Part 1

1. Summary and Business Case

1.1. Background

Health insurance company XYZ Inc. send paid claims data to their compliance and risk analysis department to 1) fulfil compliance reporting needs
2) Perform population health analysis to aid in creating tailored insurance products for member's needs
3) To help with designing member outreach program.
Compliance and reporting department is also responsible for maintaining systematic access to this historical information to and be able provide trend analysis capabilities as required.

1.2. Scope statement

Create a relational database to store paid claims details which can provide ability to systematically access historical information for reporting purposes and risk analysis purposes

1.3. Business case

Storing data in flat files generates lots of overhead and also requires significant manual effort to locate specific data required for reporting purposes. In recent years, there have been many instances where inaccurate and/or incomplete data was reported, which did not match similar reports generated from upstream applications thus, raising serious concerns about ability to provide consistent accurate reporting for future product design.

A relational database which can store and provide easy systematic access to historical data, will reduce manual intervention needed in maintaining and searching datasets and hence reducing chances of errors resulting in incomplete and/or inaccurate reporting.

2. Conceptual Model

2.1. Analysis

Looking at sample datasets received by department it appears that it contains below different types of information –

a) Member data
b) Provider data
c) Episode data
d) Claims data
e) Payment data

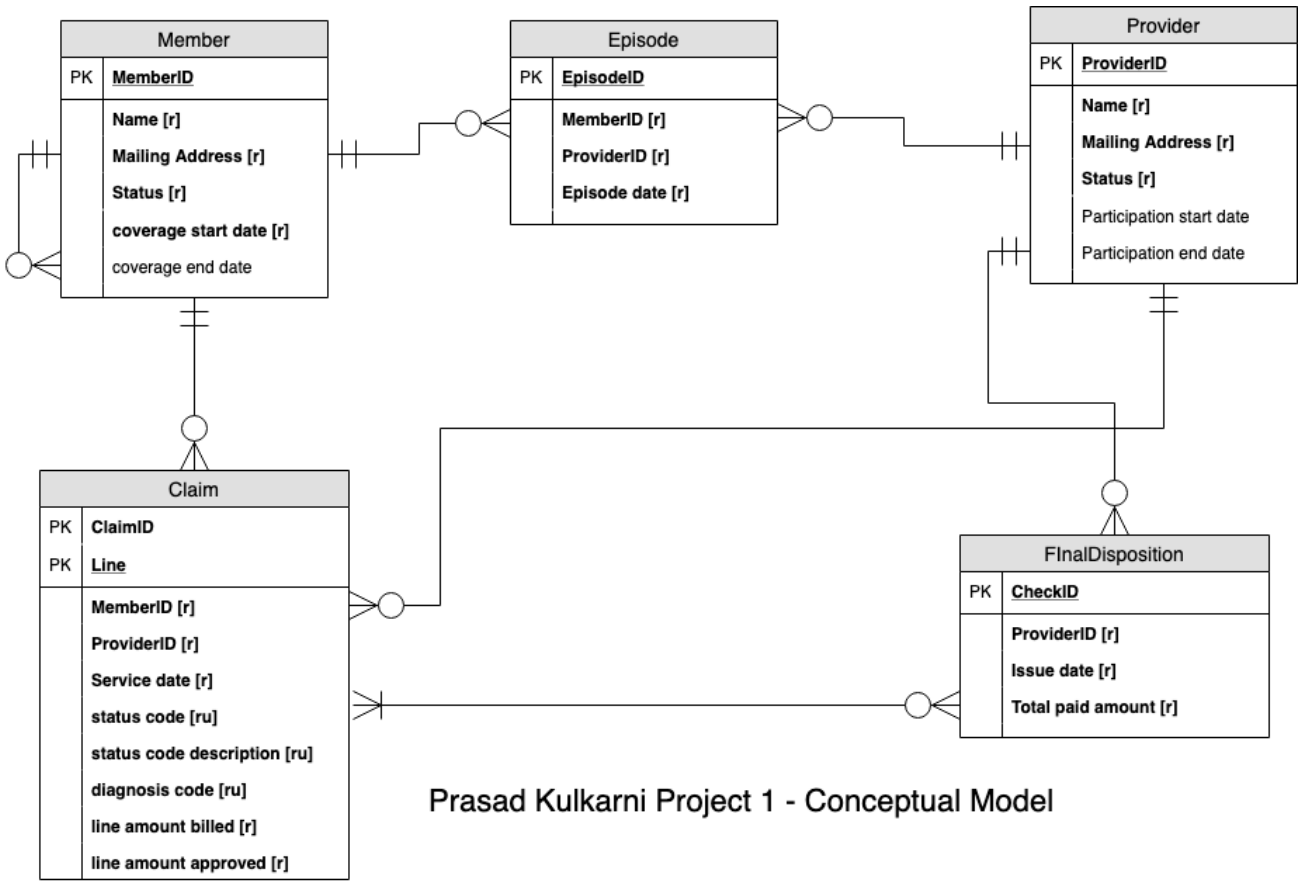
2.2. Business Rules

Each member data segment requires name, address, status (enrolled, active, terminated), coverage start and end dates. Each provider data segment requires name, address, status (in network/out of network) and optionally can have participation start and end date in case they are in network. Episode represents member and provider interaction in an outpatient setting. Along with member and provider details, episode has episode-date which represents earliest date of member and provider interaction for a specific episode. This date can be different than claims service date. Claims data segment requires service date, settlement date, status (approved, denied, pending, duplicate), multiple line items with diagnosis code on each claim line, amount billed and approved amount per claim line. Payment data is specific to a provider and requires check number, check issue date and amount

After reviewing data sets in further details and discussing with upstream application below set of relationship rules emerge –

a) Each Member can have zero or many dependents.
b) Each member can visit zero or many providers can have multiple recorded health episodes
c) Each provider can submit zero or many claims
d) Each payment check can be associated with one or more claims
e) Each claim can have zero or many payment checks associated with it.
f) Each payment check is for one and only one provider

2.3. Conceptual model diagram



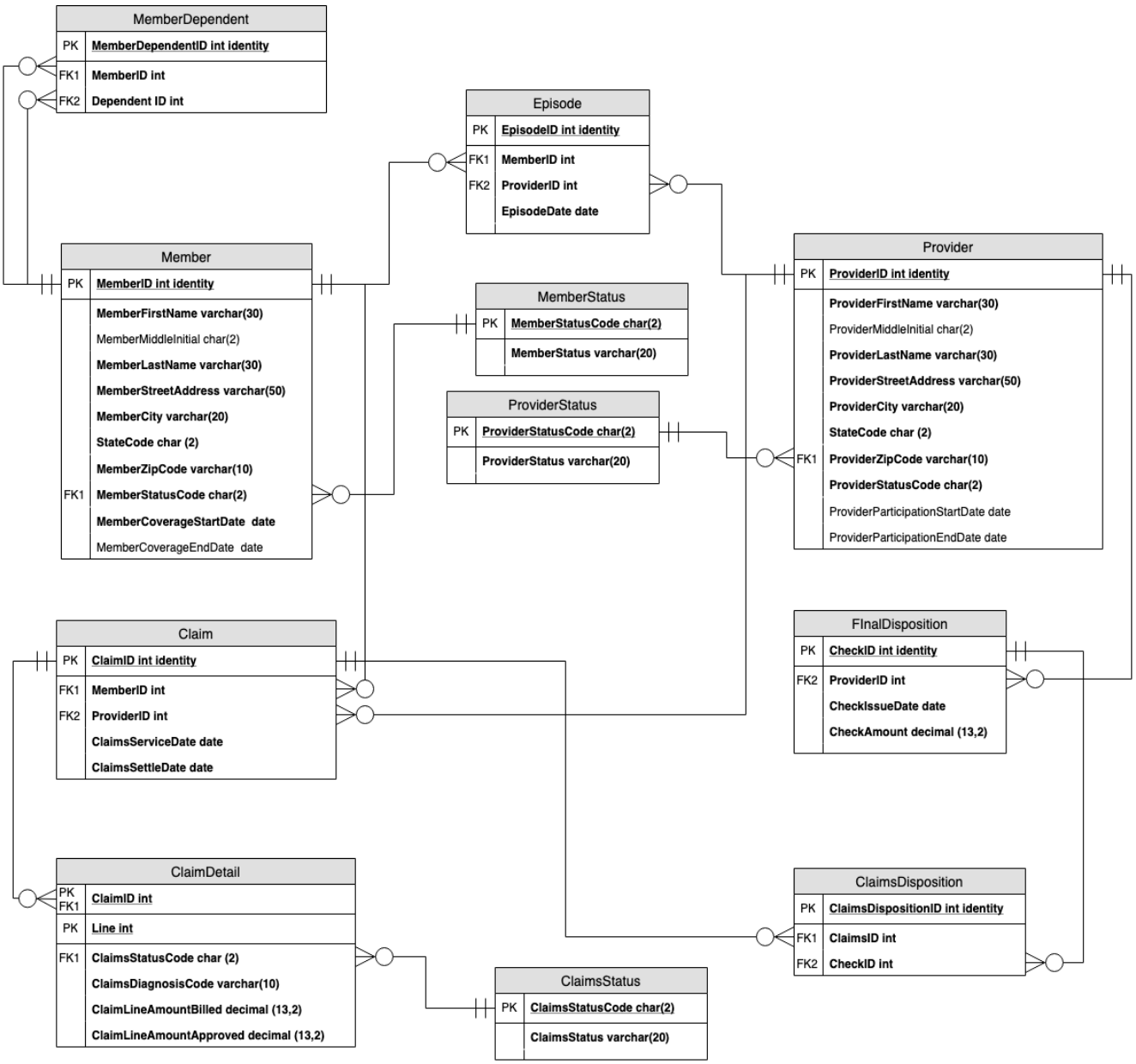
3. Normalized Logical Model

3.1. Normalization summary

This logical model relies on surrogate keys as primary keys.

a) To normalize member data a separate bridge table can be created to list member and dependent IDs
b) To normalize claims data separate claims detail table can be created to store line details vs claims header details can be stored in claims table
c) A separate bridge table between final disposition and claims table can be created to list multiple claims associated with a payment check and also list multiple checks associated with a claim
d) Finally, separate member status, provider status and claims status look up table can be created to resolve update and delete anomalies.

3.2. Normalized logical model diagram



Prasad Kulkarni Project 1 - Logical Model

This database can be used by healthcare organization to conduct analysis on paid claims. Some of the queries that can be run from database are –

- Number of claims for a member since member's enrollment
- Amount paid in approved claims for a Member or it's dependent
- Number of dependents enrolled in plan
- Number of patients served by a Physician
- Amount of approved claim charges for a physician.

