



## DataMart Reference

---

This appendix contains the following topics:

- [Custom Reporting Data Model](#), page 1
- [Data Mart Administration](#), page 7

## Custom Reporting Data Model

This section describes the various schema designs for data mart.

### Data Mart Schema Design and the Business View

The data mart schema was designed to be used with IBM Cognos Framework Manager and the business view used by the IBM Cognos reporting tools (Query Studio and Report Studio, represented as Ad-Hoc Reports and Report Designer in the Service Catalog Advanced Reporting module). As such, technical documentation in the Data Mart chapter outlines the contents and relationships of the query subjects, but does not cover the underlying data model, which is not exposed to reporting users or designers.

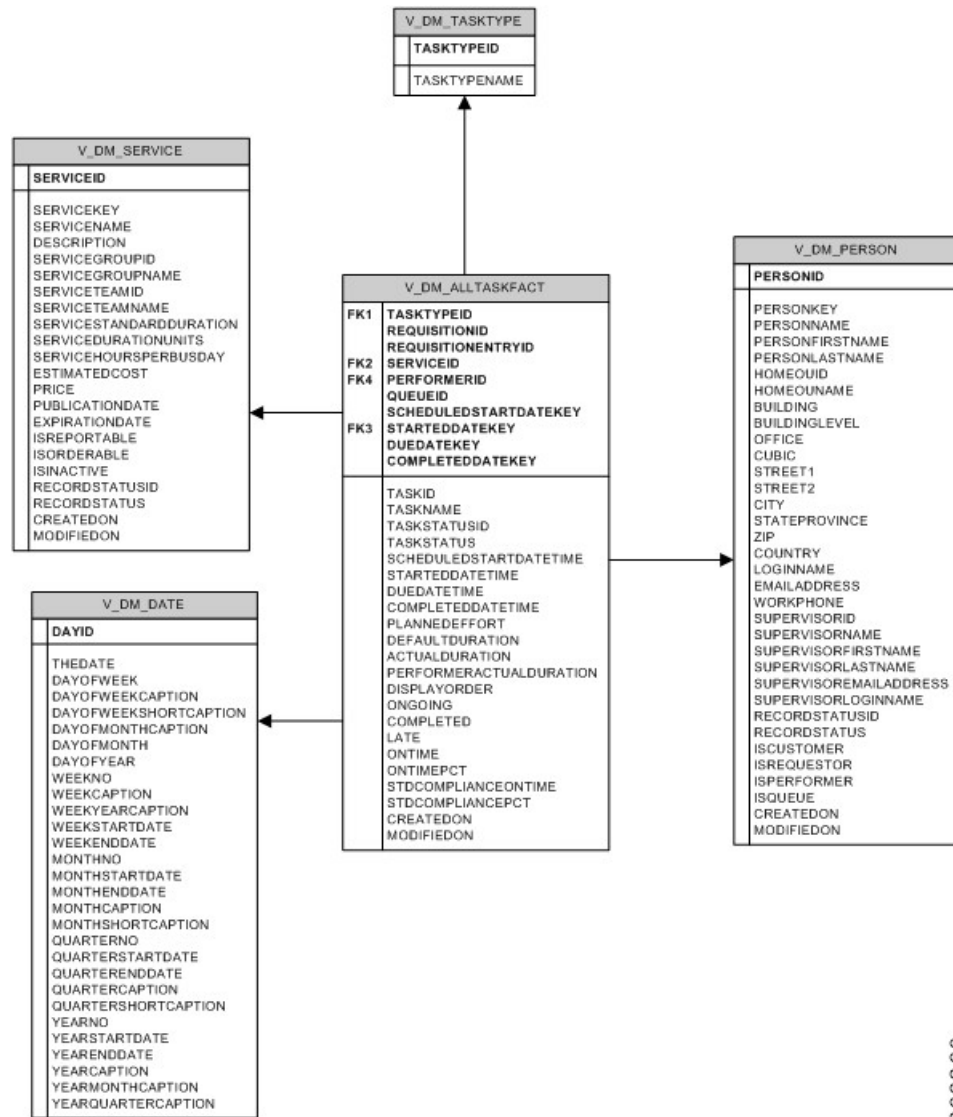
This chapter is meant for technical personnel who wish to investigate the physical data model underlying the business view exposed via the Cognos tools. In particular, it may be useful to map the names for the database objects in the diagrams that follow to corresponding names of query subjects. In the following diagrams:

- The Queue query subject is based on the V\_DM\_PERSON database view, and is joined to Task views via the QUEUEID column.
- The Performer query subject is also based on the V\_DM\_PERSON.PERSONID business view and is joined with Task views via the PERFORMERID column.
- The Customer and Requestor query subjects (dimensions) are also based on the V\_DM\_PERSON database view, and are joined to the Requisition views via the CUSTOMERID and REQUESTORID, respectively
- All date dimensions (CalendarStartDate, CalendarDueDate, CalendarClosedDate and CalendarScheduledDate) are based on the V\_DM\_DATE business view. They in turn are joined with the Task and Requisition views by the STARTEDDATEKEY, DUE DATEKEY, COMPLETEDDATEKEY and SCHEDULEDSTARTDATEKEY respectively.

