterKey DECRYPTION BY CERTIFICATE CertConfidentialData;
SELECT
    ReporterID,
    CAST(DECRYPTBYKEY(ReporterName) AS VARCHAR(100)) AS DecryptedReporterName,
    CAST(DECRYPTBYKEY(ReporterRole) AS VARCHAR(50)) AS DecryptedReporterRole,
    CAST(DECRYPTBYKEY(ContactNo) AS VARCHAR(20)) AS DecryptedContactNo,
    CAST(DECRYPTBYKEY(ReporterNotes) AS VARCHAR(MAX)) AS DecryptedReporterNotes
FROM Case_Report;
CLOSE SYMMETRIC KEY CaseReporterKey;

```

ReporterID	DecryptedReporterName	DecryptedReporterRole	DecryptedContactNo	DecryptedReporterNotes
1	John Smith	Investigator	123-456-7890	Lead investigator for clinical
2	Emma Davis	Analyst	234-567-8901	Responsible for case analysis
3	Lucas Brown	Coordinator	345-678-9012	Coordinate with the regulatory
4	Olivia Clark	Investigator	456-789-0123	Leads case follow-ups and repo
5	James Taylor	Analyst	567-890-1234	Works on case documentation an
6	Sophia Lee	Coordinator	678-901-2345	Handles follow-up reports and
7	David Wilson	Investigator	789-012-3456	Experienced in case handling a
8	Isabella Martinez	Analyst	890-123-4567	Assists in preparing case repo
9	Ethan Anderson	Coordinator	901-234-5678	Responsible for case tracking
10	Ava Robinson	Investigator	012-345-6789	Specializes in case documentat

Query executed successfully.

3. Patient:

SQLQuery11.sql - DESKTOP-UEGKJ04.DrugManagementSystem_1 (DESKTOP-UEGKJ04\Microsoft (53)) - Microsoft SQL Server Management Studio

```

-- Encrypting data
OPEN SYMMETRIC KEY PatientKey DECRYPTION BY CERTIFICATE CertConfidentialData;
UPDATE Patient
SET
    PatientName = ENCRYPTBYKEY(KEY_GUID('PatientKey'), PatientName),
    MedicalHistory = ENCRYPTBYKEY(KEY_GUID('PatientKey'), MedicalHistory),
    Allergies = ENCRYPTBYKEY(KEY_GUID('PatientKey'), Allergies),
    ContactNo = ENCRYPTBYKEY(KEY_GUID('PatientKey'), ContactNo);
CLOSE SYMMETRIC KEY PatientKey;
-- Viewing the encrypted data
SELECT * FROM Patient;

```

PatientID	PatientName	MedicalHistory	Allergies	ContactNo
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Query executed successfully.

```

-- Decrypting data
OPEN SYMMETRIC KEY PatientKey DECRYPTION BY CERTIFICATE CertConfidentialData;
SELECT
    PatientID,
    CAST(DECRYPTBYKEY(PatientName) AS VARCHAR(100)) AS DecryptedPatientName,
    CAST(DECRYPTBYKEY(MedicalHistory) AS VARCHAR(MAX)) AS DecryptedMedicalHistory,
    CAST(DECRYPTBYKEY(Allergies) AS VARCHAR(100)) AS DecryptedAllergies,
    CAST(DECRYPTBYKEY(ContactNo) AS VARCHAR(20)) AS DecryptedContactNo
FROM Patient;
CLOSE SYMMETRIC KEY PatientKey;

```

PatientID	DecryptedPatientName	DecryptedMedicalHistory	DecryptedAllergies	DecryptedContactNo
1	John Doe	No significant history	None	123-456-7890
2	Jane Smith	Diabetes	Peanuts	234-567-8901
3	Sam Green	Asthma	None	345-678-9012
4	Lisa White	Hypertension	Penicillin	456-789-0123
5	Mike Black	No significant history	None	567-890-1234
6	Sophia Blue	Chronic Migraine	None	678-901-2345
7	David Grey	No significant history	Shellfish	789-012-3456
8	Isabella Brown	Cancer	None	890-123-4567
9	Ethan Yellow	No significant history	None	901-234-5678
10	Ava Red	No significant history	None	012-345-6789

● Indexes

- 1) CREATE INDEX idx_Case_PatientID ON [Case](PatientID);
- 2) CREATE INDEX idx_Followup_CaseID ON Follow_up(CaseID);
- 3) CREATE INDEX idx_Followup_CaseID_FollowupDate ON Follow_up(CaseID, FollowupDate);

The above indexes are created to improve the performance of CaseFollowupSummary View, as PatientID is used in the ON clause of the join, Follow_up is being joined with Case using CaseID, indexing this column will improve the performance of the join, and a composite index because composite index helps to optimize the query for retrieving the most recent follow-up date (MAX(f.FollowupDate)) as well as counting the follow-ups (COUNT(f.FollowupID)), which are grouped by CaseID.

CaseFollowupSummary is defined as below:

```

CREATE VIEW CaseFollowupSummary AS
SELECT
    c.CaseID,
    p.PatientName,
    CAST(c.CaseDescription AS NVARCHAR(MAX)) AS CaseDescription,
    c.[Status] AS CaseStatus,
    MAX(f.FollowupDate) AS LastFollowupDate,
    COUNT(f.FollowupID) AS TotalFollowups
FROM [Case] c
INNER JOIN Patient p ON c.PatientID = p.PatientID
LEFT JOIN Follow_up f ON c.CaseID = f.CaseID
GROUP BY c.CaseID, p.PatientName, CAST(c.CaseDescription AS NVARCHAR(MAX)),
c.[Status];

```

```
4) CREATE INDEX idx_RCCR_AgencyID ON Regulatory_Case_Reports(AgencyID);
5) CREATE INDEX idx_RA_AgencyID ON Regulatory_Agency(AgencyID);
6) CREATE INDEX idx_RCCR_TrackingNum ON Regulatory_Case_Reports(TrackingNum);
```

The above indexes are created to improve the performance of vw_RegulatoryReports View, as AgencyID column in the Regulatory_Agency table is used in the ON clause of the join and if a TrackingNum is frequently used for filtering, grouping, or sorting (or could be in future queries against the view), creating an index on RCCR.TrackingNum would speed up those operations.

vw_RegulatoryReports is defined as below:

```
CREATE VIEW vw_RegulatoryReports AS
SELECT
    RA.AgencyName,
    RCCR.TrackingNum,
    RCCR.[Status],
    RCCR.SubmissionDate
FROM
    Regulatory_Case_Reports RCCR
JOIN
    Regulatory_Agency RA
ON
    RCCR.AgencyID = RA.AgencyID
```