Part 2:

4. Physical Database Design

Changes: After further review and analysis, changes were made to normalized logical data model. Bridge table "Episode" and child table "Dependent" were dropped and its details were included in claim detail and Member table.

```
/*

Author: Prasad Kulkarni
Course: IST659 Project
Term: October 2019

*/
-- Begin Delete tables in Memory
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'ClaimDisposition$')
BEGIN
	DROP TABLE ClaimDisposition$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'FinalDisposition$')
BEGIN
	DROP TABLE FinalDisposition$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'ClaimDetail$')
BEGIN
	DROP TABLE ClaimDetail$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Claim$')
BEGIN
	DROP TABLE Claim$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Provider$')
BEGIN
	DROP TABLE Provider$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Member$')
BEGIN
	DROP TABLE Member$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'ClaimStatus$')
BEGIN
	DROP TABLE ClaimStatus$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'MemberStatus$')
BEGIN
	DROP TABLE MemberStatus$
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'ProviderStatus$')
BEGIN
	DROP TABLE ProviderStatus$
```

```
END
-- End Delete tables in Memory

-- Begin Delete tales in database
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_ClaimDisposition')
BEGIN
	DROP TABLE HCI_ClaimDisposition
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_FinalDisposition')
BEGIN
	DROP TABLE HCI_FinalDisposition
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_ClaimDetail')
BEGIN
	DROP TABLE HCI_ClaimDetail
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_Claim')
BEGIN
	DROP TABLE HCI_Claim
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_Provider')
BEGIN
	DROP TABLE HCI_Provider
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_Member')
BEGIN
	DROP TABLE HCI_Member
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_ClaimStatus')
BEGIN
	DROP TABLE HCI_ClaimStatus
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_MemberStatus')
BEGIN
	DROP TABLE HCI_MemberStatus
END
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'HCI_ProviderStatus')
BEGIN
	DROP TABLE HCI_ProviderStatus
END
-- End Delete tables in database

 -- Creating the ProviderStatus table
CREATE TABLE HCI_ProviderStatus (
   -- Columns for table
   ProviderStatusCode char(2),
   ProviderStatus varchar(20) not null,
      -- Constraints on the table
   CONSTRAINT PK_ProviderStatus PRIMARY KEY (ProviderStatusCode)
      )
   -- End Creating the Table

  -- Creating the MemberStatus table
   CREATE TABLE HCI_MemberStatus (
   -- Columns for table
   MemberStatusCode char(2),
   MemberStatus varchar(20) not null,
      -- Constraints on the table
```

```
   CONSTRAINT PK_MemberStatus PRIMARY KEY (MemberStatusCode)
      )
   -- End Creating the Table


-- Creating the ClaimStatus table
   CREATE TABLE HCI_ClaimStatus (
   -- Columns for table
   ClaimStatusCode char(2),
   ClaimStatus varchar(20) not null,
      -- Constraints on the table
   CONSTRAINT PK_ClaimStatus PRIMARY KEY (ClaimStatusCode)
      )
   -- End Creating the Table

   -- Creating the Member table
   CREATE TABLE HCI_Member (
   -- Columns for the MemberData table
   MemberID int identity,
   MemberFirstName varchar(30) not null,
   MemberMiddleInitial char(2),
   MemberLastName varchar(30) not null,
   MemberStreetName varchar(50) not null,
   MemberCity varchar(20) not null,
   StateCode char(2) not null,
   MemberZipCode varchar(10) not null,
   MemberStatusCode char(2) not null,
   MemberCoverageStartDate datetime not null,
   MemberCoverageEndDate datetime,
   MemberSubscriberID int,
      -- Constraints on the MemberData Table
   CONSTRAINT PK_Member PRIMARY KEY (MemberID),
   CONSTRAINT FK1_Member FOREIGN KEY (MemberSubscriberID) REFERENCES HCI_Member(MemberID),
   CONSTRAINT FK2_Member FOREIGN KEY (MemberStatusCode) REFERENCES
HCI_MemberStatus(MemberStatusCode)
   )
   -- End Creating the MemberData Table

   -- Creating the ProviderData table
   CREATE TABLE HCI_Provider (
   -- Columns for the ProviderData table
   ProviderID int identity,
   ProviderFirstName varchar(30) not null,
   ProviderMiddleInitial char(2),
   ProviderLastName varchar(30) not null,
   ProviderStreetName varchar(50) not null,
   ProviderCity varchar(20) not null,
   StateCode char(2) not null,
   ProviderZipCode varchar(10) not null,
   ProviderStatusCode char(2) not null,
   ProviderParticipationStartDate datetime,
   ProviderParticipationEndDate datetime,
      -- Constraints on the ProviderData Table
   CONSTRAINT PK_Provider PRIMARY KEY (ProviderID),
   CONSTRAINT FK1_Provider FOREIGN KEY (ProviderStatusCode) REFERENCES
HCI_ProviderStatus(ProviderStatusCode)
   )
   -- End Creating the ProviderData Table
```

```
   -- Creating the Claim table
   CREATE TABLE HCI_Claim (
   -- Columns for the Claim table
   ClaimID int identity,
   MemberID int not null,
   ProviderID int not null,
   ClaimsServiceDate datetime not null,
   ClaimsSettleDate datetime not null,
      -- Constraints on the Claim Table
   CONSTRAINT PK_Claim PRIMARY KEY (ClaimID),
   CONSTRAINT FK1_Caim FOREIGN KEY (MemberID) REFERENCES HCI_Member(MemberID),
   CONSTRAINT FK2_Caim FOREIGN KEY (ProviderID) REFERENCES HCI_Provider(ProviderID)
   )
   -- End Creating the Claim Table

   -- Creating the ClaimDetail table
    CREATE TABLE HCI_ClaimDetail (
   -- Columns for the ClaimDetail table
   ClaimID int,
   ClaimLine int,
   ClaimStatusCode char(2) not null,
   ClaimDiagnosisCode varchar(10) not null,
   ClaimLineAmountBilled decimal(13,2),
   ClaimLineAmountApproved decimal(13,2),
      -- Constraints on the ClaimDetail Table
   CONSTRAINT PK_ClaimDetail PRIMARY KEY (ClaimID,ClaimLine),
   CONSTRAINT FK1_ClaimDetail FOREIGN KEY (ClaimStatusCode) REFERENCES
HCI_ClaimStatus(ClaimStatusCode)
   )
   -- End Creating the ClaimDetail Table

   -- Creating the FinalDisposition table
    CREATE TABLE HCI_FinalDisposition (
   -- Columns for the FinalDisposition table
   CheckID int identity,
   ProviderID int not null,
   CheckIssueDate datetime not null,
   CheckAmount decimal(13,2),
      -- Constraints on the FinalDisposition Table
   CONSTRAINT PK_FinalDisposition PRIMARY KEY (CheckID),
   CONSTRAINT FK1_FinalDisposition FOREIGN KEY (ProviderID) REFERENCES
HCI_Provider(ProviderID)
   )
   -- End Creating the FinalDisposition Table

   -- Creating the ClaimDisposition table
    CREATE TABLE HCI_ClaimDisposition (
   -- Columns for the ClaimDisposition table
   ClaimDispositionID int identity,
   ClaimID int not null,
   CheckID int not null,
      -- Constraints on the ClaimDisposition Table
   CONSTRAINT PK_ClaimDisposition PRIMARY KEY (ClaimDispositionID),
   CONSTRAINT FK1_ClaimDisposition FOREIGN KEY (ClaimID) REFERENCES HCI_Claim(ClaimID),
   CONSTRAINT FK2_ClaimDisposition FOREIGN KEY (CheckID) REFERENCES
HCI_FinalDisposition(CheckID)
   )
```

```
-- End Creating the ClaimDisposition Table

SELECT * from INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME LIKE 'HCI_%'
```

100 %

Results | Messages

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	IST659_M403_pkulka04	dbo	HCI_Claim	BASE TABLE
2	IST659_M403_pkulka04	dbo	HCI_ClaimDetail	BASE TABLE
3	IST659_M403_pkulka04	dbo	HCI_ClaimDisposition	BASE TABLE
4	IST659_M403_pkulka04	dbo	HCI_ClaimStatus	BASE TABLE
5	IST659_M403_pkulka04	dbo	HCI_FinalDisposition	BASE TABLE
6	IST659_M403_pkulka04	dbo	HCI_Member	BASE TABLE
7	IST659_M403_pkulka04	dbo	HCI_MemberStatus	BASE TABLE
8	IST659_M403_pkulka04	dbo	HCI_Provider	BASE TABLE
9	IST659_M403_pkulka04	dbo	HCI_ProviderStatus	BASE TABLE

