



Learn TeraData 13.0 Lab #1.1

“Building Customer Correspondence Data Model”

This document briefs you the step by step approach to create customer correspondence data model for TD learner in Teradata 13.0 software. Join our professional training to learn from Experts.

History:

Version	Description Change	Author	Publish Date
0.1	Initial Draft	Lokesh kumar	10 th Jan 2011
0.1	1 st Review	Amit Sharma	10 th Jan 2011
0.1	2 nd Review	Manohar Krishna	10 th Jan 2011

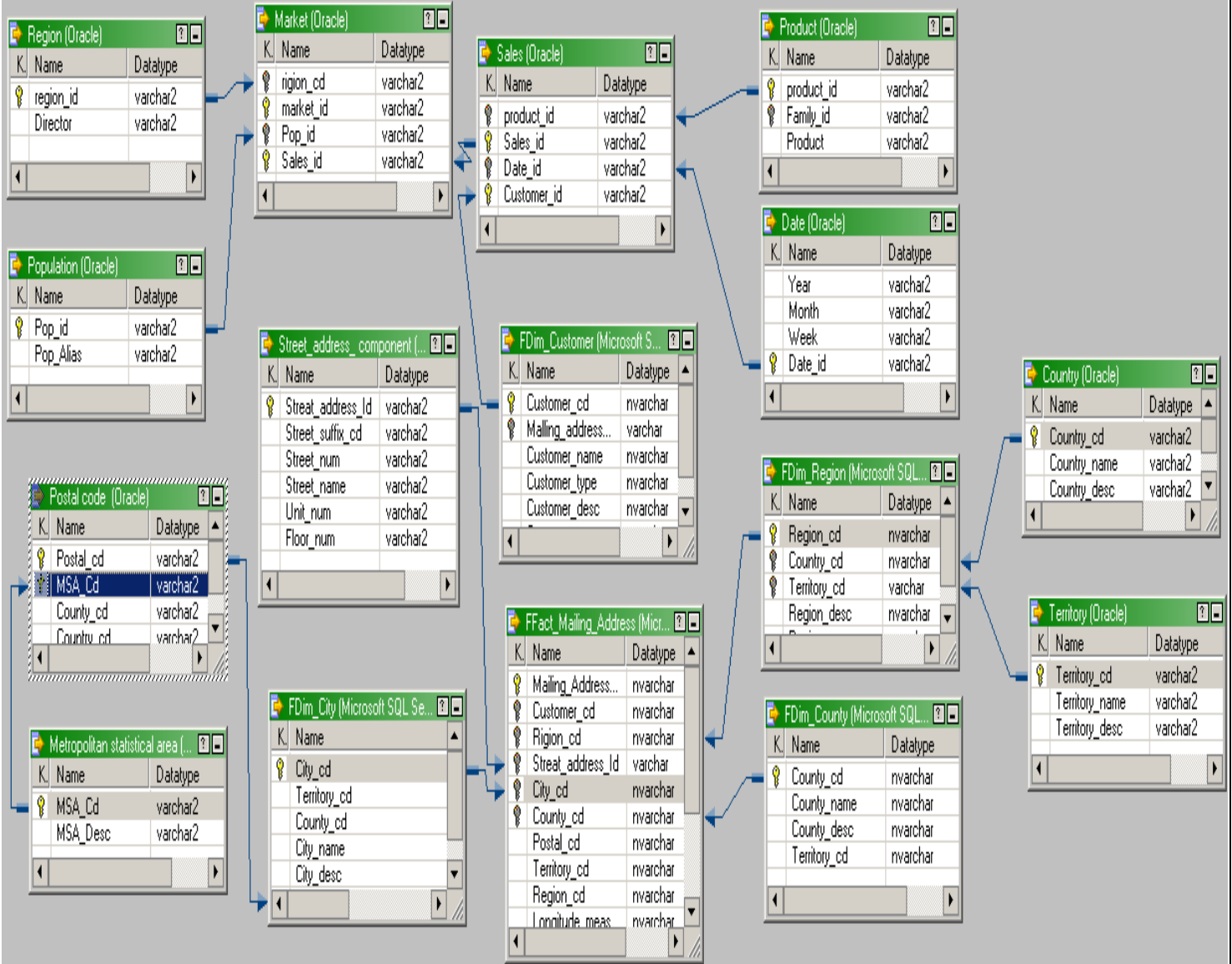
www.bisptrainings.com
www.hyperionguru.com
learnodi.blogspot.com
essbaseguru-amit.blogspot.com
learnplanning.wordpress.com
learnsqlquery.wordpress.com
learncognosreports.wordpress.com
bispsolutions.wordpress.com
odinetwork.blogspot.com
learnsoa.wordpress.com
bispinformatica.wordpress.com
bispteradata.wordpress.com

