MODULE 1 – SQL SERVER

Data Source 1: Inpatient_provdr_Covid_Healthcare

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

--2nd Normalization Form

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

--Drug ProviderDetails Table

```
CREATE TABLE ProviderDetails -- PARENT TABLE
(
	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

DROP TABLE ProviderDetails

INSERT INTO ProviderDetails VALUES
(10001, 39,  'SOUTHEAST ALABAMA MEDICAL CENTER',     4763.736264),
(10005, 39,  'MARSHALL MEDICAL CENTER SOUTH', 4976.714286),
(10006, 39,  'ELIZA COFFEE MEMORIAL HOSPITAL',       4453.791667),
(10011, 39,  'ST VINCENTS EAST', 4129.16),
(10016, 39,  'SHELBY BAPTIST MEDICAL CENTER', 4851.444444),
(10023,      39, 'BAPTIST MEDICAL CENTER SOUTH',     5374.149254),
(10029, 39,  'EAST ALABAMA MEDICAL CENTER AND SNF', 4761.411765),
(10033, 39,  'UNIVERSITY OF ALABAMA HOSPITAL',       5858.5),
(10039, 39,  'HUNTSVILLE HOSPITAL',     5228.4),
(10040, 39,  'GADSDEN REGIONAL MEDICAL CENTER',      4386.941176),
(10046, 39,  'RIVERVIEW REGIONAL MEDICAL CENTER',    4493.571429),
(10055, 39,  'FLOWERS HOSPITAL', 4408.2),
(10056, 39,  'ST VINCENTSS BIRMINGHAM',4186.023256),
(10078, 39,  'NORTHEAST ALABAMA REGIONAL MED CENTER',      4376.238095),
(10083, 39,  'SOUTH BALDWIN REGIONAL MEDICAL CENTER',      4383.733333),
(10085, 39,  'DECATUR GENERAL HOSPITAL',      4509.111111),
(10090, 39,  'PROVIDENCE HOSPITAL',     3972.851852),
(10092, 39,  'D C H REGIONAL MEDICAL CENTER', 5179.387097),
(10100, 39,  'THOMAS HOSPITAL',  3898.888889),
(10103, 39,  'BAPTIST MEDICAL CENTER-PRINCETON',     4962.454545),
(10104, 39,  'TRINITY MEDICAL CENTER', 4471.689655),
(10113, 39,  'MOBILE INFIRMARY', 4219.909091),
```

```
(10139, 39,  'BROOKWOOD MEDICAL CENTER',      3944.421053),
(20017, 39,  'ALASKA REGIONAL HOSPITAL',      6413.782609),
(30002, 39,  'BANNER GOOD SAMARITAN MEDICAL CENTER',6951.454545),
(30006, 39,  'TUCSON MEDICAL CENTER',  5764.875),
(30007, 39,  'VERDE VALLEY MEDICAL CENTER',   8008.111111),
(30010, 39,  'CARONDELET ST  MARYS HOSPITAL', 5379.833333),
(30011, 39,  'CARONDELET ST JOSEPHS HOSPITAL',      4903.333333),
(30012, 39,  'YAVAPAI REGIONAL MEDICAL CENTER',     6133.571429)

--Drug ProviderAddress Table

CREATE TABLE ProviderAddress -- CHILD TABLE
(
      Provider_ID FLOAT CONSTRAINT FK_ID 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'),
(10033,      '619 SOUTH 19TH STREET'),
(10039,      '101 SIVLEY RD'),
(10040,      '1007 GOODYEAR AVENUE'),
(10046,      '600 SOUTH THIRD STREET'),
(10055,      '4370 WEST MAIN STREET'),
(10056,      '810 ST VINCENTS DRIVE'),
(10078,      '400 EAST 10TH STREET'),
(10083,      '1613 NORTH MCKENZIE STREET'),
(10085,      '1201 7TH STREET SE'),
(10090,      '6801 AIRPORT BOULEVARD'),
(10092,      '809 UNIVERSITY BOULEVARD EAST'),
(10100,      '750 MORPHY AVENUE'),
(10103,      '701 PRINCETON AVENUE SOUTHWEST'),
(10104,      '800 MONTCLAIR RD'),
(10113,      '5 MOBILE INFIRMARY CIRCLE'),
(10139,      '2010 BROOKWOOD MEDICAL CENTER DRIVE'),
(20017,      '2801 DEBARR ROAD'),
(30002,      '1111 EAST MCDOWELL ROAD'),
(30006,      '5301 EAST GRANT ROAD'),
(30007,      '269 SOUTH CANDY LANE'),
(30010,      '1601 WEST ST MARYS ROAD'),
(30011,      '350 NORTH WILMOT ROAD'),
(30012,      '1003 WILLOW CREEK ROAD')
```