5. Data Creation

Data was loaded into tables by using import file function and then executing transform and load steps. Data was loaded in same sequence as creation to account for foreign key dependencies between tables.

```
-- INSERT into ProviderStatus table
INSERT INTO HCI_ProviderStatus (ProviderStatusCode, ProviderStatus)
SELECT ProviderStatusCode, ProviderStatus FROM ProviderStatus$

-- -- INSERT into MemberStatus table
INSERT INTO HCI_MemberStatus (MemberStatusCode, MemberStatus)
SELECT MemberStatusCode, MemberStatus FROM MemberStatus$

-- INSERT into ClaimStatus table
INSERT INTO HCI_ClaimStatus (ClaimStatusCode, ClaimStatus)
SELECT ClaimStatusCode, ClaimStatus FROM ClaimStatus$


-- INSERT into Member table
SET IDENTITY_INSERT dbo.HCI_Member ON
INSERT INTO HCI_Member (MemberID,
                                        MemberFirstName,
                                        MemberMiddleInitial,
                                        MemberLastName,
                                        MemberStreetName,
                                        MemberCity,
                                        StateCode,
                                        MemberZipCode,
                                        MemberStatusCode,
                                        MemberCoverageStartDate,
                                        MemberCoverageEndDate,
```

```
                                        PrimarySubscriberID)
                    SELECT              MemberID,
                                        MemberFirstName,
                                        MemberMiddleInitial,
                                        MemberLastName,
                                        MemberStreetAddress,
                                        MemberCity,
                                        StateCode,
                                        MemberZipCode,
                                        MemberStatusCode,
                                        MemberCoverageStartDate,
                                        MemberCoverageEndDate,
                                        PrimarySubscriberID
                    FROM Member$
SET IDENTITY_INSERT dbo.HCI_Member OFF

-- INSERT into Member table
SET IDENTITY_INSERT dbo.HCI_Member ON
INSERT INTO HCI_Member (MemberID,
                                        MemberFirstName,
                                        MemberMiddleInitial,
                                        MemberLastName,
                                        MemberStreetName,
                                        MemberCity,
                                        StateCode,
                                        MemberZipCode,
                                        MemberStatusCode,
                                        MemberCoverageStartDate,
                                        MemberCoverageEndDate,
                                        PrimarySubscriberID)
                    SELECT              MemberID,
                                        MemberFirstName,
                                        MemberMiddleInitial,
                                        MemberLastName,
                                        MemberStreetAddress,
                                        MemberCity,
                                        StateCode,
                                        MemberZipCode,
                                        MemberStatusCode,
                                        MemberCoverageStartDate,
                                        MemberCoverageEndDate,
                                        PrimarySubscriberID
                    FROM Dependent$
SET IDENTITY_INSERT dbo.HCI_Member OFF

-- -- INSERT into Provider table
SET IDENTITY_INSERT dbo.HCI_Provider ON
INSERT INTO HCI_Provider (ProviderID,
                                        ProviderFirstName,
                                        ProviderMiddleInitial,
                                        ProviderLastName,
                                        ProviderStreetName,
                                        ProviderCity,
                                        StateCode,
                                        ProviderZipCode,
                                        ProviderStatusCode,
                                        ProviderParticipationStartDate,
                                        ProviderParticipationEndDate)
```

```
                    SELECT              ProviderID,
                                        ProviderFirstName,
                                        ProviderMiddleInitial,
                                        ProviderLastName,
                                        ProviderStreetAddress,
                                        ProviderCity,
                                        StateCode,
                                        ProviderZipCode,
                                        ProviderStatusCode,
                                        ProviderParticipationStartDate,
                                        ProviderParticipationEndDate
                    FROM Provider$
SET IDENTITY_INSERT dbo.HCI_Provider OFF

-- INSERT into Claim table
SET IDENTITY_INSERT dbo.HCI_Claim ON
INSERT INTO HCI_Claim (ClaimID, MemberID, ProviderID, ClaimsServiceDate, ClaimsSettleDate)
SELECT ClaimID, MemberID, ProviderID, ClaimServiceDate, ClaimSettleDate FROM Claim$
SET IDENTITY_INSERT dbo.HCI_Claim OFF

-- INSERT into ClaimDetail table
--SET IDENTITY_INSERT dbo.HCI_ClaimDetail ON
INSERT INTO HCI_ClaimDetail (ClaimID, ClaimLine, ClaimStatusCode, ClaimDiagnosisCode,
ClaimLineAmountBilled, ClaimLineAmountApproved)
SELECT ClaimID, Line, ClaimStatusCode, ClaimDiagnosisCode, ClaimLineAmountBilled,
ClaimLineAmountApproved FROM ClaimDetail$
--SET IDENTITY_INSERT dbo.HCI_ClaimDetail OFF

-- INSERT into FinalDisposition table
SET IDENTITY_INSERT dbo.HCI_FinalDisposition ON
INSERT INTO HCI_FinalDisposition (CheckID, ProviderID, CheckIssueDate, CheckAmount)
SELECT CheckID, ProviderID, CheckIssueDate, CheckAmount FROM FinalDisposition$
SET IDENTITY_INSERT dbo.HCI_FinalDisposition OFF

-- INSERT into ClaimDisposition table
SET IDENTITY_INSERT dbo.HCI_ClaimDisposition ON
INSERT INTO HCI_ClaimDisposition (ClaimDispositionID, ClaimID, CheckID)
SELECT ClaimsDispositionID, ClaimID, CheckID FROM ClaimsDisposition$
SET IDENTITY_INSERT dbo.HCI_ClaimDisposition OFF
```

```
Select * from ClaimStatus
```

Results Messages

	ClaimStatusCode	ClaimStatus
1	AP	Approved
2	DN	Denied
3	DP	Duplicate

```
Select * from HCI_ProviderStatus
```

100 %

Results Messages

	ProviderStatusCode	ProviderStatus
1	IN	In Network
2	ON	Out of Network

```
Select * from HCI_MemberStatus
```

100 %

Results Messages

	MemberStatusCode	MemberStatus
1	AC	Active
2	TR	Terminated

```
Select * from HCI_Member
```

100 %

Results Messages

	MemberID	MemberFirstName	MemberMiddleInitial	MemberLastName	MemberStreetName	MemberCity	StateCode	MemberZipCode	MemberStatusCode	MemberCoverageStartDate	MemberCoverageEndDate	PrimarySubscriberID
1	11001	KRIS	NULL	MARRIER	228 RUNAMUCK PL #2808	BALTIMORE	MD	21217	TR	2018-01-01 00:00:00.000	2018-07-01 00:00:00.000	NULL
2	11002	MINNA	NULL	AMIGON	2371 JERROLD AVE	PHILADELPHIA	PA	19132	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
3	11003	ABEL	NULL	MACLEAD	37275 ST RT 17M M	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
4	11004	CAMMY	NULL	ALBARES	56 E MOREHEAD ST	DALLAS	TX	75207	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
5	11005	BETTE	NULL	NICKA	6 S 33RD ST	PHILADELPHIA	PA	19132	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
6	11006	WILLARD	NULL	KOLMETZ	618 W YAKIMA AVE	DALLAS	TX	75207	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
7	11007	MARYANN	NULL	ROYSTER	74 S WESTGATE ST	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
8	11008	EZEKIEL	NULL	CHUI	2 CEDAR AVE #84	BALTIMORE	MD	21217	TR	2018-01-01 00:00:00.000	2018-11-01 00:00:00.000	NULL
9	11009	WILLOW	NULL	KUSKO	90991 THORBURN AVE	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
10	11010	BERNARDO	NULL	FIGEROA	386 9TH AVE N	DALLAS	TX	75207	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
11	11011	ALISHIA	NULL	SERGI	2742 DISTRIBUTION WAY	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
12	11012	JOSE	NULL	STOCKHAM	128 BRANSTEN RD	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
13	11013	VALENTINE	NULL	GILLIAN	775 W 17TH ST	DALLAS	TX	75207	TR	2018-01-01 00:00:00.000	2018-09-01 00:00:00.000	NULL
14	11014	BLAIR	NULL	MALET	209 DECKER DR	PHILADELPHIA	PA	19132	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
15	11015	BROCK	NULL	BOLOGNIA	4486 W O ST #1	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
16	11016	MARJORY	NULL	MASTELLA	71 SAN MATEO AVE	PHILADELPHIA	PA	19132	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
17	11017	TONETTE	NULL	WENNER	4545 COURTHOUSE RD	NEW YORK	NY	10011	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL
18	11018	AMBER	NULL	MONARREZ	14288 FOSTER AVE #4121	PHILADELPHIA	PA	19132	TR	2018-01-01 00:00:00.000	2018-12-01 00:00:00.000	NULL
19	11019	DEEANNA	NULL	JUHAS	14302 PENNSYLVANIA AVE	PHILADELPHIA	PA	19132	AC	2018-01-01 00:00:00.000	2020-01-01 00:00:00.000	NULL

SUID: 928949766
Prasad Kulkarni Week 4 Project Deliverable 2 12/14/2019

```
Select * from HCI_Provider
```

100 %

Results Messages

	ProviderID	ProviderFirstName	ProviderMiddleInitial	ProviderLastName	ProviderStreetName	ProviderCity	StateCode	ProviderZipCode	ProviderStatusCode	ProviderParticipationStartDate	ProviderParticipationEndDate
2	51651	JENNIFER	NULL	SHEEHAN	5801 BREMO RD	PHILADELPHIA	PA	19132	IN	2005-01-01 00:00:00.000	NULL
3	52015	WILLIAM	NULL	MIZIKAR	320 E NORTH AVE	BALTIMORE	MD	21217	ON	2007-01-01 00:00:00.000	NULL
4	52737	LAURIE	NULL	SUTHERLAND	22 BRAMHALL ST	PHILADELPHIA	PA	19132	IN	2007-01-01 00:00:00.000	NULL
5	53152	EUGENE	NULL	GARVIN	622 W 168TH ST	NEW YORK	NY	10011	IN	2005-01-01 00:00:00.000	NULL
6	53185	PANKAJ	NULL	SHAH	11100 EUCLID AVE	DALLAS	TX	75207	IN	2007-01-01 00:00:00.000	NULL
7	53813	SHELLEY	NULL	CHILDRESS	462 GRIDER ST	BALTIMORE	MD	21217	IN	2007-01-01 00:00:00.000	NULL
8	53825	ALEXANDER	NULL	RIPS	2269 OCEAN AVE	BALTIMORE	MD	21217	IN	2005-01-01 00:00:00.000	NULL
9	53830	HANNAH	NULL	CLARK	243 CHARLES ST	DALLAS	TX	75207	IN	2007-01-01 00:00:00.000	NULL
10	54594	EDWARD	NULL	REQUENEZ	622 W 168TH ST	NEW YORK	NY	10011	IN	2005-01-01 00:00:00.000	NULL
11	54605	LAURENCE	NULL	MARK	622 W 168TH ST	NEW YORK	NY	10011	IN	2007-01-01 00:00:00.000	NULL
12	54613	STEPHAN	NULL	HATCH	365 MONTAUK AVE	PHILADELPHIA	PA	19132	ON	2007-01-01 00:00:00.000	NULL
13	54617	DEREK	NULL	ROSNER	75 FRANCIS STREET CWN 41	DALLAS	TX	75207	IN	2005-01-01 00:00:00.000	NULL
14	54791	DEBBIE	NULL	PIGOZZI	815 FREEPORT RD	NEW YORK	NY	10011	IN	2007-01-01 00:00:00.000	NULL
15	56082	ELIZABETH	NULL	MEDINA	600 N WOLFE ST	BALTIMORE	MD	21217	IN	2005-01-01 00:00:00.000	NULL
16	56192	MARIA	NULL	BORTKIEWICZ	243 CHARLES ST	DALLAS	TX	75207	IN	2007-01-01 00:00:00.000	NULL
17	56372	WILLIAM	NULL	MARSH	111 CONTINENTAL DR	NEW YORK	NY	10011	ON	2007-01-01 00:00:00.000	NULL
18	56567	JEEMISA	NULL	SNYDER	900 WALNUT ST	PHILADELPHIA	PA	19132	IN	2007-01-01 00:00:00.000	NULL
19	57055	MICHAEL	NULL	MONETA	11782 SW BARNES RD	DALLAS	TX	75207	IN	2005-01-01 00:00:00.000	NULL
20	57726	CHRISTOPHER	NULL	ROMANOWSKI	9003 GARDENIA RD	BALTIMORE	MD	21217	ON	2007-01-01 00:00:00.000	NULL

```
Select * from HCI_CLaim
```

100 %

Results Messages

	ClaimID	MemberID	ProviderID	ClaimsServiceDate	ClaimsSettleDate
1	110250	11093	54594	2008-12-15 00:00:00.000	2009-01-29 00:00:00.000
2	110718	11017	54594	2008-12-19 00:00:00.000	2009-02-02 00:00:00.000
3	112571	11084	52737	2018-01-01 00:00:00.000	2018-02-15 00:00:00.000
4	112572	11111	54791	2018-01-01 00:00:00.000	2018-02-15 00:00:00.000
5	113482	11099	54791	2018-01-01 00:00:00.000	2018-02-15 00:00:00.000
6	114402	11101	53185	2018-02-01 00:00:00.000	2018-03-18 00:00:00.000
7	115304	11025	57726	2018-02-01 00:00:00.000	2018-03-18 00:00:00.000
8	116222	11061	51227	2018-03-01 00:00:00.000	2018-04-15 00:00:00.000
9	118104	11043	54617	2018-01-04 00:00:00.000	2018-02-18 00:00:00.000
10	118105	11096	53813	2018-04-01 00:00:00.000	2018-05-16 00:00:00.000
11	119960	11074	53152	2018-05-01 00:00:00.000	2018-06-15 00:00:00.000
12	119961	11088	56372	2018-05-01 00:00:00.000	2018-06-15 00:00:00.000
13	121801	11004	54617	2018-06-01 00:00:00.000	2018-07-16 00:00:00.000
14	122739	11019	52737	2018-01-06 00:00:00.000	2018-02-20 00:00:00.000
15	123642	11048	52737	2018-07-01 00:00:00.000	2018-08-15 00:00:00.000
16	127177	11066	54617	2018-09-01 00:00:00.000	2018-10-16 00:00:00.000
17	127178	11088	56372	2018-09-01 00:00:00.000	2018-10-16 00:00:00.000
18	128100	11060	51651	2018-09-01 00:00:00.000	2018-10-16 00:00:00.000
19	130790	11041	56192	2018-01-11 00:00:00.000	2018-02-25 00:00:00.000

```
Select * from HCI_ClaimDetail
```

