

Module 1: SQL Server ----> Data Source 1: Inpatient_provd_r_Covid_Healthcare

2nd NORMALIZATION FORM

1. ProviderDetails Table (2nd Normalization Form)

The screenshot shows the DBeaver 22.2.0 interface with the following SQL scripts:

```
--1st Normalization Form: Table is already in 1st NF
*****
--2nd Normalization Form

--Inpatient_provd_r_Covid_Healthcare Table can be Split into 2 by separating Provider_ID, Provider_Name, Average_Medicare_Payments in one table and Provider_ID, Provider_Street_Address in another
--It is because Provider_Street_Address is dependent on Provider_ID and it breaks the 2NF rule

--Drug ProviderDetails Table

CREATE TABLE ProviderDetails
(
    Provider_ID FLOAT PRIMARY KEY,
    Drug_ID FLOAT,
    Provider_Name VARCHAR(50) NOT NULL,
    Average_Medicare_Payments FLOAT CHECK (Average_Medicare_Payments >= 3000)
)

SELECT * FROM ProviderDetails

INSERT INTO ProviderDetails VALUES
(10001, 39, 'SOUTHEAST ALABAMA MEDICAL CENTER', 4763.736264),
```

The results pane shows the following data:

Provider_ID	Drug_ID	Provider_Name	Average_Medicare_Payments
10,001	39	SOUTHEAST ALABAMA MEDICAL CENTER	4,763.736264
10,005	39	MARSHALL MEDICAL CENTER SOUTH	4,976.714286
10,006	39	ELIZA COFFEE MEMORIAL HOSPITAL	4,453.791667
10,011	39	ST VINCENTS EAST	4,129.16
10,016	39	SHELBY BAPTIST MEDICAL CENTER	4,851.444444
10,023	39	BAPTIST MEDICAL CENTER SOUTH	5,374.140254

2. ProviderAddress Table (2nd Normalization Form)

The screenshot shows the DBeaver 22.2.0 interface with the following SQL scripts:

```
--Drug ProviderAddress Table

CREATE TABLE ProviderAddress
(
    Provider_ID FLOAT FOREIGN KEY REFERENCES ProviderDetails(Provider_ID),
    Provider_Street_Address VARCHAR(50) UNIQUE
)

SELECT * FROM ProviderAddress

drop table ProviderAddress

INSERT INTO ProviderAddress VALUES
(10001, '1108 ROSS CLARK CIRCLE'),
(10005, '2505 U S HIGHWAY 431 NORTH'),
(10006, '205 MARENGO STREET'),
(10011, '50 MEDICAL PARK EAST DRIVE'),
(10016, '1000 FIRST STREET NORTH'),
(10023, '2105 EAST SOUTH BOULEVARD'),
(10029, '2000 PEPPERELL PARKWAY')
```

The results pane shows the following data:

Provider_ID	Provider_Street_Address
10,001	1108 ROSS CLARK CIRCLE
10,005	2505 U S HIGHWAY 431 NORTH
10,006	205 MARENGO STREET
10,011	50 MEDICAL PARK EAST DRIVE
10,016	1000 FIRST STREET NORTH
10,023	2105 EAST SOUTH BOULEVARD
10,029	2000 PEPPERELL PARKWAY

3rd NORMALIZATION FORM

3. ProviderDetailss Table (3rd Normalization Form)

DBBeaver 22.2.0 - <nehat> NORMALIZATION

File Edit Navigate Search SQL Editor Database Window Help

nehat dbo@nehat

--3rd Normalization Form

--Inpatient_provd Covid_Healthcare Table can be Split into 2 by separating Provider_ID, Drug_ID, Provider_Name, Provider_Zip_Code in one table and Provider_Zip_Code

--It is because Provider_Zip_Code is dependent on Provider_ID that makes non primary attributes (Provider_City, Provider_State) transitively dependent on Provider_ID