Table of Contents

<u>Customer Correspondence Data Model.....</u>	<u>03</u>
<u>List of Tables</u>	<u>04</u>
<u>Create table Syntax.....</u>	<u>05</u>
<u>Description of table content.....</u>	<u>06</u>

BISPR

Customer Correspondence Data Model



List of Tables

- Region
- Market
- Sales
- Product
- Population
- Date
- FDim_customer
- Street_address_component
- Postal_code
- Country
- Fdim_Region
- Territory
- Fdim_County
- FFact_mailing_address
- FDim_city
- Metropolitan_statcal_area

SYNTAX OF "CREATE TABLE" SCRIPT :-

```
CREATE (SET|MULTISET) TABLE <database_name> . <table_name>,
    FALLBACK | NO FALLBACK ,
    BEFORE JOURNAL | AFTER JOURNAL
    WITH JOURNAL TABLE = <TABLE-NAME>,
    FREE SPACE <NUMBER> PERCENT,
    DATABLOCKSIZE=<NUMBER>BYTES | {(MINIMUM|MAXIMUM) DATABLOCKSIZE}
    (
<column_name2> <data_type> <size> <column_level_attribute> <constraint>,
<column_name2> <data_type><size> <column_level_attribute> <constraint>,
<column_name N> <data_type><size> <column_level_attribute><constraint>
    )
    CONSTRAINT <constraint name> CHECK <Comparison>
    CONSTRAINT <constraint name > FOREIGN KEY <column_name| Column_list>
    [WITH [NO] CHECK OPTION] REFERENCES < table_name> (column name|
column list)
    [constraint <constraint name > UNIQUE [PK] <COLUMN_NAME | COLUMN LIST>
```

CREATE- Create Keyword is used to create and reserve space for new table, user or database.

SET – Row duplication not allow.

MULTISET - Duplication of rows allow. Not used with primary indexes.

TABLE – Table is a keyword. Which shows particular operation will be perform on specific table.

database_name - User define database. Followed by table name . e.g. samples, financial etc.

.(dot)- differentiate between database name and table name.

table_name – user define table name. must be unique.

FALLBACK - The benefits of Fallback include protecting your data from hardware (disk) failure, protecting your data from software (node) failure, automatic recovery and minimum recovery time after repairs or fixes are complete.

When a table is created, or any time after its creation, the user may specify whether or not the system should keep a fallback copy. If Fallback is specified, it is automatic and transparent to the user.

Fallback guarantees that the two copies of a row will always be on different AMPs.

It is costlier than NO fallback, they are: twice the disk space for storage and twice the I/O for Inserts, Updates, and Deletes.

NO FALLBACK - Do not save another copy of data.

JOURNAL - started automatically when the system has a failed or down AMP.

BEFORE JOURNAL-

AFTER JOURNAL-

WITH JOURNAL TABLE = <TABLE-NAME>

FREE SPACE <NUMBER> PERCENT

DATABLOCKSIZE=<NUMBER>BYTES

(MINIMUM|MAXIMUM DATABLOCKSIZE

column_name1- should not a keyword. Tow column names should not be same in one table.

data_type- Any valid data type.

size

column_level_attribute

Constraint:-

CONSTRAINT <constraint name> CHECK <Comparison>

CONSTRAINT <constraint name > FOREIGN KEY <column_name| Column_list>

WITH [NO] CHECK OPTION] REFERENCES < table_name> (column name | column list)

[constraint <constraint name > UNIQUE [PK] <COLUMN_NAME |COLUMN LIST>

Index :- Access rows from a table without having to search the entire table.