100 %

Results Messages

	ClaimID	ClaimLine	ClaimStatusCode	ClaimDiagnosisCode	ClaimLineAmountBilled	ClaimLineAmountApproved
1	110250	1	AP	V4512	2700.00	1800.00
2	110718	1	AP	V861	150.00	100.00
3	112571	1	AP	78702	300.00	200.00
4	112572	1	AP	4019	75.00	50.00
5	113482	1	AP	78652	150.00	100.00
6	114402	1	AP	71596	60.00	40.00
7	115304	1	AP	V069	60.00	40.00
8	116222	1	AP	2722	30.00	20.00
9	118104	1	AP	V8909	75.00	50.00
10	118105	1	AP	4417	90.00	60.00
11	119960	1	AP	78903	1200.00	800.00
12	119961	1	AP	25000	15.00	10.00
13	121801	1	AP	71988	60.00	40.00
14	121801	2	AP	V069	50.00	40.00
15	122739	1	AP	2723	45.00	30.00
16	123642	1	AP	78602	120.00	80.00
17	123642	2	AP	25000	250.00	200.00
18	127177	1	DN	78966	75.00	50.00
19	127178	1	DN	25000	30.00	20.00

```
Select * from HCI_FinalDisposition
```

Results Messages

	CheckID	ProviderID	CheckIssueDate	CheckAmount
1	1001	54594	2009-01-29 00:00:00.000	1800.00
2	1002	54594	2009-02-02 00:00:00.000	100.00
3	1003	52737	2018-02-15 00:00:00.000	200.00
4	1004	54791	2018-02-15 00:00:00.000	50.00
5	1005	54791	2018-02-15 00:00:00.000	100.00
6	1006	53185	2018-03-18 00:00:00.000	40.00
7	1007	57726	2018-03-18 00:00:00.000	40.00
8	1008	51227	2018-04-15 00:00:00.000	20.00
9	1009	54617	2018-02-18 00:00:00.000	50.00
10	1010	53813	2018-05-16 00:00:00.000	60.00
11	1011	53152	2018-06-15 00:00:00.000	800.00
12	1012	56372	2018-06-15 00:00:00.000	10.00
13	1013	54617	2018-07-16 00:00:00.000	80.00
14	1014	52737	2018-02-20 00:00:00.000	30.00
15	1015	52737	2018-08-15 00:00:00.000	280.00
16	1016	54617	2018-10-16 00:00:00.000	50.00
17	1017	56372	2018-10-16 00:00:00.000	20.00
18	1018	51651	2018-10-16 00:00:00.000	40.00
19	1019	56192	2018-02-25 00:00:00.000	20.00

```
Select * from HCI_ClaimDisposition
```

Results | Messages

	ClaimDispositionID	ClaimID	CheckID
1	6001	110250	1001
2	6002	110718	1002
3	6003	112571	1003
4	6004	112572	1004
5	6005	113482	1005
6	6006	114402	1006
7	6007	115304	1007
8	6008	116222	1008
9	6009	118104	1009
10	6010	118105	1010
11	6011	119960	1011
12	6012	119961	1012
13	6013	121801	1013
14	6014	122739	1014
15	6015	123642	1015
16	6016	127177	1016
17	6017	127178	1017
18	6018	128100	1018
19	6019	130790	1019

6. Data Manipulation

FUNCTIONS:

Business Need: Create a function to return number dependents for a given Member

```
CREATE FUNCTION dbo.NumberOfDependents(@MemberID int)
       RETURNS int AS
       BEGIN
              DECLARE @Deps int
              SELECT @Deps = COUNT(a.MemberID) FROM HCI_Member a
              WHERE a.PrimarySubscriberID =@MemberID
              RETURN @Deps
       END;

SELECT b.MemberID, dbo.NumberOfDependents(b.MemberID) as NumberOfDependents
FROM HCI_MEMBER b
WHERE b.PrimarySubscriberID is null
GROUP BY b.MemberID
ORDER BY NumberOfDependents DESC
```

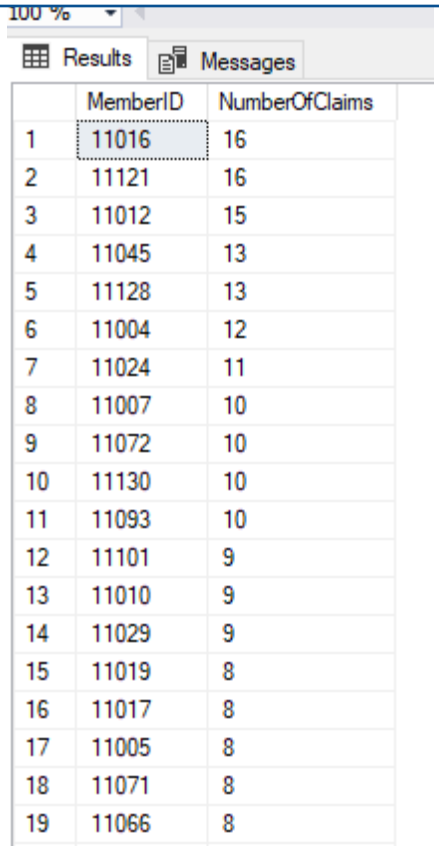
Results | Messages

	MemberID	NumberOfDependents
1	11064	4
2	11083	3
3	11087	3
4	11086	1
5	11061	1
6	11062	1
7	11063	1
8	11068	1
9	11069	1
10	11070	1
11	11071	1
12	11072	1
13	11073	1
14	11074	1
15	11075	1
16	11076	1
17	11077	1
18	11078	1
19	11079	1

Business Need: create a function to return number of claims for a given Member

```
CREATE FUNCTION dbo.NumberOfClaims(@MemberID int)
       RETURNS int AS
       BEGIN
              DECLARE @claim int
              SELECT @claim = COUNT(a.ClaimID) FROM HCI_Claim a
              WHERE a.MemberID =@MemberID
              RETURN @claim
       END;

SELECT b.MemberID, dbo.NumberOfClaims(b.MemberID) as NumberOfClaims
FROM HCI_MEMBER b
GROUP BY b.MemberID
ORDER BY NumberOfClaims DESC
```

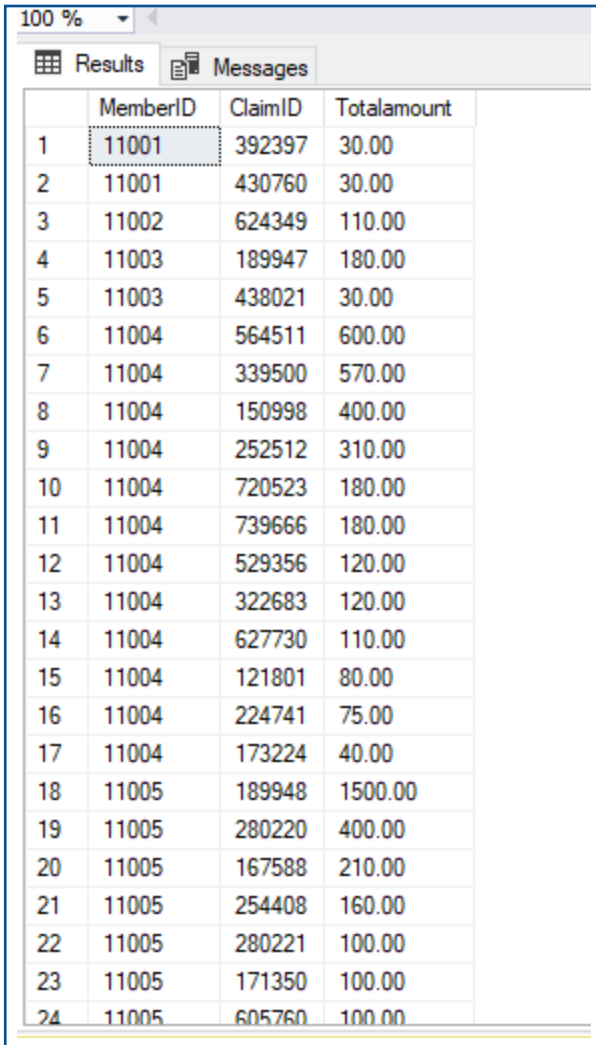


Results | Messages

	MemberID	NumberOfClaims
1	11016	16
2	11121	16
3	11012	15
4	11045	13
5	11128	13
6	11004	12
7	11024	11
8	11007	10
9	11072	10
10	11130	10
11	11093	10
12	11101	9
13	11010	9
14	11029	9
15	11019	8
16	11017	8
17	11005	8
18	11071	8
19	11066	8

Business Need: Create a function to return sum of approved claim line amounts for a given claim number

```
CREATE FUNCTION dbo.SumOfClaimLineAmounts(@ClaimID int)
       RETURNS Decimal (13,2) AS
       BEGIN
              DECLARE @amount int
              SELECT @amount = SUM(a.ClaimLineAmountApproved) FROM HCI_ClaimDetail a
              WHERE a.ClaimID =@ClaimID
              RETURN @amount
       END;
SELECT a.MemberID, b.ClaimID, dbo.SumOfClaimLineAmounts(b.ClaimID) as Totalamount
FROM HCI_MEMBER a
JOIN HCI_Claim b ON a.MemberID = b.MemberID
GROUP BY a.MemberID, b.ClaimID
ORDER BY MemberID, TotalAmount DESC
```