--ProviderDetailss Table

```
CREATE TABLE ProviderDetailss
(
    Provider_ID FLOAT PRIMARY KEY,
    Drug_ID FLOAT,
    Provider_Name VARCHAR(50) NOT NULL,
    Provider_Zip_Code FLOAT
)
```

```
SELECT * FROM ProviderDetailss
```

```
INSERT INTO ProviderDetailss VALUES
(10001, 39, 'SOUTHEAST ALABAMA MEDICAL CENTER', 36301),
```

Results 1 x

SELECT * FROM ProviderDetailss

Grid	Provider_ID	Drug_ID	Provider_Name	Provider_Zip_Code
1	10.001	39	SOUTHEAST ALABAMA MEDICAL CENTER	36.301
2	10.005	39	MARSHALL MEDICAL CENTER SOUTH	35.957
3	10.006	39	ELIZA COFFEE MEMORIAL HOSPITAL	35.631
4	10.011	39	ST VINCENTS EAST	35.235
5	10.016	39	SHELBY BAPTIST MEDICAL CENTER	35.007
6	10.023	39	BAPTIST MEDICAL CENTER SOUTH	36.116
7	10.029	39	EAST ALABAMA MEDICAL CENTER AND SNF	36.801

Value x

Save Cancel Script

IST en Writable Smart Insert 113 : 22 : 4462 Sel: 0 | 0

4. Provider_ZipCode Table (3rd Normalization Form)

DBBeaver 22.2.0 - <nehat> NORMALIZATION

File Edit Navigate Search SQL Editor Database Window Help

nehat dbo@nehat

--Provide Zip_Code Table

```
CREATE TABLE Provider_ZipCode
(
    Provider_Zip_Code FLOAT,
    Provider_City VARCHAR(50),
    Provider_State VARCHAR(50)
)
```

```
SELECT * FROM Provider_ZipCode
```

```
INSERT INTO Provider_ZipCode VALUES
(36301, 'DOTHAN', 'AL'),
(35957, 'BOAZ', 'AL'),
(35631, 'FLORENCE', 'AL'),
(35235, 'BIRMINGHAM', 'AL'),
(35007, 'ALABASTER', 'AL'),
(36116, 'MONTGOMERY', 'AL'),
(36801, 'OPELIKA', 'AL'),
```

Results 1 x

SELECT * FROM Provider_ZipCode

Grid	Provider_Zip_Code	Provider_City	Provider_State
1	36.301	DOTHAN	AL
2	35.957	BOAZ	AL
3	35.631	FLORENCE	AL
4	35.235	BIRMINGHAM	AL
5	35.007	ALABASTER	AL
6	36.116	MONTGOMERY	AL
7	36.801	OPELIKA	AL

Value x

Save Cancel Script

IST en Writable Smart Insert 156 : 30 : 6248 Sel: 0 | 0

JOINS

5. INNER JOIN

DBBeaver 22.2.0 - <nehat> NORMALIZATION

File Edit Navigate Search SQL Editor Database Window Help

Auto SQL Commit Rollback T Auto nehat dbo@nehat

Database Navigator Projects

Enter a part of object name here

dlithe - rewansolution.database.windows.net:1433

nehat - rewansolution.database.windows.net:1433

Databases

nehat

Schemas

dbo

Tables

BANK_TRANSACTION 1.1M

CAR_ProductDetails 72K

CAR_SupplierAddress 72K

CAR_SupplierDetails 72K

CAR_SupplierName 72K

Car_Sale 840K

Cinema_Tickets 6.8M

DEPARTMENT 72K

DEPARTMENTFF 72K

DIM_EMPLOYEE 72K

Dim_Employees 72K

EMP 72K

EMPLOYEE 72K

Project - General

Name DataSource

Bookmarks

Diagrams

Scripts

<nehat> NORMALIZATION_INSU <nehat> NORMALIZATION <dlithe> RANK <dlithe> JOINS

```
-- INNER JOIN
-- (Returns Only Common data from other table)

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetailss AS PD
INNER JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- LEFT JOIN
-- (Returns all rows from the left table (First table) and matching rows from the right table)

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetailss AS PD
LEFT JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- RIGHT JOIN JOIN
-- (Returns all rows from the right table (Second table) and matching rows from the left table)
```