Unique Primary Index (UPI):- Unique Primary Index (or Primary index) [index_name] <column name| list> [PARTITION BY <partition_expression>

Unique Primary Index:- UPI is used for uniform distribution of the rows of that table. Index choice of column(s) is Unique.

Accessing data:- one amp access can return at most one row.

NUPI or Non-UPI :- Column(s) which have non-unique values (duplication possible) in Non – UPI. While not a guarantor of uniform row distribution, the uniqueness of the index will determine the degree of uniformity of the distribution.

No UPI:- distribute data between AMP's based on random generator code.

1. Region

Region table is connecting with Market Table. This table has two fields containing region id and name of corresponding director. Region id is VARCHAR of size 10. PI can't be null in Table.

Region					
Column name	Description	Data type	Size	Attribute	Constraint
Region_id		VARCHAR	10	PRIMARY INDEX (PI) And PRIMERY KEY (PK)	NOT NULL
Director	Name of region Director	VARCHAR	20	-	

```
create table samples.Region,
```

```
FALLBACK,
```

```
NO BEFORE JOURNAL,
```

```
Teradata 13.0 Create table Script |
```

©Business Intelligence Solution Providers

NO AFTER JOURNAL

```
(  
Region_id VARCHAR(10) NOT NULL,  
Director VARCHAR(20)  
)  
UNIQUE PRIMARY INDEX(Region_id);
```

2. **Market:** Market table connect with sales table, region table and population table. Market id is PK and PI. Market table containing 3 FKs of 3 different tables. PI can't be null in Table.

Market					
Column_name	Description	Data_type	Size	Attribute	Constraint
Market_id		VARCHAR	10	PI & PK	NOT NULL
Pop_id	PK in Population table	VARCHAR	10	FKEY	
Region_id	PK in Region table	VARCHAR	10	FKEY	
Sales_id	PK in sales table	VARCHAR	20	FKEY	

```
create table Samples. Market,  
FALLBACK,  
NO AFTER JOURNAL,
```

```

NO BEFORE JOURNAL (
Market_id VARCHAR(10) NOT NULL,
Pop_id VARCHAR(10) NOT NULL,
Region_id VARCHAR(10) NOT NULL,
Sales_id VARCHAR(20) ,
FOREIGN KEY (Pop_id ) REFERENCES SAMPLES.Population( Pop_id ),
FOREIGN KEY (Region_id) REFERENCES SAMPLES.Region( Region_id ) ,
FOREIGN KEY (Sales_id) REFERENCES SAMPLES. Sales (Sales)
)
UNIQUE PRIMARY INDEX (Market_id);

```

3. **Sales:** Sales table connected with Product table, Market table, customer table(FDim_customer) and Date table. PI can't be null in Table. Date_id is only date type here and remaining all VARCHAR type. Date format is 'mm/dd/yyyy' and size in memory is 4 bytes. Sales table work as a Fact table in this data model.

Sales					
Column_name	Description	Data_type	Size	Attribute	Constraint
Product_id	Pk in Product table	VARCHAR	10	FK	
Customer_id	Pk in FDim_Customer table	VARCHAR	10	FK	
Date_id	Pk in Date table	DATE 'mm/dd/yyyy'		FK	
Sales_id	Pk in sales table	VARCHAR	10	PI and PK	NOT NULL

```

create table Samples. Sales (
sales _id VARCHAR(10),
Product_id VARCHAR(10),
Customer_id VARCHAR(10),

```

```

Date_id DATE FORMAT 'mm/dd/yyyy' ,
FOREIGN KEY (Product _id ) REFERENCES SAMPLES. Product (Product _id ),
FOREIGN KEY (Customer _id ) REFERENCES SAMPLES. FDim_Customer (Customer
_id ) ,
FOREIGN KEY (Date _id) REFERENCES SAMPLES. Date (Date_id)
)
UNIQUE PRIMARY INDEX (sales _id);

```

4. **Product** : Product table connected with only sales table. Table containing product name and id of product. PI can't be null in Table. All columns are VARCHAR type.