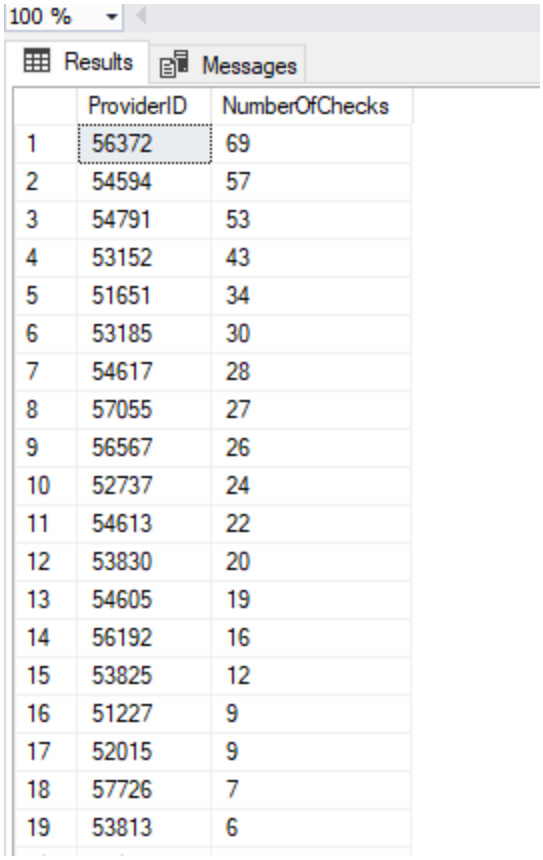
100 %

Results | Messages

	MemberID	ClaimID	Totalamount
1	11001	392397	30.00
2	11001	430760	30.00
3	11002	624349	110.00
4	11003	189947	180.00
5	11003	438021	30.00
6	11004	564511	600.00
7	11004	339500	570.00
8	11004	150998	400.00
9	11004	252512	310.00
10	11004	720523	180.00
11	11004	739666	180.00
12	11004	529356	120.00
13	11004	322683	120.00
14	11004	627730	110.00
15	11004	121801	80.00
16	11004	224741	75.00
17	11004	173224	40.00
18	11005	189948	1500.00
19	11005	280220	400.00
20	11005	167588	210.00
21	11005	254408	160.00
22	11005	280221	100.00
23	11005	171350	100.00
24	11005	605760	100.00

Business Need Create a function to return number of checks issued to Provider

```
CREATE FUNCTION dbo.NumberOfChecks(@ProviderID int)
       RETURNS int AS
       BEGIN
              DECLARE @check int
              SELECT @check = COUNT(a.CheckID) FROM HCI_FinalDisposition a
              WHERE a.ProviderID =@ProviderID
              RETURN @check
       END;
SELECT a.ProviderID, dbo.NumberOfChecks(a.ProviderID) as NumberOfChecks
FROM HCI_Provider a
GROUP BY a.ProviderID
ORDER BY NumberOfChecks DESC
```



100 %

Results Messages

	ProviderID	NumberOfChecks
1	56372	69
2	54594	57
3	54791	53
4	53152	43
5	51651	34
6	53185	30
7	54617	28
8	57055	27
9	56567	26
10	52737	24
11	54613	22
12	53830	20
13	54605	19
14	56192	16
15	53825	12
16	51227	9
17	52015	9
18	57726	7
19	53813	6

SUID: 928949766
Prasad Kulkarni Week 4 Project Deliverable 2 12/14/2019

Views:

Business Need: Create a View to list Member ID, Member Name, Member resident State and Total number of Claims submitted for each Member

```
CREATE VIEW ClaimsByMemberState AS
SELECT b.MemberID, CONCAT(b.MemberFirstName, ' ', b.MemberLastName) as MemberName,
b.StateCode, dbo.NumberOfClaims(b.MemberID) as NumberOfClaims
FROM HCI_MEMBER b
GROUP BY b.StateCode, b.MemberID, b.MemberFirstName, b.MemberLastName

SELECT * FROM dbo.ClaimsByMemberState ORDER BY StateCode, NumberOfClaims DESC
```

100 %

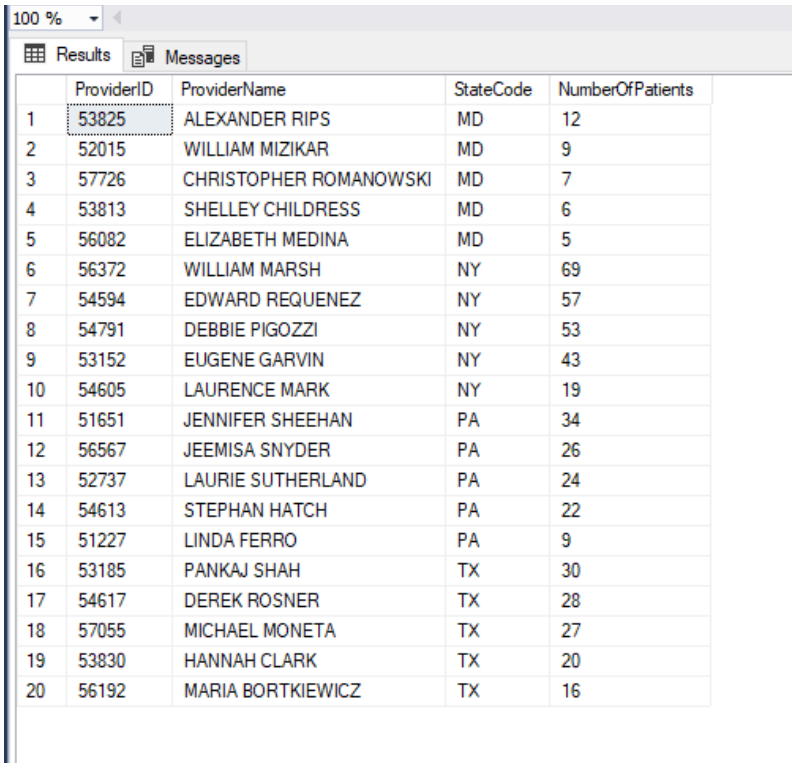
Results | Messages

	MemberID	MemberName	StateCode	NumberOfClaims
7	11112	ANNELLE TAGALA	MD	2
8	11036	GLORY KULZER	MD	2
9	11001	KRIS MARRIER	MD	2
10	11042	FERNANDA JILLSON	MD	2
11	11052	SYLVIA COUSEY	MD	1
12	11008	EZEKIEL CHUI	MD	1
13	11025	LAUREL REITLER	MD	1
14	11124	IZETTA DEWAR	MD	1
15	11076	DETRA COYIER	MD	1
16	11083	LORETA TIMENEZ	MD	1
17	11097	LASHAUNDA LIZAMA	MD	1
18	11121	ELLI MCLAIRD	NY	16
19	11012	JOSE STOCKHAM	NY	15
20	11045	OZELL SHEALY	NY	13
21	11128	SHAREN BOURBON	NY	13
22	11024	MOON PARLATO	NY	11
23	11007	MARYANN ROYSTER	NY	10
24	11130	GILMA LIUKKO	NY	10
25	11093	DERICK DHAMER	NY	10
26	11099	SHALON SHADRICK	NY	8
27	11017	TONETTE WENNER	NY	8
28	11067	ALAINE BERGESEN	NY	8
29	11021	TAWNA BUVENS	NY	7
30	11059	FAUSTO AGRAMONTE	NY	6
31	11053	NANA WRINKLES	NY	6
32	11094	KIRK HERRITT	NY	6
33	11122	LESLIE THREETS	NY	6
34	11114	JANINE RHODEN	NY	6
35	11111	MIREYA FRERKING	NY	5
36	11127	JESS CHAFFINS	NY	5
37	11088	BARBRA ADKIN	NY	5

Business Need: Create a view to list Provider ID Provider Name, Provider State and number of Patients served by Provider

```
CREATE VIEW PatientsServedByProvider AS
SELECT b.ProviderID, CONCAT(b.ProviderFirstName, ' ', b.ProviderLastName) as ProviderName,
b.StateCode, COUNT(C.MemberID) as NumberOfPatients
FROM HCI_Provider b JOIN HCI_Claim c ON b.ProviderID = c.ProviderID
GROUP BY b.StateCode, b.ProviderID, b.ProviderFirstName, b.ProviderLastName

SELECT * FROM dbo.PatientsServedByProvider ORDER BY StateCode, NumberOfPatients DESC
```



100 %

Results | Messages

	ProviderID	ProviderName	StateCode	NumberOfPatients
1	53825	ALEXANDER RIPS	MD	12
2	52015	WILLIAM MIZIKAR	MD	9
3	57726	CHRISTOPHER ROMANOWSKI	MD	7
4	53813	SHELLEY CHILDRESS	MD	6
5	56082	ELIZABETH MEDINA	MD	5
6	56372	WILLIAM MARSH	NY	69
7	54594	EDWARD REQUENEZ	NY	57
8	54791	DEBBIE PIGOZZI	NY	53
9	53152	EUGENE GARVIN	NY	43
10	54605	LAURENCE MARK	NY	19
11	51651	JENNIFER SHEEHAN	PA	34
12	56567	JEEMISA SNYDER	PA	26
13	52737	LAURIE SUTHERLAND	PA	24
14	54613	STEPHAN HATCH	PA	22
15	51227	LINDA FERRO	PA	9
16	53185	PANKAJ SHAH	TX	30
17	54617	DEREK ROSNER	TX	28
18	57055	MICHAEL MONETA	TX	27
19	53830	HANNAH CLARK	TX	20
20	56192	MARIA BORTKIEWICZ	TX	16

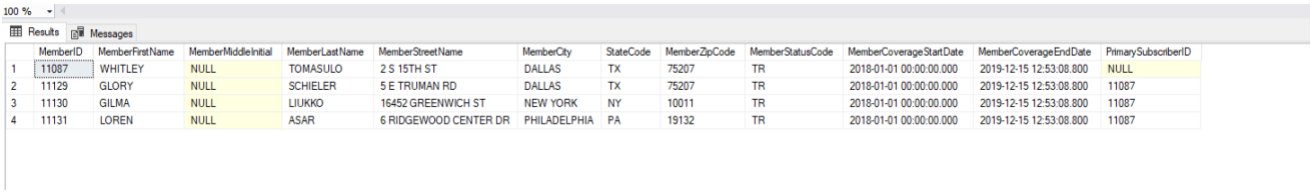
Stored Procedures:

Business Need: Create a procedure to a terminate a Member and all it's dependents

```
CREATE PROCEDURE HCI_TerminateMember (@MemberID int)
AS
BEGIN
-- Step1: Check if Member has any Dependents. Update Dependent Status First
       IF EXISTS (SELECT * FROM HCI_Member WHERE PrimarySubscriberID = @MemberID)
       BEGIN
              UPDATE HCI_Member
              SET MemberStatusCode = 'TR',
                  MemberCoverageEndDate = GETDATE ()
              WHERE PrimarySubscriberID = @MemberID
       END
-- Step2: Update Member Status next
       UPDATE HCI_Member
       SET MemberStatusCode = 'TR',
       MemberCoverageEndDate = GETDATE ()
       WHERE MemberID = @MemberID
END;

EXEC HCI_TerminateMember '11087'

SELECT * FROM HCI_Member where MemberID = '11087' or PrimarySubscriberID = '11087'
```