Results 1 x

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State

Provider_ID	Drug_ID	Provider_Name	Provider_Zip_Code	Provider_State
10.001	39	SOUTHEAST ALABAMA MEDICAL CENTER	36.301	AL
10.005	39	MARSHALL MEDICAL CENTER SOUTH	35.957	AL
10.006	39	ELIZA COFFEE MEMORIAL HOSPITAL	35.631	AL
10.011	39	ST VINCENTS EAST	35.235	AL
10.016	39	SHELBY BAPTIST MEDICAL CENTER	35.007	AL
10.023	39	BAPTIST MEDICAL CENTER SOUTH	36.116	AL
10.029	39	EAST ALABAMA MEDICAL CENTER AND SNF	36.801	AL

IST en Writable Smart Insert 291:24:10659 Sel:0|0

6. LEFT JOIN

DBBeaver 22.2.0 - <nehat> NORMALIZATION

File Edit Navigate Search SQL Editor Database Window Help

Auto SQL Commit Rollback T Auto nehat dbo@nehat

Database Navigator Projects

Enter a part of object name here

dlithe - rewansolution.database.windows.net:1433

nehat - rewansolution.database.windows.net:1433

Databases

nehat

Schemas

dbo

Tables

BANK_TRANSACTION 1.1M

CAR_ProductDetails 72K

CAR_SupplierAddress 72K

CAR_SupplierDetails 72K

CAR_SupplierName 72K

Car_Sale 840K

Cinema_Tickets 6.8M

DEPARTMENT 72K

DEPARTMENTFF 72K

DIM_EMPLOYEE 72K

Dim_Employees 72K

EMP 72K

EMPLOYEE 72K

Project - General

Name DataSource

Bookmarks

Diagrams

Scripts

<nehat> NORMALIZATION_INSU <nehat> NORMALIZATION <dlithe> RANK <dlithe> JOINS

```
-- INNER JOIN
-- (Returns Only Common data from other table)

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetailss AS PD
INNER JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- LEFT JOIN
-- (Returns all rows from the left table (First table) and matching rows from the right table)

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetailss AS PD
LEFT JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- RIGHT JOIN JOIN
-- (Returns all rows from the right table (Second table) and matching rows from the left table)
```

Results 1 x

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State

Provider_ID	Drug_ID	Provider_Name	Provider_Zip_Code	Provider_State
10.001	39	SOUTHEAST ALABAMA MEDICAL CENTER	36.301	AL
10.005	39	MARSHALL MEDICAL CENTER SOUTH	35.957	AL
10.006	39	ELIZA COFFEE MEMORIAL HOSPITAL	35.631	AL
10.011	39	ST VINCENTS EAST	35.235	AL
10.016	39	SHELBY BAPTIST MEDICAL CENTER	35.007	AL
10.023	39	BAPTIST MEDICAL CENTER SOUTH	36.116	AL
10.029	39	EAST ALABAMA MEDICAL CENTER AND SNF	36.801	AL

IST en Writable Smart Insert 299:34:10989 Sel:0|0

7. RIGHT JOIN

The screenshot shows the DBeaver 22.2.0 interface. The left sidebar displays the database structure for 'nehath' on 'rewansolution.database.windows.net:1433', including tables like 'BANK_TRANSACTION', 'CAR_ProductDetails', 'CAR_SupplierAddress', 'CAR_SupplierDetails', 'CAR_SupplierName', 'Car_Sale', 'Cinema_Tickets', 'DEPARTMENT', 'DEPARTMENTFF', 'Dim_Employees', 'EMP', and 'EMPLOYEE'. The main SQL editor shows a query for a RIGHT JOIN between 'ProviderDetailss' (AS PD) and 'ProviderZipCode' (AS PZ) on the 'Provider_ZipCode' column. The query is:
`-- RIGHT JOIN JOIN
--(Returns all rows from the right table (Second table) and matching rows from the left table)

--SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetailss AS PD
RIGHT JOIN ProviderZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- FULL JOIN
--(Returns all rows from the joined tables regardless of matching data)

--SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State, PZ.Provider_City
FROM ProviderDetailss AS PD
RIGHT JOIN ProviderZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- CROSS JOIN
--(Returns matrix)

--SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State, PZ.Provider_City
FROM ProviderDetailss AS PD
CROSS JOIN ProviderZipCode AS PZ`
The 'Results' tab shows 30 rows. The first 7 rows are visible in the grid view, showing data from the 'ProviderDetailss' table joined with 'ProviderZipCode'. The status bar indicates 30 rows fetched in 140ms on 2022-10-05 at 13:57:2.

8. FULL JOIN

The screenshot shows the DBeaver 22.2.0 interface. The left sidebar displays the database structure for 'nehath' on 'rewansolution.database.windows.net:1433'. The main SQL editor shows a query for a FULL JOIN between 'ProviderDetailss' (AS PD) and 'ProviderZipCode' (AS PZ) on the 'Provider_ZipCode' column. The query is:
`-- RIGHT JOIN JOIN
--(Returns all rows from the right table (Second table) and matching rows from the left table)

--SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetailss AS PD
RIGHT JOIN ProviderZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- FULL JOIN
--(Returns all rows from joined table regardless of matching data)

--SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State, PZ.Provider_City
FROM ProviderDetailss AS PD
FULL JOIN ProviderZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- CROSS JOIN
--(Returns matrix 30*30=900 ROWS)

--SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State, PZ.Provider_City
FROM ProviderDetailss AS PD
CROSS JOIN ProviderZipCode AS PZ`
The 'Results' tab shows 32 rows. The first 7 rows are visible in the grid view, showing data from the 'ProviderDetailss' table joined with 'ProviderZipCode'. The status bar indicates 32 rows fetched in 116ms on 2022-10-11 at 13:29:3.

9. CROSS JOIN

The screenshot shows the DBeaver 22.2.0 interface with the SQL Editor open. The query is as follows:

```
RIGHT JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- FULL JOIN
-- (Returns all rows from joined table regardless of matching data)

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State, PZ.Provider_City
FROM ProviderDetailss AS PD
RIGHT JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- CROSS JOIN
-- (Returns matrix 30*30=900 ROWS)

SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State, PZ.Provider_City
FROM ProviderDetailss AS PD, Provider_ZipCode AS PZ
```

The Results tab shows the following data:

Provider_ID	Drug_ID	Provider_Name	Provider_Zip_Code	Provider_State	Provider_City
895	30.002	BANNER GOOD SAMARITAN MEDICAL CENT	85,006	AZ	PRESCOTT
896	30.006	TUCSON MEDICAL CENTER	85,712	AZ	PRESCOTT
897	30.007	VERDE VALLEY MEDICAL CENTER	86,326	AZ	PRESCOTT
898	30.010	CARONDELET ST MARYS HOSPITAL	85,745	AZ	PRESCOTT
899	30.011	CARONDELET ST JOSEPHS HOSPITAL	85,711	AZ	PRESCOTT
900	30.012	YAVAPAI REGIONAL MEDICAL CENTER	86,301	AZ	PRESCOTT