Product					
Column_name	Description	Data_type	Size	Attribute	Constraint
Product_id		VARCHAR	10	P KEY and PI	NOT NULL
Family_id	PK in Family Table	VARCHAR	10	FOREIGN KEY	
PRODUCT	Name of product	VARCHAR	20		

```

create table samples. product,
FALLBACK,

```

```

NO BEFORE JOURNAL,
NO AFTER JOURNAL
(
product_id VARCHAR(10) NOT NULL,
family_id VARCHAR(10),
product VARCHAR(20),
FOREIGN KEY ( family_id ) REFERENCES SAMPLES. family( family_id )
)
UNIQUE PRIMARY INDEX(product_id) ;

```

5. **Population:** Population table connected with only Market table. Table containing population id and new name (alias). Aliasing removes circular join problem in data accessing. PI can't be null in Table. All columns are VARCHAR type.

Population

Column_name	Description	Data_type	Size	Attribute	Constraint
Pop_id		VARCHAR	10	PRIMARY KEY and PI	NOT NULL
Pop_Alias	Second name	VARCHAR	20	-	

```

create table Samples.population,
FALLBACK,
NO BEFORE JOURNAL,
NO AFTER JOURNAL

```

```

(
Pop_id VARCHAR(10) NOT NULL,
Pop_alias VARCHAR(20)
)
UNIQUE PRIMARY INDEX (Pop_id);

```

6. **Date:** Date table connected with only sales table. It contained year, month and week separately and date id as a PI. Date id 's format is "mm/dd/yyyy" e.g. 02/06/2011 , 12/05/1968 etc. PI can't be null in Table. Week s are w1, w2, w3,w4 and w5 in a month.

Date					
Column_name	Description	Data_type	Size	Attribute	Constraint
Year	All four values e.g. 1960	VARCAHR	4		
Month	Two values(01-12) e.g. 02,12	VARCAR	8		
Week	Week (W01 to W05)	VARCHAR	3		
Date_id	"mm/dd/yyyy"	DATE		PK and PI	NOT NULL

```

create table samples. Date,
FALLBACK,
NO AFTER JOURNAL,NO BEFORE JOURNAL
(
Date_id DATE FORMAT 'mm/dd/yyyy' NOT NULL,
year VARCHAR(4),
month VARCHAR(8),
week VARCHAR(3)
)
UNIQUE PRIMARY INDEX(date_id) ;

```

7. **FDim_customer**: FDim_customer table containing customer details. Table connected with only sales table. PI can't be null in Table. Customer cd and name is mandatory fields. Customer type is Permanent, general, global etc.

FDim_Customer					
Column_name	Description	Data_type	Size	Attribute	Constraint
Customer_cd		VARCHAR	10	PRIMERY KEY	NOT NULL
Customer_name		VARCHAR	20		NOT NULL
Customer_Desc	Decription	VARCHAR	20		
Customer_type		VARCHAR	20		
Mailing_address		VARCHAR	20		

```
create table Samples.Dim_customer,  
FALLBACK,  
NO AFTER JOURNAL,  
NO BEFORE JOURNAL  
(  
  Customer_id VARCHAR(10) NOT NULL,  
  Customer_name VARCHAR(20) NOT NULL,  
  Customer_desc VARCHAR(20)  
)  
UNIQUE PRIMARY INDEX (customer_id);
```

8. **Street_address_component:** Street address component containing street details . Table connected with only FFact_mailing_Address and used as a part of customer address. PI can't be null in Table.

FDim_Customer					
Column_name	Description	Data_type	Size	Attribute	Constraint
Street_address_id		VARCHAR	10	PI and PK	NOT NULL
Street_suffix_cd		VARCHAR	20	FK	
Street_num		VARCHAR	20		
Street_name		VARCHAR	20		
Unit_num		Number			
Floor_num		Number			

```

create SET table Samples.Street_Address_component,
FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL(
  Street_address_id VARCHAR(20),
  Street_suffix_cd CHAR(10),

```



```

Street_num VARCHAR(20),
Street_name VARCHAR(20),
Unit_num Number,
Floor_num Number) Unique Primary Index (Street_address_id );

```

9. Postal_code

Postal code used in address part. Table connected with FDim_city table and Metropolitan statistical area. PI can't be null in Table.