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

--**REFERNETIAL 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
```

--2. 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
```

```sql
--3. ON DELETE SET DEFAULT ON UPDATE SET DEFAULT
--(On deleting or updating the PARENT TABLE DATA the foreign key of deleted one will
be set to DEFAULT value in the CHILD TABLE)

CREATE TABLE ProviderAddress -- CHILD TABLE
(
	Provider_ID FLOAT DEFAULT 1 CONSTRAINT FK_ID
	REFERENCES ProviderDetails(Provider_ID) ON DELETE SET DEFAULT ON UPDATE SET
	DEFAULT ,
	Provider_Street_Address VARCHAR(50) UNIQUE
)

SELECT * FROM ProviderAddress

DROP TABLE ProviderAddress

DELETE FROM ProviderDetails WHERE Provider_ID = 30012
UPDATE ProviderDetails SET Drug_ID = 40 WHERE Provider_ID = 30011

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

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'),
(10033,	'619 SOUTH 19TH STREET'),
(10039,	'101 SIVLEY RD'),
(10040,	'1007 GOODYEAR AVENUE'),
(10046,	'600 SOUTH THIRD STREET'),
(10055,	'4370 WEST MAIN STREET'),
(10056,	'810 ST VINCENTS DRIVE'),
(10078,	'400 EAST 10TH STREET'),
(10083,	'1613 NORTH MCKENZIE STREET'),
(10085,	'1201 7TH STREET SE'),
(10090,	'6801 AIRPORT BOULEVARD'),
(10092,	'809 UNIVERSITY BOULEVARD EAST'),
(10100,	'750 MORPHY AVENUE'),
(10103,	'701 PRINCETON AVENUE SOUTHWEST'),
(10104,	'800 MONTCLAIR RD'),
(10113,	'5 MOBILE INFIRMARY CIRCLE'),
(10139,	'2010 BROOKWOOD MEDICAL CENTER DRIVE'),
(20017,	'2801 DEBARR ROAD'),
(30002,	'1111 EAST MCDOWELL ROAD'),
(30006,	'5301 EAST GRANT ROAD'),
(30007,	'269 SOUTH CANDY LANE'),
(30010,	'1601 WEST ST MARYS ROAD'),
(30011,	'350 NORTH WILMOT ROAD'),
(30012,	'1003 WILLOW CREEK ROAD')
```

--3rd Normalization Form

--Inpatient_provdr_Covid_Healthcare Table can be Split into 2 by separting Provider_ID, Drug_ID, Provider_Name, Provider_Zip_Code in one table and Provider_Zip_Code, Provider_City, Provider_State in another table

--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 and it breaks the 3NF rule