	MemberID	MemberFirstName	MemberMiddleInitial	MemberLastName	MemberStreetName	MemberCity	StateCode	MemberZipCode	MemberStatusCode	MemberCoverageStartDate	MemberCoverageEndDate	PrimarySubscriberID
1	11087	WHITLEY	NULL	TOMASULO	2 S 15TH ST	DALLAS	TX	75207	TR	2018-01-01 00:00:00.000	2019-12-15 12:53:08.800	NULL
2	11129	GLORY	NULL	SCHIELER	5 E TRUMAN RD	DALLAS	TX	75207	TR	2018-01-01 00:00:00.000	2019-12-15 12:53:08.800	11087
3	11130	GILMA	NULL	LIUKKO	16452 GREENWICH ST	NEW YORK	NY	10011	TR	2018-01-01 00:00:00.000	2019-12-15 12:53:08.800	11087
4	11131	LOREN	NULL	ASAR	6 RIDGEWOOD CENTER DR	PHILADELPHIA	PA	19132	TR	2018-01-01 00:00:00.000	2019-12-15 12:53:08.800	11087

Business Need: Through Appeals process a previously denied claim can now be approved. Create a procedure to update Claims Status code to Approved

```
CREATE PROCEDURE HCI_ApproveClaim (@ClaimID int)
AS
BEGIN
-- Step 1 Update Claim Detail to mark all lines for the claim as approved
       UPDATE HCI_ClaimDetail
       SET ClaimStatusCode = 'AP'
       WHERE ClaimID = @ClaimID
-- Step 2 Update Claim table to set overall claim settle date to today's date
       UPDATE HCI_CLAIM
       SET ClaimsSettleDate = GETDATE ()
       WHERE ClaimID = @ClaimID
END;
-- Data prior to executing stored procedure
SELECT * FROM HCI_ClaimDetail WHERE ClaimID = '144521'
```

	ClaimID	ClaimLine	ClaimStatusCode	ClaimDiagnosisCode	ClaimLineAmountBilled	ClaimLineAmountApproved
1	144521	1	DN	78969	75.00	50.00
2	144521	2	DN	1013	70.00	50.00

```
-- Execute Stored Procedure
EXEC HCI_ApproveClaim '144521'
-- Data after executing Stored Procedure
SELECT * FROM HCI_ClaimDetail WHERE ClaimID = '144521'
```

	ClaimID	ClaimLine	ClaimStatusCode	ClaimDiagnosisCode	ClaimLineAmountBilled	ClaimLineAmountApproved
1	144521	1	AP	78969	75.00	50.00
2	144521	2	AP	1013	70.00	50.00

```
SELECT * FROM HCI_Claim WHERE ClaimID = '144521'
```

	ClaimID	MemberID	ProviderID	ClaimsServiceDate	ClaimsSettleDate
1	144521	11011	54791	2018-01-18 00:00:00.000	2019-12-15 13:15:38.303

7. Answering Data Questions

Business Need: Determine sum of claim amounts for all family members against their Primary subscriber

```
SELECT coalesce(PrimarySubscriberID,MemberID) PrimarySubscriber, sum(ClaimLineAmountApproved)
as TotalAmount FROM
(
SELECT a.PrimarySubscriberID, a.MemberID, c.ClaimLineAmountApproved FROM
HCI_Member a
left join HCI_Claim b on a.MemberID= b.MemberID
inner join HCI_ClaimDetail c on b.ClaimID = c.ClaimID
) d
Group BY coalesce(PrimarySubscriberID,MemberID)
```

Results Messages

	PrimarySubscriber	TotalAmount
1	11001	100.00
2	11002	150.00
3	11003	210.00
4	11004	3835.00
5	11005	2690.00
6	11006	1700.00
7	11007	2150.00
8	11008	1890.00
9	11009	1470.00
10	11010	1720.00
11	11011	1080.00
12	11012	4920.00
13	11013	840.00
14	11014	900.00
15	11015	210.00
16	11016	3290.00
17	11017	1970.00
18	11018	5050.00
19	11019	780.00

Business Need: What are the claims and associated Members submitted by provider in state of Texas

```
SELECT a.ProviderID, a.StateCode as ProviderState, CONCAT(a.ProviderFirstName, ' ',
a.ProviderLastName) as ProviderName,
b.ClaimID, b.ClaimsServiceDate as ServiceDate, c.MemberID, CONCAT(c.MemberFirstName, ' ',
c.MemberLastName) as MemberName, c.StateCode as MemberState
FROM HCI_Provider a JOIN HCI_Claim b ON a.ProviderID = b.ProviderID
JOIN HCI_Member c ON b.MemberID = c.MemberID
WHERE a.StateCode = 'TX'
ORDER BY ProviderID DESC
```

100 %

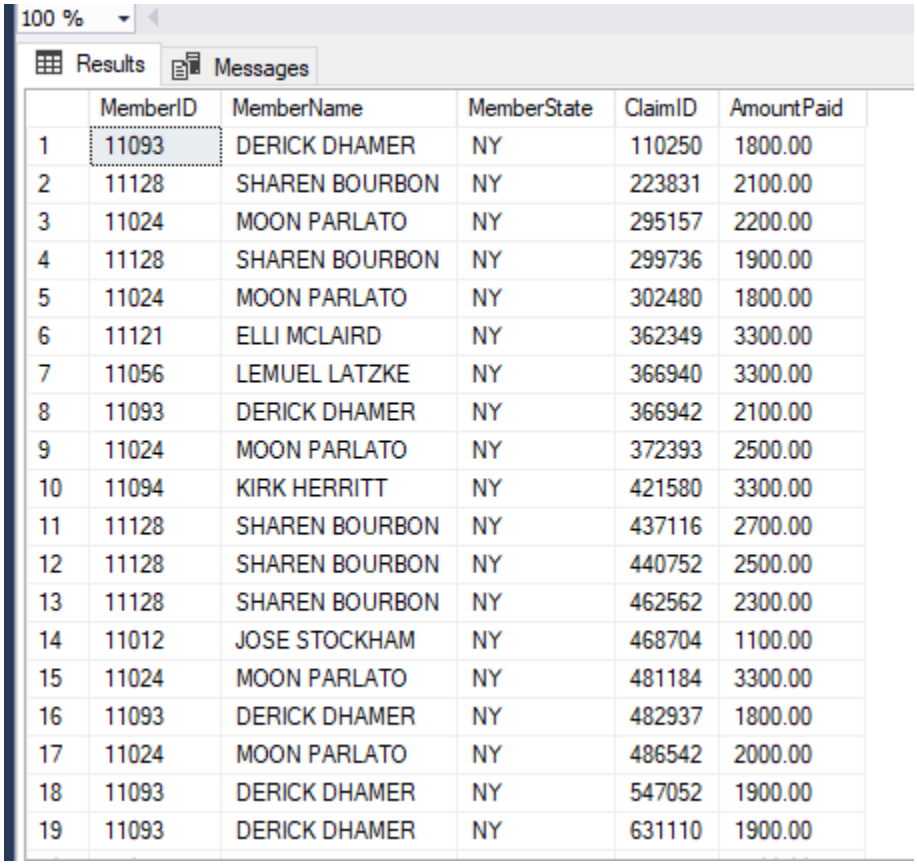
Results Messages

	ProviderID	ProviderState	ProviderName	ClaimID	ServiceDate	MemberID	MemberName	MemberState
1	57055	TX	MICHAEL MONETA	140012	2018-01-16 00:00:00.000	11010	BERNARDO FIGEROA	TX
2	57055	TX	MICHAEL MONETA	165765	2018-01-30 00:00:00.000	11010	BERNARDO FIGEROA	TX
3	57055	TX	MICHAEL MONETA	172249	2018-02-02 00:00:00.000	11072	MARVEL RAYMO	TX
4	57055	TX	MICHAEL MONETA	214638	2018-02-25 00:00:00.000	11072	MARVEL RAYMO	TX
5	57055	TX	MICHAEL MONETA	234924	2018-08-03 00:00:00.000	11034	SUE KOWNACKI	TX
6	57055	TX	MICHAEL MONETA	305163	2018-04-15 00:00:00.000	11072	MARVEL RAYMO	TX
7	57055	TX	MICHAEL MONETA	318057	2018-04-22 00:00:00.000	11034	SUE KOWNACKI	TX
8	57055	TX	MICHAEL MONETA	400396	2018-06-06 00:00:00.000	11072	MARVEL RAYMO	TX
9	57055	TX	MICHAEL MONETA	418893	2018-06-16 00:00:00.000	11072	MARVEL RAYMO	TX
10	57055	TX	MICHAEL MONETA	435308	2018-06-25 00:00:00.000	11034	SUE KOWNACKI	TX
11	57055	TX	MICHAEL MONETA	437115	2018-06-26 00:00:00.000	11072	MARVEL RAYMO	TX
12	57055	TX	MICHAEL MONETA	431741	2018-06-23 00:00:00.000	11057	BARRETT TOYAMA	TX
13	57055	TX	MICHAEL MONETA	475785	2018-07-18 00:00:00.000	11086	LILLI SCRIVEN	TX
14	57055	TX	MICHAEL MONETA	476658	2018-07-18 00:00:00.000	11034	SUE KOWNACKI	TX
15	57055	TX	MICHAEL MONETA	545289	2018-08-26 00:00:00.000	11010	BERNARDO FIGEROA	TX
16	57055	TX	MICHAEL MONETA	557489	2018-09-02 00:00:00.000	11010	BERNARDO FIGEROA	TX
17	57055	TX	MICHAEL MONETA	588787	2018-09-20 00:00:00.000	11010	BERNARDO FIGEROA	TX
18	57055	TX	MICHAEL MONETA	625219	2018-11-10 00:00:00.000	11072	MARVEL RAYMO	TX
19	57055	TX	MICHAEL MONETA	640310	2018-10-20 00:00:00.000	11034	SUE KOWNACKI	TX

Query executed successfully.