Postal_code					
Column_name	Description	Data_type	Size	Attribute	Constraint
Postal_cd		VARCHAR	10	PI and PK	NOT NULL
Country_cd		VARCHAR	10	FK	
County_cd		VARCHAR	10	fk	
MSA_cd		VARCHAR	10	Fk	

```

create SET table Samples.FDim_Postal,
FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL (
    Postal_cd VARCHAR(10),

```

```

Country_cd VARCHAR(10),
County_cd VARCHAR(10),
MSA_cd VARCHAR(10),
FOREIGN KEY ( Country_cd ) REFERENCES SAMPLES. Country( Country_cd
),
FOREIGN KEY ( County_cd ) REFERENCES SAMPLES.
County( County_cd ),
FOREIGN KEY ( MSA_cd ) REFERENCES
SAMPLES.FDim_Metropolitan_Statistical_area(MSA_cd)
)UNIQUE PRIMARY INDEX (Postal_cd);

```

10. Country: Postal code used in address part. It uniquely identified the city. Table connected with FDim_city table and Metropolitan statistical area. PI can't be null in Table.

Country					
Column_name	Description	Data_type	Size	Attribute	Constraint
Country_cd		VARCHAR	10	PI and PK	NOT NULL
Country_name		VARCHAR	10		NOT NULL
Country_desc		VARCHAR	10		

```
create SET table Samples.FDim_Country,
```

```

FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL
(
Country_cd VARCHAR(10),
Country_name VARCHAR(20),
Country_desc VARCHAR(20)
)
UNIQUE PRIMARY INDEX (Country_cd);

```

11. Fdim_Region: Fdim_region table contained region details. Three tables Country table , territory table and FFact_mailing_address table connect with this table.

FDim_Region					
Column_name	Description	Data_type	Size	Attribute	Constraint
Region_cd		VARCHAR	10	PI and PK	NOT NULL
Country_cd		VARCHAR	20	Fk	
Region_desc		VARCHAR	20		
Region_name		VARCHAR	20		

```
create SET table Samples.FDim_Region,  
FALLBACK,  
NO AFTER JOURNAL,  
NO BEFORE JOURNAL  
(  
  Region_cd VARCHAR(10),  
  Country_cd VARCHAR(10),  
  Region_desc VARCHAR(20),  
  Region_name VARCHAR(20),  
  FOREIGN KEY ( Country_cd ) REFERENCES SAMPLES.  
Country( Country_cd )  
)  
UNIQUE PRIMARY INDEX (Region_cd);
```

12. Territor: Territory table connected with only FDim_Region table. This table contain territory details.

Territory					
Column_name	Description	Data_type	Size	Attribute	Constraint
Territory_cd		VARCHAR	10	PI and PK	NOT NULL
Territory_name		VARCHAR	20		
Territory_desc		VARCHAR	20		

```

create SET table Samples.FDim_Territory,
FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL
(
Territory_cd VARCHAR(10),
Territory_name VARCHAR(20),
Territory_desc VARCHAR(20)
)
UNIQUE PRIMARY INDEX(Territory_cd );

```

13. Fdim_County: Fdim_County table connected with FFact_mailing_Address table. This table contain county related information and territory code also.