--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),
(10005, 39,  'MARSHALL MEDICAL CENTER SOUTH', 35957),
(10006, 39,  'ELIZA COFFEE MEMORIAL HOSPITAL',       35631),
(10011, 39,  'ST VINCENTS EAST', 35235),
(10016, 39,  'SHELBY BAPTIST MEDICAL CENTER', 35007),
(10023,      39, 'BAPTIST MEDICAL CENTER SOUTH',     36116),
(10029, 39,  'EAST ALABAMA MEDICAL CENTER AND SNF', 36801),
(10033, 39,  'UNIVERSITY OF ALABAMA HOSPITAL',       35233),
(10039, 39,  'HUNTSVILLE HOSPITAL',     35801),
(10040, 39,  'GADSDEN REGIONAL MEDICAL CENTER',      35903),
(10046, 39,  'RIVERVIEW REGIONAL MEDICAL CENTER',    35901),
(10055, 39,  'FLOWERS HOSPITAL', 36305),
(10056, 39,  'ST VINCENTSS BIRMINGHAM',35205),
(10078, 39,  'NORTHEAST ALABAMA REGIONAL MED CENTER',      36207),
(10083, 39,  'SOUTH BALDWIN REGIONAL MEDICAL CENTER',      36535),
(10085, 39,  'DECATUR GENERAL HOSPITAL',      35609),
(10090, 39,  'PROVIDENCE HOSPITAL',     36608),
(10092, 39,  'D C H REGIONAL MEDICAL CENTER', 35401),
(10100, 39,  'THOMAS HOSPITAL',  36532),
(10103, 39,  'BAPTIST MEDICAL CENTER-PRINCETON',     35211),
(10104, 39,  'TRINITY MEDICAL CENTER', 35213),
(10113, 39,  'MOBILE INFIRMARY', 36652),
(10139, 39,  'BROOKWOOD MEDICAL CENTER',      35209),
(20017, 39,  'ALASKA REGIONAL HOSPITAL',      99508),
(30002, 39,  'BANNER GOOD SAMARITAN MEDICAL CENTER',85006),
(30006, 39,  'TUCSON MEDICAL CENTER',  85712),
(30007, 39,  'VERDE VALLEY MEDICAL CENTER',   86326),
(30010, 39,  'CARONDELET ST  MARYS HOSPITAL', 85745),
(30011, 39,  'CARONDELET ST JOSEPHS HOSPITAL',       85711),
(30012, 39,  'YAVAPAI REGIONAL MEDICAL CENTER',      86301)
```

```sql
--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'),
(35233,	'BIRMINGHAM','AL'),
(35801,	'HUNTSVILLE','AL'),
(35903,	'GADSDEN',	'AL'),
(35901,	'GADSDEN',	'AL'),
(36305,	'DOTHAN',	'AL'),
(35205,	'BIRMINGHAM','AL'),
(36207,	'ANNISTON',	'AL'),
(36535,	'FOLEY',	'AL'),
(35609,	'DECATUR',	'AL'),
(36608,	'MOBILE',	'AL'),
(35401,	'TUSCALOOSA','AL'),
(36532,	'FAIRHOPE',	'AL'),
(35211,	'BIRMINGHAM','AL'),
(35213,	'BIRMINGHAM','AL'),
(36652,	'MOBILE',	'AL'),
(35209,	'BIRMINGHAM','AL'),
(99508,	'ANCHORAGE',	'AK'),
(85006,	'PHOENIX',	'AZ'),
(85712,	'TUCSON',	'AZ'),
(86326,	'COTTONWOOD','AZ'),
(85745,	'TUCSON',	'AZ'),
(85711,	'TUCSON',	'AZ'),
(86301,	'PRESCOTT',	'AZ')
```

```sql
DELETE FROM Provider_ZipCode WHERE Provider_Zip_Code=36301

--JOINS

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

```sql
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)

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

-- FULL JOIN
--(Returs 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 Provider_ZipCode AS PZ
ON PD.Provider_Zip_Code = PZ.Provider_Zip_Code

-- CROSS JOIN
--(Returs matix 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
```

```
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
```

```
--VIEW

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

CREATE VIEW V_ProviderDetailss
AS
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

SELECT * FROM V_ProviderDetailss

DROP VIEW V_ProviderDetailss
```

Data Source 2: Incurance Claim_State_Health_Data

--1st Normalization Form: Table is already in 1st NF

--2nd Normalization Form

```
--Incurance Claim_State_Health_Data Table can be Split into 2 by separting PID,
GENDER, AGE, HOSPITAL_NAME, SURGURY_CODE in one table and SURGURY_CODE, SURGERY in
another table
--It is because SURGERY is dependent on SURGERY_CODE and it breaks the 2NF rule
```