Business Need: What are Claims submitted for Members living in state of NY where final approved amount is greater than $1000.00

```
SELECT a.MemberID, CONCAT(a.MemberFirstName, ' ', a.MemberLastName) as MemberName,
a.StateCode as MemberState, b.ClaimID, c.ClaimLineAmountApproved as AmountPaid
FROM HCI_Member a JOIN HCI_Claim b ON a.MemberID = b.MemberID
JOIN HCI_ClaimDetail c ON b.ClaimID = c.ClaimID
WHERE a.StateCode = 'NY' and c.ClaimLineAmountApproved > 1000
```



100 %

Results | Messages

	MemberID	MemberName	MemberState	ClaimID	AmountPaid
1	11093	DERICK DHAMER	NY	110250	1800.00
2	11128	SHAREN BOURBON	NY	223831	2100.00
3	11024	MOON PARLATO	NY	295157	2200.00
4	11128	SHAREN BOURBON	NY	299736	1900.00
5	11024	MOON PARLATO	NY	302480	1800.00
6	11121	ELLI MCLAIRD	NY	362349	3300.00
7	11056	LEMUEL LATZKE	NY	366940	3300.00
8	11093	DERICK DHAMER	NY	366942	2100.00
9	11024	MOON PARLATO	NY	372393	2500.00
10	11094	KIRK HERRITT	NY	421580	3300.00
11	11128	SHAREN BOURBON	NY	437116	2700.00
12	11128	SHAREN BOURBON	NY	440752	2500.00
13	11128	SHAREN BOURBON	NY	462562	2300.00
14	11012	JOSE STOCKHAM	NY	468704	1100.00
15	11024	MOON PARLATO	NY	481184	3300.00
16	11093	DERICK DHAMER	NY	482937	1800.00
17	11024	MOON PARLATO	NY	486542	2000.00
18	11093	DERICK DHAMER	NY	547052	1900.00
19	11093	DERICK DHAMER	NY	631110	1900.00

Business Need: What are the number of checks issued and total amount paid for all Providers grouped by their participation status

```
SELECT a.ProviderID, CONCAT(a.ProviderFirstName, ' ', a.ProviderLastName) as ProviderName,
c.ProviderStatus, COUNT(b.CheckID) as NumberOfChecksIssued, SUM(b.CheckAmount) as
TotalAmountPaid
FROM HCI_Provider a JOIN HCI_FinalDisposition b ON a.ProviderID = b.ProviderID
JOIN HCI_ProviderStatus c ON a.ProviderStatusCode = c.ProviderStatusCode
GROUP BY a.ProviderID, a.ProviderFirstName, a.ProviderLastName,c.ProviderStatus
ORDER BY c.ProviderStatus, a.ProviderID DESC
```

100 %

Results Messages

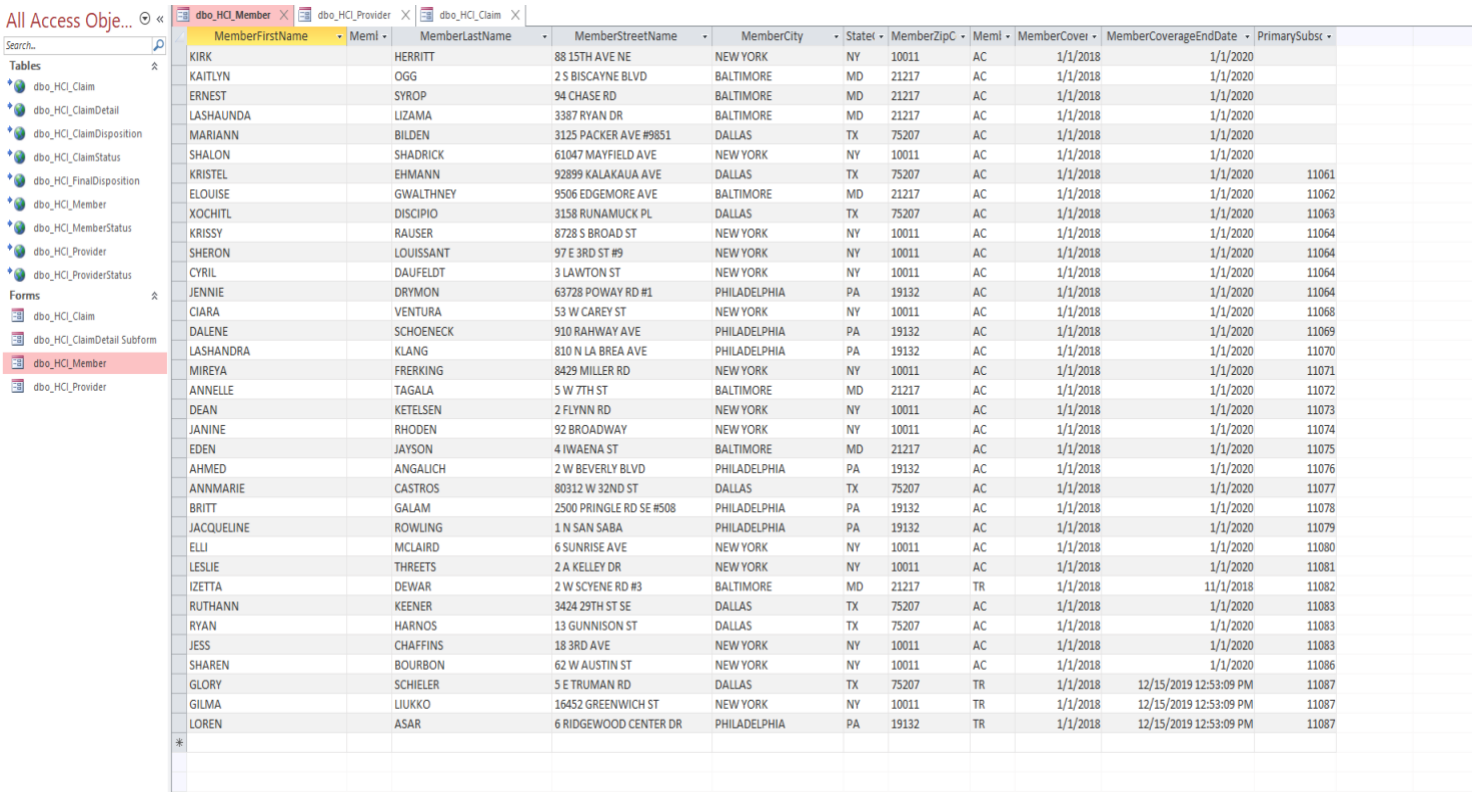
	ProviderID	ProviderName	ProviderStatus	NumberOfChecksIssued	TotalAmountPaid
1	57055	MICHAEL MONETA	In Network	27	7870.00
2	56567	JEEMISA SNYDER	In Network	26	3790.00
3	56192	MARIA BORTKIEWICZ	In Network	16	1625.00
4	56082	ELIZABETH MEDINA	In Network	5	1690.00
5	54791	DEBBIE PIGOZZI	In Network	53	22385.00
6	54617	DEREK ROSNER	In Network	28	5075.00
7	54605	LAURENCE MARK	In Network	19	6930.00
8	54594	EDWARD REQUENEZ	In Network	57	25300.00
9	53830	HANNAH CLARK	In Network	20	3330.00
10	53825	ALEXANDER RIPS	In Network	12	1860.00
11	53813	SHELLEY CHILDRESS	In Network	6	900.00
12	53185	PANKAJ SHAH	In Network	30	11665.00
13	53152	EUGENE GARVIN	In Network	43	16580.00
14	52737	LAURIE SUTHERLAND	In Network	24	5120.00
15	51651	JENNIFER SHEEHAN	In Network	34	8310.00
16	57726	CHRISTOPHER ROMANOWSKI	Out of Network	7	2150.00
17	56372	WILLIAM MARSH	Out of Network	69	28570.00
18	54613	STEPHAN HATCH	Out of Network	22	5040.00
19	52015	WILLIAM MIZIKAR	Out of Network	9	450.00
20	51227	LINDA FERRO	Out of Network	9	2680.00

8. Implementation

The user interface was programmed using MS Access, creating forms that allows users to insert the data into the SQL tables, updating the tables and then calling them back into the app.