10. RANK

The screenshot shows the DBeaver 22.2.0 interface with the SQL Editor open. The query is as follows:

```
INSERT INTO Provider_ZipCode VALUES
(35007, 'DOTHAN', 'AL')

-- ROW_NUMBER, RANK, DENSE_RANK
-- (ROW_NUMBER Displays the row number)
-- (RANK will be ranking according to the row number leaving the gap. If there is a tie between 1st and 2nd row, it will go like 1,1,3)
-- (DENSE_RANK never leaves the gap. If there is a tie between 1st and 2nd row, it will go like 1,1,2)

SELECT
Provider_Zip_Code,
Provider_City,
Provider_State,
ROW_NUMBER () OVER (ORDER BY Provider_Zip_Code) AS ROW_NUMBER,
RANK() OVER (ORDER BY Provider_Zip_Code) AS RANK,
DENSE_RANK() OVER (ORDER BY Provider_Zip_Code) AS DENSE_RANK
FROM Provider_ZipCode
```

The Results tab shows the following data:

Provider_Zip_Code	Provider_City	Provider_State	ROW_NUMBER	RANK	DENSE_RANK
35,007	ALABASTER	AL	1	1	1
35,007	DOTHAN	AL	2	1	1
35,205	BIRMINGHAM	AL	3	3	2
35,209	BIRMINGHAM	AL	4	4	3
35,211	BIRMINGHAM	AL	5	5	4
35,213	BIRMINGHAM	AL	6	6	5
35,233	BIRMINGHAM	AL	7	7	6

VIEW

11. View of Productdetails

The screenshot shows the DBeaver 22.2.0 interface. On the left, the Database Navigator shows the 'dbo' schema with a table 'V_ProviderDetails'. The main SQL Editor displays the following SQL script:

```
--VIEW
--(It does not store in memory, just hold the table structure which can be viewed and changed)

CREATE VIEW V_ProviderDetails
AS
SELECT PD.Provider_ID, PD.Drug_ID, PD.Provider_Name, PD.Provider_Zip_Code, PZ.Provider_State
FROM ProviderDetails AS PD
INNER JOIN Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

SELECT * FROM V_ProviderDetails

DROP VIEW V_ProviderDetails
```

The Results window shows the output of the query:

Provider_ID	Drug_ID	Provider_Name	Provider_Zip_Code	Provider_State
10.001	39	SOUTHEAST ALABAMA MEDICAL CENTER	36.301	AL
10.005	39	MARSHALL MEDICAL CENTER SOUTH	35.957	AL
10.006	39	ELIZA COFFEE MEMORIAL HOSPITAL	35.631	AL
10.011	39	ST VINCENTS EAST	35.235	AL
10.016	39	SHELBY BAPTIST MEDICAL CENTER	35.007	AL
10.016	39	SHELBY BAPTIST MEDICAL CENTER	35.007	AL
10.023	39	BAPTIST MEDICAL CENTER SOUTH	36.116	AL

REFERENTIAL INTEGRITY

12. On Delete Cascade On Update Cascade OP

The screenshot shows the DBeaver 22.2.0 interface. On the left, the Database Navigator shows the 'ProviderAddress' table with a foreign key 'FK_ID'. The main SQL Editor displays the following SQL script:

```
--REFERENTIAL INTEGRITY (To delete or update the table which is linked to other table using foreign key)
--1. ON DELETE CASCADE ON UPDATE CASCADE
--(Not only deletes or updates the PARENT TABLE but also delete or updates the CHILD TABLE data)

ALTER TABLE ProviderAddress DROP CONSTRAINT FK_ID

ALTER TABLE ProviderAddress
ADD CONSTRAINT FK_ID
FOREIGN KEY (Provider_ID)
REFERENCES ProviderDetails(Provider_ID)
ON DELETE CASCADE ON UPDATE CASCADE

DELETE FROM ProviderDetails WHERE Provider_ID = 10001
UPDATE ProviderDetails SET Drug_ID = 40 WHERE Provider_ID = 10005

SELECT * FROM ProviderDetails
```

The Results window shows the output of the query:

Provider_ID	Drug_ID	Provider_Name	Average_Medicare_Payments
10.005	40	MARSHALL MEDICAL CENTER SOUTH	4.976.714286
10.006	39	ELIZA COFFEE MEMORIAL HOSPITAL	4.453.791667
10.011	39	ST VINCENTS EAST	4.129.16
10.016	39	SHELBY BAPTIST MEDICAL CENTER	4.851.444444
10.023	39	BAPTIST MEDICAL CENTER SOUTH	5.374.149254
10.029	39	EAST ALABAMA MEDICAL CENTER AND SNF	4.761.411765
10.033	39	UNIVERSITY OF ALABAMA HOSPITAL	5.858.5

13. On Delete Set Null On Update Set Null OP

DBeaver 22.2.0 - <nehat> NORMALIZATION

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto nehat dbo@nehat

<nehat> NORMALIZATION_neha <nehat> NORMALIZATION_INSU <nehat> Script-7 <nehat> NORMALIZATION

```
--1. ON DELETE SET NULL ON UPDATE SET NULL
--(On deleting or updating the PARENT TABLE DATA the foreign key of deleted one will be set to NULL in the CHILD TABLE)

ALTER TABLE ProviderAddress DROP CONSTRAINT FK_ID

=ALTER TABLE ProviderAddress
ADD CONSTRAINT FK_ID
FOREIGN KEY (Provider_ID)
REFERENCES ProviderDetails(Provider_ID)
ON DELETE SET NULL ON UPDATE SET NULL

=DELETE FROM ProviderDetails WHERE Provider_ID = 10005
UPDATE ProviderDetails SET Drug_ID = 40 WHERE Provider_ID = 10006

=SELECT * FROM ProviderDetails -- PARENT TABLE
SELECT * FROM ProviderAddress -- CHILD TABLE
*****
```

Results 1

SELECT * FROM ProviderAddress Enter a SQL expression to filter results (use Ctrl+Space)

Provider_ID	Provider_Street_Address
1	[NULL] 2505 U S HIGHWAY 431 NORTH
2	10,006 205 MARENGO STREET
3	10,011 50 MEDICAL PARK EAST DRIVE
4	10,016 1000 FIRST STREET NORTH
5	10,023 2105 EAST SOUTH BOULEVARD
6	10,029 2000 PEPPERELL PARKWAY
7	10,033 619 SOUTH 19TH STREET

Value

Save Cancel Script 200 Rows: 1 29 row(s) fetched - 143ms, on 2022-10-05 at 10:43:3

IST en Writable Smart Insert 135:1 [44] Sel: 44 | 1

14. On Delete Set Null On Update Set Null

DBeaver 22.2.0 - <nehat> NORMALIZATION

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto nehat dbo@nehat

<nehat> NORMALIZATION_neha <nehat> NORMALIZATION_INSU <nehat> Script-7 <nehat> NORMALIZATION

```
--1. ON DELETE SET NULL ON UPDATE SET NULL
--(On deleting or updating the PARENT TABLE DATA the foreign key of deleted one will be set to NULL in the CHILD TABLE)

ALTER TABLE ProviderAddress DROP CONSTRAINT FK_ID

=ALTER TABLE ProviderAddress
ADD CONSTRAINT FK_ID
FOREIGN KEY (Provider_ID)
REFERENCES ProviderDetails(Provider_ID)
ON DELETE SET NULL ON UPDATE SET NULL

=DELETE FROM ProviderDetails WHERE Provider_ID = 10005
UPDATE ProviderDetails SET Drug_ID = 40 WHERE Provider_ID = 10006

=SELECT * FROM ProviderDetails -- PARENT TABLE
SELECT * FROM ProviderAddress -- CHILD TABLE
*****
```

Statistics 1

Name	Value
Updated Rows	2
Query	DELETE FROM ProviderDetails WHERE Provider_ID = 10005 UPDATE ProviderDetails SET Drug_ID = 40 WHERE Provider_ID = 10006
Finish time	Wed Oct 05 10:41:44 IST 2022

Save Cancel Script 200 Rows: 1 2 row(s) updated - 47ms, on 2022-10-05 at 10:41:44

IST en Writable Smart Insert 131:1 [122] Sel: 122 | 2