-- PatientDetails Table

```
CREATE TABLE PatientDetails
(
	PID INT PRIMARY KEY,
	AGE INT,
	GENDER VARCHAR(50),
	HOSPITAL_NAME VARCHAR(50),
	SURGERY_CODE VARCHAR(50)
)

SELECT * FROM PatientDetails

INSERT INTO PatientDetails VALUES
(1,	56,	'Female',	'Rims Govt. General Hospital, Srikakulam',	'M6.5'),
(2,	37,	'Male',	'Rims Govt. General Hospital, Srikakulam',	'M6.5'),
(3,	50,	'Male',	'Rims Govt. General Hospital, Srikakulam',	'M6.5'),
(4,	45,	'Male',	'Rims Govt. General Hospital, Srikakulam',	'M6.5'),
(5,	54,	'Male',	'Rims Govt. General Hospital, Srikakulam',	'M6.5'),
(6,	35,	'Male',	'Rims Govt. General Hospital, Srikakulam',	'M6.5'),
(7,	52,	'Male',	'Govt General Hospital Kurnool', 'M6.5'),
(8,	73,	'Male',	'Queens Nri Hospitals',	'M6.5'),
(9,	56,	'Male',	'Karumuri Hospital',	'S7.1.1.1'),
(10,	49,	'Male',	'Karumuri Hospital',	'S7.2.1.1'),
(11,	52,	'Male',	'Queens Nri Hospitals',	'M6.5'),
(12,	56,	'Male',	'Karumuri Hospital',	'M5.1.2'),
(13,	65,	'Female',	'Karumuri Hospital',	'M5.1.5'),
(14,	75,	'Male',	'Karumuri Hospital',	'M5.1.5'),
(15,	52,	'Male',	'Karumuri Hospital',	'M5.1.5'),
(16,	56,	'Male',	'Karumuri Hospital',	'M5.1.2'),
(17,	1,	'Male(Child)',	'Ent Nursing Home', 'S16.1.1'),
(18,	54,	'Female',	'Karumuri Hospital',	'S7.2.1.1'),
(19,	48,	'Male',	'Karumuri Hospital',	'S7.2.1.1'),
(20,	57,	'Female',	'Karumuri Hospital',	'M5.1.2'),
(21,	55,	'Male',	'Queens Nri Hospitals',	'M6.5'),
(22,	74,	'Female',	'KIMS SAI SESHDRI HOSPITAL',	'M5.1.2'),
(23,	57,	'Male',	'KIMS SAI SESHDRI HOSPITAL',	'M5.1.2'),
(24,	56,	'Male',	'Karumuri Hospital',	'S7.2.1.1'),
(25,	31,	'Male',	'KIMS SAI SESHDRI HOSPITAL',	'S5.1.2'),
(26,	68,	'Male',	'KIMS SAI SESHDRI HOSPITAL',	'M5.1.2'),
(27,	72,	'Female',	'KIMS SAI SESHDRI HOSPITAL',	'M5.1.2'),
```

```
(28,   66,    'Female',    'Karumuri Hospital',      'M5.1.2'),
(29,   35,    'Male',      'KIMS SAI SESHDRI HOSPITAL',     'M5.1.2'),
(30,   72,    'Female',    'KIMS SAI SESHDRI HOSPITAL',     'M5.1.2')


-- SurgeryDetails Table

CREATE TABLE SurgeryDetails
(
      SURGERY_CODE VARCHAR(50),
      SURGERY VARCHAR(100)
)

SELECT * FROM SurgeryDetails

INSERT INTO SurgeryDetails VALUES
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('S7.1.1.1', 'Coronary Balloon Angioplasty with stent(00.45)'),
('S7.2.1.1', 'Coronary Bypass Surgery'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M5.1.5',   'Medical Management of  Refractory Cardiac Failure'),
('M5.1.5',   'Medical Management of  Refractory Cardiac Failure'),
('M5.1.5',   'Medical Management of  Refractory Cardiac Failure'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('S16.1.1',  'Cochlear Implant Surgery'),
('S7.2.1.1', 'Coronary Bypass Surgery'),
('S7.2.1.1', 'Coronary Bypass Surgery'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M6.5',     'Maintenance Hemodialysis For Crf'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('S7.2.1.1', 'Coronary Bypass Surgery'),
('S5.1.2',   'Excision Or Other Operations For Scaphoid Fractures'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M5.1.2',   'Management Of Acute MI With Angiogram'),
('M5.1.2',   'Management Of Acute MI With Angiogram')
```