Fdim_county					
Column_name	Description	Data_type	Size	Attribute	Constraint
County_cd		VARCHAR	10	PI and PK	NOT NULL
County_name		VARCHAR	20		
County_desc		VARCHAR	20		
Territory_cd		VARCHAR	10	Fk	

```

create SET table Samples.FDim_County,
FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL
(
County_cd VARCHAR(10),

```

```

County_name VARCHAR(20),
County_desc VARCHAR(20),
Territory_cd VARCHAR(10),

FOREIGN KEY ( Territory_cd) REFERENCES SAMPLES.
Territory( Territory_cd )
)

UNIQUE PRIMARY INDEX (County_cd);

```

14. Fact_mailing_address

Fact_mailing_address					
Column_name	Description	Data_type	Size	Attribute	Constraint
Mailing_Address_id		VARCHAR	10	PI and PK	NOT NULL
Street_Address_id		VARCHAR	10	fk	
Customer_cd		VARCHAR	10	fk	
Country_cd		VARCHAR	10	Fk	
City_cd		VARCHAR	10	fk	
County_cd		VARCHAR	10	fk	
Postal_cd		VARCHAR	10	fk	
Territory_cd		VARCHAR	10	fk	
Region_cd		VARCHAR	10	fk	
Longitude_meas		VARCHAR	20		
Latitude_Meas		VARCHAR	20		
Address_Line_1		VARCHAR	20		
Address_Line_2		VARCHAR	20		
Address_Line_3		VARCHAR	20		

create SET table Samples.FFact_Mailing_Address,
FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL(
 Mailing_Address_id VARCHAR(20),
 Street_Address_id VARCHAR(20),
 Customer_cd VARCHAR(10),
 Country_cd VARCHAR(10),


```
City_cd VARCHAR(10),
County_cd VARCHAR(10),
Postal_cd VARCHAR(10),
Territory_cd VARCHAR(10),
Region_cd VARCHAR(10),
Longitude_meas VARCHAR(20),
Latitude_Meas VARCHAR(20),
Address_Line_1 VARCHAR(20),
Address_Line_2 VARCHAR(20),
Address_Line_3 VARCHAR(20),

FOREIGN KEY ( Customer_cd ) REFERENCES
SAMPLES.FDim_Customer( Customer_cd),

FOREIGN KEY (Street_Address_id) REFERENCES
SAMPLES.Street_address (Street_Address_id),

FOREIGN KEY ( Country_cd ) REFERENCES
SAMPLES.Country( Country_cd ),

FOREIGN KEY ( City_cd ) REFERENCES SAMPLES.City( City_cd ),

FOREIGN KEY ( County_cd ) REFERENCES
SAMPLES.County( County_cd ),

FOREIGN KEY ( Postal_cd ) REFERENCES
SAMPLES.Postal( Postal_cd),

FOREIGN KEY ( Territory_cd) REFERENCES
SAMPLES.Territory( Territory_cd ),

FOREIGN KEY ( Region_cd ) REFERENCES
SAMPLES.Region( Region_cd)
```

```
) UNIQUE PRIMARY INDEX (Mailing_Address_id);
```

15. FDim_city

Fdim_city					
Column_name	Description	Data_type	Size	Attribute	Constraint
City_cd		VARCHAR	10	PI and PK	NOT NULL
Territory_cd		VARCHAR	10	Fk	
County_cd		VARCHAR	10	Fk	
City_name		VARCHAR	10		
City_desc		VARCHAR	20		

```
create table Samples.FDim_City,  
FALLBACK,  
NO AFTER JOURNAL,  
NO BEFORE JOURNAL  
  
(  
  
    City_cd VARCHAR (10),
```

```

Territory_cd VARCHAR (10),
County_cd VARCHAR (10),
City_name VARCHAR(20),
City_desc VARCHAR(20),

FOREIGN KEY ( Territory_cd) REFERENCES
SAMPLES.Territory( Territory_cd ),

FOREIGN KEY ( County_cd) REFERENCES SAMPLES.County(County_cd )
) UNIQUE PRIMARY INDEX (City_cd);

```

16. Metropolitan_statistical_area

Fdim_city					
Column_name	Description	Data_type	Size	Attribute	Constraint
MSA_Cd		VARCHAR	10	PI and PK	NOT NULL
MSA_Desc		VARCHAR	20		

```

create SET table Samples. Metropolitan_Statistical_area,
FALLBACK,
NO AFTER JOURNAL,
NO BEFORE JOURNAL
(

```

```
MSA_Cd VARCHAR (10),  
MSA_Desc VARCHAR(20)  
)  
UNIQUE PRIMARY INDEX (MSA_Cd);
```

BISP