

Upper and Lower

The image displays two SQL queries side-by-side in SQL Server Enterprise Manager, demonstrating the use of the `UPPER` and `LOWER` functions.

Left Query (SQLQuery3.sql):

```
SELECT TOP (1000) [Patient_LastName]
, [Doc_LastName]
, [TreatmentID]
, [DateTime]
, [Results]
FROM [hospital].[dbo].[TreatmentAdministration_Vw]
```

Right Query (SQLQuery2.sql):

```
SELECT UPPER(Patient_LastName) AS patientName, LOWER(Doc_LastName) as docName
FROM TreatmentAdministration_Vw
```

Results:

Left Results Table:

	Patient_LastName	Doc_LastName	TreatmentID	DateTime	Results
1	Martin	Green	1	2023-09-18 16:30:00.000	Tumor Detection
2	Martin	Green	2	2023-09-18 16:30:00.000	Tumor Detection
3	Lewis	Lewis	1	2023-09-18 10:30:00.000	Normal Anatomy
4	Harris	Thompson	1	2023-09-20 15:20:00.000	Cardiac detected
5	Harris	Thompson	12	2023-09-20 15:20:00.000	Negative
6	Smith	Clark	4	2023-09-18 08:45:00.000	Improved mobility
7	Thomas	Carlson	2	2023-09-21 16:20:00.000	Gum disease
8	Young	Scott	10	2023-09-21 13:30:00.000	Fluid removed
9	Williams	Baker	7	2023-09-21 07:30:00.000	Benign finding

Right Results Table:

	patientName	docName
1	MARTIN	green
2	MARTIN	green
3	LEWIS	lewis
4	HARRIS	thompson
5	HARRIS	thompson
6	SMITH	clark
7	THOMAS	carlson
8	YOUNG	scott
9	WILLIAMS	baker

Intersect

The image displays two SQL queries side-by-side in SQL Server Enterprise Manager, demonstrating different ways to find the intersection of PatientID values from two tables: Patient and TreatmentAdministration.

Left Query (SQLQuery1.sql):

```
SELECT PatientID
FROM Patient
INTERSECT
SELECT PatientID
FROM TreatmentAdministration
```

Right Query (SQLQuery2.sql):

```
SELECT DISTINCT p.PatientID
FROM Patient p
INNER JOIN TreatmentAdministration t
ON p.PatientID = t.PatientID
```

Results:

Both queries return the same set of PatientID values, which are the intersection of the two tables. The results are displayed in a grid below each query window.

	PatientID
1	2
2	4
3	6
4	7
5	8
6	10
7	11

Except

SQLQuery1.sql - P...HUONG\heeph (70))*

```
SELECT DoctorID
FROM Physician
EXCEPT
SELECT DoctorID
FROM TreatmentAdministration
```

SQLQuery2.sql - P...HUONG\heeph (64))*

```
SELECT DISTINCT p.DoctorID
FROM Physician p
LEFT JOIN TreatmentAdministration t
ON p.DoctorID = t.DoctorID
WHERE t.DoctorID IS NULL
```

74 %

Results Messages

	DoctorID
1	1
2	7
3	10
4	11
5	12
6	13
7	14
8	15
9	16
10	17
11	18
12	19
13	20
14	21
15	22
16	23
17	24

74 %

Results Messages

	DoctorID
1	1
2	7
3	10
4	11
5	12
6	13
7	14
8	15
9	16
10	17
11	18
12	19
13	20
14	21
15	22
16	23
17	24

✓ PHUONG\SQLEXPRESS (16.0 RTM) PHUONG\heeph (70) hospital 00:00:00 103 rows

✓ Query execu... PHUONG\SQLEXPRESS (16.0 RTM) PHUONG\heeph (64) hospital 00:00:00 103 row

Union

The image displays two side-by-side SQL query windows in SQL Server Enterprise Manager. The left window, titled 'SQLQuery3.sql - P...HUONG\heeph (54))*', contains the following SQL query:

```
SELECT AdmitDoc FROM AdmitDischarge a1
UNION
SELECT DischargeDoc FROM AdmitDischarge a2
```

The right window, titled 'SQLQuery2.sql - P...HUONG\heeph (64))*', contains the following SQL query:

```
SELECT DISTINCT COALESCE(a1.AdmitDoc, a2.DischargeDoc) AS Doc
FROM AdmitDischarge a1
FULL OUTER JOIN AdmitDischarge a2
ON a1.AdmitDoc = a2.DischargeDoc
```

Both windows show the 'Results' tab with a table of 10 rows. The left window's result table has a column named 'AdmitDoc' and the right window's result table has a column named 'Doc'. Both tables contain the same data: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

	AdmitDoc
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

	Doc
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10