--3rd Normalization Form

--Incurance Claim_State_Health_Data Table can be Split into 2 by separting PID, AGE, GENDER, DISTRICT_NAME, CATEGORY_CODE in one table and CATEGORY_CODE, CATEGORY_NAME, SURGURY_CODE, SURGERY in another table

--It is because CATEGORY_CODE is dependent on PID that makes non primary attributes (CATEGORY_NAME, SURGERY_CODE, SURGERY) transitively dependent on PID and it breaks the 3NF rule

--PatientDetailss Table

```
CREATE TABLE PatientDetailss
(
	PID INT PRIMARY KEY,
	AGE INT,
	GENDER VARCHAR(50),
	DISTRICT_NAME VARCHAR(50),
	CATEGORY_CODE VARCHAR(50)
)

SELECT * FROM PatientDetailss

INSERT INTO PatientDetailss VALUES
(1,	56,	'Female',	'Srikakulam','M6'),
(2,	37,	'Male',	'Srikakulam','M6'),
(3,	50,	'Male',	'Srikakulam','M6'),
(4,	45,	'Male',	'Srikakulam','M6'),
(5,	54,	'Male',	'Srikakulam','M6'),
(6,	35,	'Male',	'Srikakulam','M6'),
(7,	52,	'Male',	'Kurnool',	'M6'),
(8,	73,	'Male',	'Vizianagaram',	'M6'),
(9,	56,	'Male',	'Guntur',	'S7'),
(10,	49,	'Male',	'Guntur',	'S7'),
(11,	52,	'Male',	'Vishakhapatnam',	'M6'),
(12,	56,	'Male',	'Guntur',	'M5'),
(13,	65,	'Female',	'Guntur',	'M5'),
(14,	75,	'Male',	'Guntur',	'M5'),
(15,	52,	'Male',	'Guntur',	'M5'),
(16,	56,	'Male',	'Guntur',	'M5'),
(17,	1,	'Male(Child)',	'Guntur',	'S16'),
(18,	54,	'Female',	'Guntur',	'S7'),
(19,	48,	'Male',	'Guntur',	'S7'),
(20,	57,	'Female',	'Guntur',	'M5'),
(21,	55,	'Male',	'West Godavari',	'M6'),
(22,	74,	'Female',	'Srikakulam','M5'),
(23,	57,	'Male',	'Srikakulam','M5'),
(24,	56,	'Male',	'Guntur',	'S7'),
(25,	31,	'Male',	'Srikakulam','S5'),
(26,	68,	'Male',	'Srikakulam','M5'),
(27,	72,	'Female',	'Srikakulam','M5'),
(28,	66,	'Female',	'Guntur',	'M5'),
(29,	35,	'Male',	'Srikakulam','M5'),
(30,	72,	'Female',	'Srikakulam','M5')
```

```sql
--SurgeryDetailss Table

CREATE TABLE SurgeryDetailss
(
	CATEGORY_CODE VARCHAR(50),
	CATEGORY_NAME VARCHAR(50),
	SURGERY_CODE VARCHAR(50),
	SURGERY VARCHAR(100)
)

SELECT * FROM SurgeryDetailss

INSERT INTO SurgeryDetailss VALUES
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('S7', 'CARDIAC AND CARDIOTHORACIC SURGERY',	'S7.1.1.1',	'Coronary Balloon Angioplasty with stent(00.45)'),
('S7', 'CARDIAC AND CARDIOTHORACIC SURGERY',	'S7.2.1.1',	'Coronary Bypass Surgery'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M5', 'CARDIOLOGY','M5.1.5',	'Medical Management of  Refractory Cardiac Failure'),
('M5', 'CARDIOLOGY','M5.1.5',	'Medical Management of  Refractory Cardiac Failure'),
('M5', 'CARDIOLOGY','M5.1.5',	'Medical Management of  Refractory Cardiac Failure'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('S16',	'COCHLEAR IMPLANT SURGERY',	'S16.1.1',	'Cochlear Implant Surgery'),