- Add Member form
- Add Provider form
- Add Claim Form

Add Member

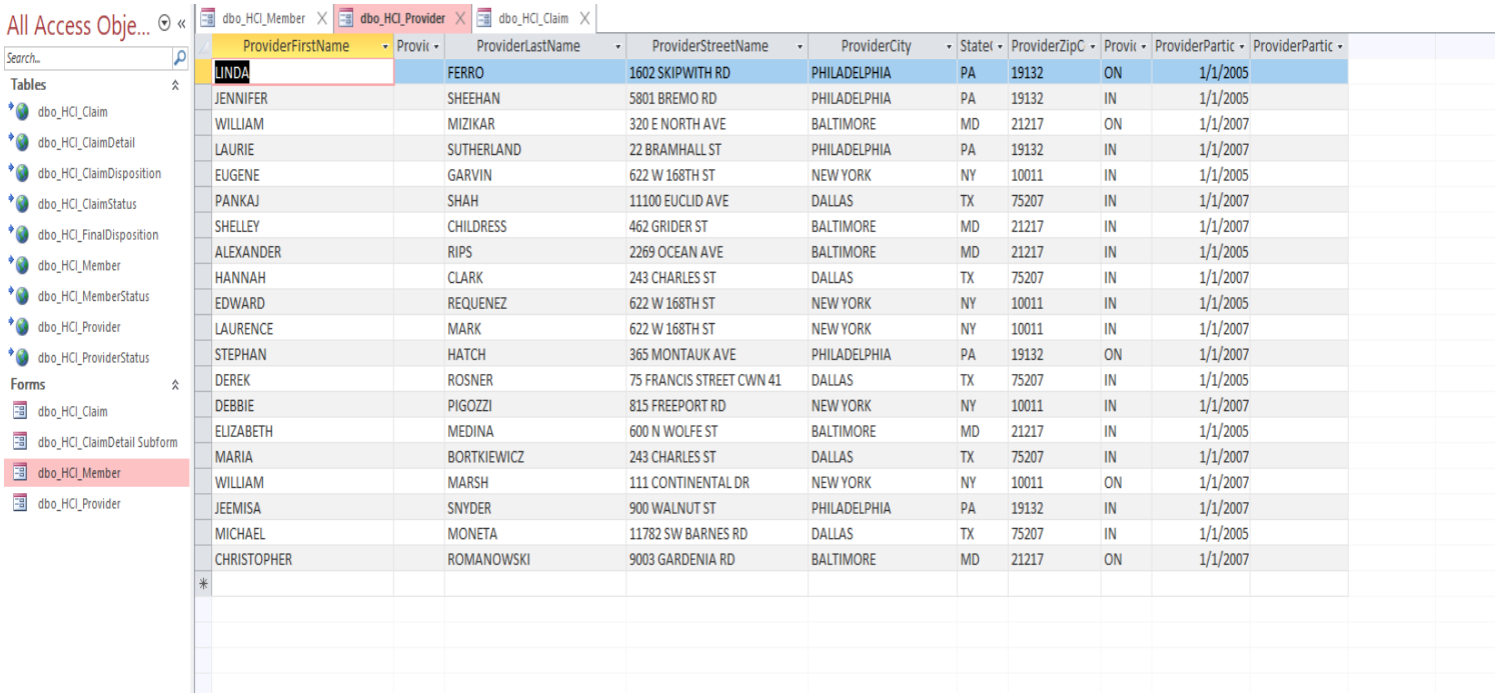


MemberFirstName	Meml	MemberLastName	MemberStreetName	MemberCity	State	MemberZipC	Meml	MemberCover	MemberCoverageEndDate	PrimarySubsc
KIRK		HERRITT	88 15TH AVE NE	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	
KAITLYN		OGG	2 S BISCAYNE BLVD	BALTIMORE	MD	21217	AC	1/1/2018	1/1/2020	
ERNEST		SYROP	94 CHASE RD	BALTIMORE	MD	21217	AC	1/1/2018	1/1/2020	
LASHAUNDA		LIZAMA	3387 RYAN DR	BALTIMORE	MD	21217	AC	1/1/2018	1/1/2020	
MARIANN		BILDEN	3125 PACKER AVE #9851	DALLAS	TX	75207	AC	1/1/2018	1/1/2020	
SHALON		SHADRICK	61047 MAYFIELD AVE	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	
KRISTEL		EHMANN	92899 KALAKAUA AVE	DALLAS	TX	75207	AC	1/1/2018	1/1/2020	11061
ELOUISE		GWALTHNEY	9506 EDGEMORE AVE	BALTIMORE	MD	21217	AC	1/1/2018	1/1/2020	11062
XOCHITL		DISCIPIO	3158 RUNAMUCK PL	DALLAS	TX	75207	AC	1/1/2018	1/1/2020	11063
KRISSY		RAUSER	8728 S BROAD ST	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11064
SHERON		LOUISSANT	97 E 3RD ST #9	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11064
CYRIL		DAUFELDT	3 LAWTON ST	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11064
JENNIE		DRYMON	63728 POWAY RD #1	PHILADELPHIA	PA	19132	AC	1/1/2018	1/1/2020	11064
CIARA		VENTURA	53 W CAREY ST	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11068
DALENE		SCHOENECK	910 RAHWAY AVE	PHILADELPHIA	PA	19132	AC	1/1/2018	1/1/2020	11069
LASHANDRA		KLANG	810 N LA BREA AVE	PHILADELPHIA	PA	19132	AC	1/1/2018	1/1/2020	11070
MIREYA		FRERKING	8429 MILLER RD	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11071
ANNELLE		TAGALA	5 W 7TH ST	BALTIMORE	MD	21217	AC	1/1/2018	1/1/2020	11072
DEAN		KETELSEN	2 FLYNN RD	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11073
JANINE		RHODEN	92 BROADWAY	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11074
EDEN		JAYSON	4 IWAENA ST	BALTIMORE	MD	21217	AC	1/1/2018	1/1/2020	11075
AHMED		ANGALICH	2 W BEVERLY BLVD	PHILADELPHIA	PA	19132	AC	1/1/2018	1/1/2020	11076
ANNMARIE		CASTROS	80312 W 32ND ST	DALLAS	TX	75207	AC	1/1/2018	1/1/2020	11077
BRITT		GALAM	2500 PRINGLE RD SE #508	PHILADELPHIA	PA	19132	AC	1/1/2018	1/1/2020	11078
JACQUELINE		ROWLING	1 N SAN SABA	PHILADELPHIA	PA	19132	AC	1/1/2018	1/1/2020	11079
ELLI		MCLAIRD	6 SUNRISE AVE	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11080
LESLIE		THREETS	2 A KELLEY DR	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11081
IZETTA		DEWAR	2 W SCYENE RD #3	BALTIMORE	MD	21217	TR	1/1/2018	11/1/2018	11082
RUTHANN		KEENER	3424 29TH ST SE	DALLAS	TX	75207	AC	1/1/2018	1/1/2020	11083
RYAN		HARNOS	13 GUNNISON ST	DALLAS	TX	75207	AC	1/1/2018	1/1/2020	11083
JESS		CHAFFINS	18 3RD AVE	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11083
SHAREN		BOURBON	62 W AUSTIN ST	NEW YORK	NY	10011	AC	1/1/2018	1/1/2020	11086
GLORY		SCHIELER	5 E TRUMAN RD	DALLAS	TX	75207	TR	1/1/2018	12/15/2019 12:53:09 PM	11087
GILMA		LIUKKO	16452 GREENWICH ST	NEW YORK	NY	10011	TR	1/1/2018	12/15/2019 12:53:09 PM	11087
LOREN		ASAR	6 RIDGEWOOD CENTER DR	PHILADELPHIA	PA	19132	TR	1/1/2018	12/15/2019 12:53:09 PM	11087

SUID: 928949766

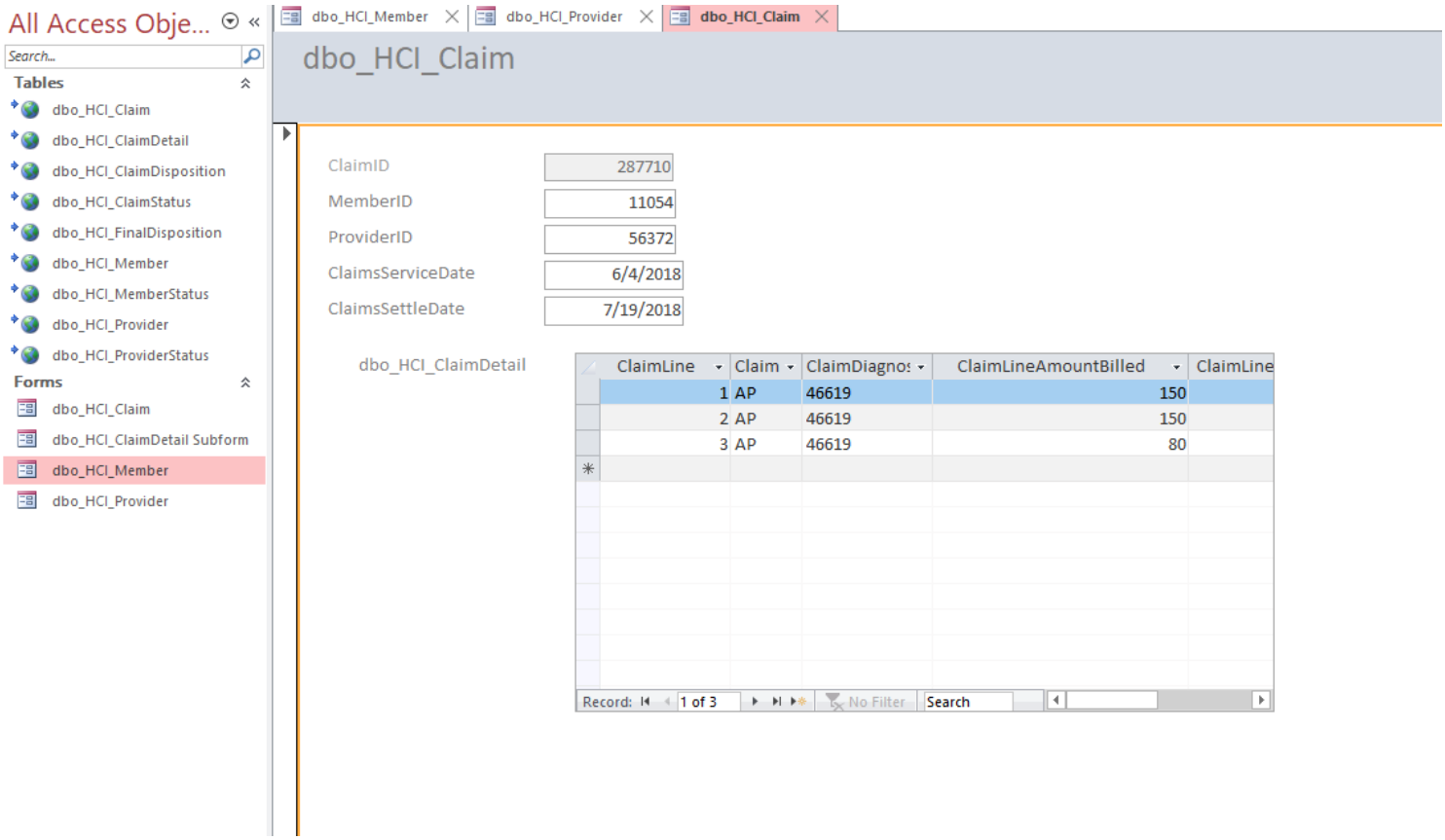
Prasad Kulkarni Week 4 Project Deliverable 2 12/14/2019

Add Provider



ProviderFirstName	Provi	ProviderLastName	ProviderStreetName	ProviderCity	State	ProviderZipC	Provi	ProviderPartic	ProviderPartic
LINDA		FERRO	1602 SKIPWITH RD	PHILADELPHIA	PA	19132	ON	1/1/2005	
JENNIFER		SHEEHAN	5801 BREMO RD	PHILADELPHIA	PA	19132	IN	1/1/2005	
WILLIAM		MIZIKAR	320 E NORTH AVE	BALTIMORE	MD	21217	ON	1/1/2007	
LAURIE		SUTHERLAND	22 BRAMHALL ST	PHILADELPHIA	PA	19132	IN	1/1/2007	
EUGENE		GARVIN	622 W 168TH ST	NEW YORK	NY	10011	IN	1/1/2005	
PANKAJ		SHAH	11100 EUCLID AVE	DALLAS	TX	75207	IN	1/1/2007	
SHELLEY		CHILDRESS	462 GRIDER ST	BALTIMORE	MD	21217	IN	1/1/2007	
ALEXANDER		RIPS	2269 OCEAN AVE	BALTIMORE	MD	21217	IN	1/1/2005	
HANNAH		CLARK	243 CHARLES ST	DALLAS	TX	75207	IN	1/1/2007	
EDWARD		REQUENEZ	622 W 168TH ST	NEW YORK	NY	10011	IN	1/1/2005	
LAURENCE		MARK	622 W 168TH ST	NEW YORK	NY	10011	IN	1/1/2007	
STEPHAN		HATCH	365 MONTAUK AVE	PHILADELPHIA	PA	19132	ON	1/1/2007	
DEREK		ROSNER	75 FRANCIS STREET CWN 41	DALLAS	TX	75207	IN	1/1/2005	
DEBBIE		PIGOZZI	815 FREEPORT RD	NEW YORK	NY	10011	IN	1/1/2007	
ELIZABETH		MEDINA	600 N WOLFE ST	BALTIMORE	MD	21217	IN	1/1/2005	
MARIA		BORTKIEWICZ	243 CHARLES ST	DALLAS	TX	75207	IN	1/1/2007	
WILLIAM		MARSH	111 CONTINENTAL DR	NEW YORK	NY	10011	ON	1/1/2007	
JEEMISA		SNYDER	900 WALNUT ST	PHILADELPHIA	PA	19132	IN	1/1/2007	
MICHAEL		MONETA	11782 SW BARNES RD	DALLAS	TX	75207	IN	1/1/2005	
CHRISTOPHER		ROMANOWSKI	9003 GARDENIA RD	BALTIMORE	MD	21217	ON	1/1/2007	

Add Claim



9. Reflection

After working extensively on this project, I realized that designing, creating and implementing a database requires much more strategic thinking. Consistent focus on simplifying user/application interaction, at the same time achieving necessary data integrity, data security, and concurrent handling through proper designing techniques (such as Normalization, Security measures etc.) are key factors in implementing operational database.

In first part, during initial stage I had thought about capturing patient and provider interactions through "Episodes" table but later realized that it will make current normalization structure between Member, Provider and Claims very complicated. Instead, this can be also be achieved by creating views for specific requirements post implementation.

If there are things that I would do differently, it would probably be on payments or final disposition tables. Merging claims and final disposition would provide consistent view of final payments across provider and claims data.

As an Information professional this exercise will definitely help me better anticipate some design and security challenges early in process so as to account for them in logical data modelling phase and help create better,

more robust, and easier to read database that - hopefully - will not need extensive structural changes post implementation.