('S7', 'CARDIAC AND CARDIOTHORACIC SURGERY',	'S7.2.1.1',	'Coronary Bypass Surgery'),
('S7', 'CARDIAC AND CARDIOTHORACIC SURGERY',	'S7.2.1.1',	'Coronary Bypass Surgery'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M6', 'NEPHROLOGY','M6.5',	'Maintenance Hemodialysis For Crf'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('S7', 'CARDIAC AND CARDIOTHORACIC SURGERY',	'S7.2.1.1',	'Coronary Bypass Surgery'),
('S5', 'ORTHOPEDIC  SURGERY AND PROCEDURES',	'S5.1.2',	'Excision Or Other Operations For Scaphoid Fractures'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram'),
('M5', 'CARDIOLOGY','M5.1.2',	'Management Of Acute MI With Angiogram')
```

```
--JOINS

-- INNER JOIN
--(Returns Only Common data from another table)

SELECT Pat.PID, Pat.AGE, Pat.GENDER, Pat.DISTRICT_NAME, Pat.CATEGORY_CODE,
Sur.SURGERY
FROM PatientDetailss AS Pat
INNER JOIN SurgeryDetailss AS Sur
ON Pat.CATEGORY_CODE = Sur.CATEGORY_CODE

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

SELECT Pat.PID, Pat.AGE, Pat.GENDER, Pat.DISTRICT_NAME, Pat.CATEGORY_CODE,
Sur.SURGERY
FROM PatientDetailss AS Pat
LEFT JOIN SurgeryDetailss AS Sur
ON Pat.CATEGORY_CODE = Sur.CATEGORY_CODE

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

SELECT Pat.PID, Pat.AGE, Pat.GENDER, Pat.DISTRICT_NAME, Pat.CATEGORY_CODE,
Sur.SURGERY
FROM PatientDetailss AS Pat
RIGHT JOIN SurgeryDetailss AS Sur
ON Pat.CATEGORY_CODE = Sur.CATEGORY_CODE

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

SELECT Pat.PID, Pat.AGE, Pat.GENDER, Pat.DISTRICT_NAME, Pat.CATEGORY_CODE,
Sur.SURGERY_CODE, Sur.SURGERY
FROM PatientDetailss AS Pat
FULL JOIN SurgeryDetailss AS Sur
ON Pat.CATEGORY_CODE = Sur.CATEGORY_CODE

-- CROSS JOIN
--(Returs matix 30*30=900 ROWS)

SELECT Pat.PID, Pat.AGE, Pat.GENDER, Pat.DISTRICT_NAME, Pat.CATEGORY_CODE,
Sur.SURGERY_CODE, Sur.SURGERY
FROM PatientDetailss AS Pat, SurgeryDetailss AS Sur
```

--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
	CATEGORY_CODE,
	CATEGORY_NAME,
	SURGERY_CODE,
	SURGERY,
	ROW_NUMBER () OVER (ORDER BY CATEGORY_CODE) AS ROW_NUMBER,
	RANK() OVER (ORDER BY CATEGORY_CODE) AS RANK,
	DENSE_RANK() OVER (ORDER BY CATEGORY_CODE) AS DENSE_RANK
FROM SurgeryDetailss
```

--VIEW

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

CREATE VIEW V_PatientDetailss
AS
SELECT Pat.PID, Pat.AGE, Pat.GENDER, Pat.DISTRICT_NAME, Pat.CATEGORY_CODE,
Sur.SURGERY
FROM PatientDetailss AS Pat
INNER JOIN SurgeryDetailss AS Sur
ON Pat.CATEGORY_CODE = Sur.CATEGORY_CODE

SELECT * FROM V_PatientDetailss

DROP VIEW V_PatientDetailss
```