- All fact tables also include a join (not depicted in the diagrams) to Bundled Service data, available in V\_DM\_SERVICEBUNDLE.

## Star Schema Design for AllTaskFact (All Tasks)

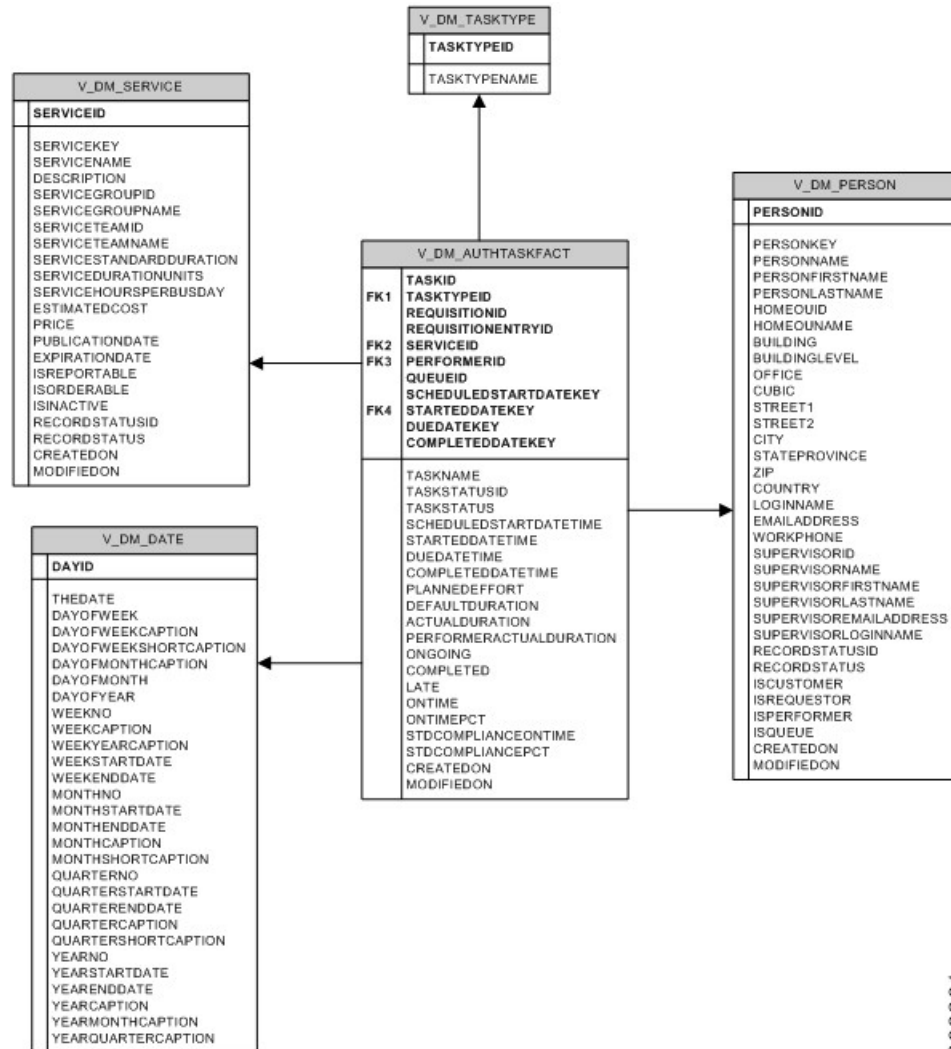
Figure 1: Schema Design for AllTaskFact



362360

## Star Schema Design for AuthTaskFact (Authorization Tasks)

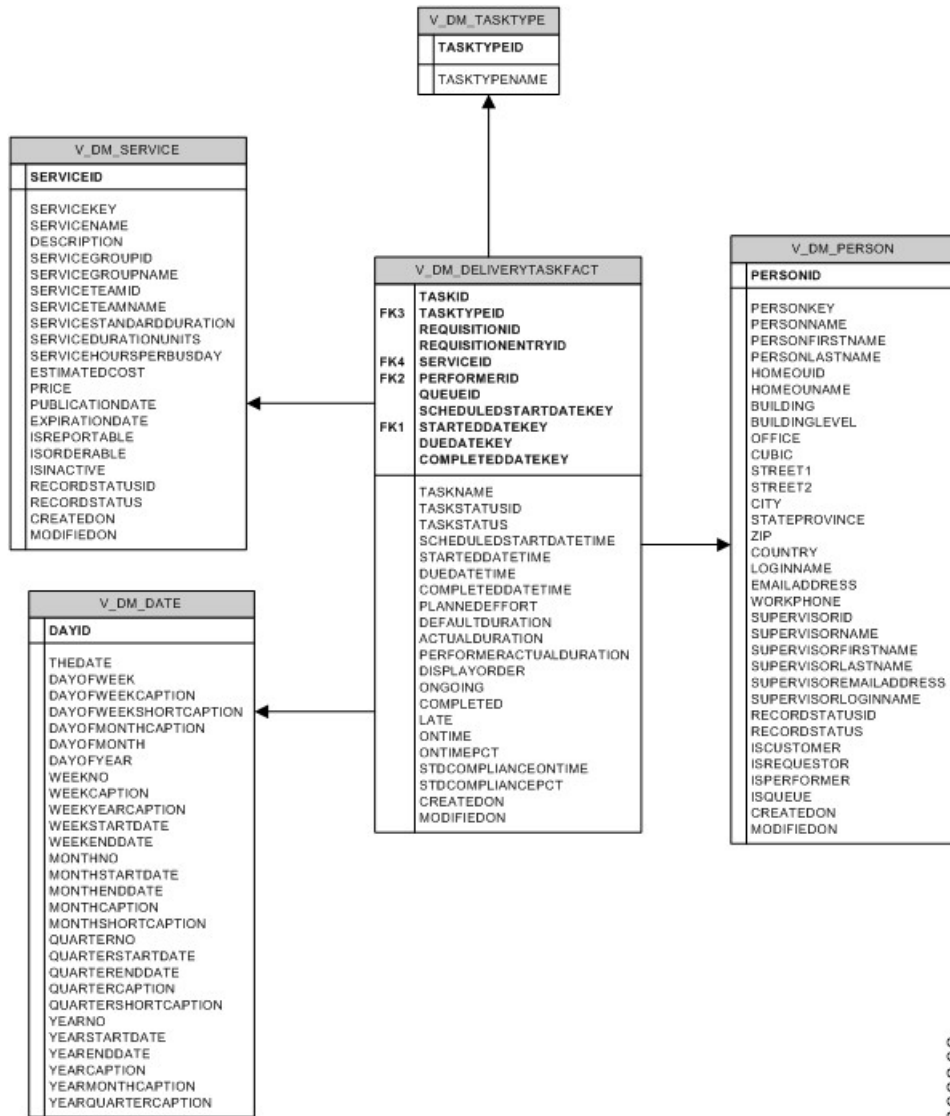
Figure 2: Schema Design for AuthTaskFact



362361

## Star Schema Design for DeliveryTaskFact (Delivery Tasks)

Figure 3: Schema Design for DeliveryTaskFact



362362

**Figure 4: Schema Design for ServiceRequestFact**



The above diagrams are incomplete. All fact tables actually have one-to-many relationships to dictionary- and service-based dimension tables. For services that do not contain any grid dictionaries, the relationships are always one-to-one (see the `DM_FDR_SERVICETABLE_n` XREF for more information). These relationships are implemented via the column `REQUISITIONENTRYID`, which is present in all fact and dimension tables, but is not exposed in the business view's dimensions.

## Cisco Prime Service Catalog 12.1 Reporting Guide

The dictionary- and service-based query subjects which appear in the data mart under the Dictionaries and Services folders correspond to tables in the physical database named DM\_FDR\_DICTIONARYTABLE\_n and DM\_FDR\_SERVICETABLE\_n, respectively, where

- n is a sequential number starting with 1, and
- the number of tables of each type is determined by the number specified when the Advanced Reporting option is configured.

The mapping between the physical tables and the reportable objects is maintained in the tables DM\_FDR\_DICTIONARYMETADATA and DM\_FDR\_SERVICEMETADATA. These tables are populated when an object is designated as reportable, and used by the ETL processes to dynamically adjust the business view of the data mart to include the reportable objects.

If you wish to supplement the use of the IBM Cognos Business Intelligence tools, bypassing the business view offered by the Ad-Hoc Reports and Report Designer modules of Advanced Reporting, you can do so by interrogating those METADATA tables and constructing database VIEWS which match the dictionary- or service-based query subjects. A sample (SQLServer-specific) SQL statement for building a database view of the MemoryDetails dictionary is shown below. This is (obviously) just a starting point for such an effort.

```
SELECT distinct 'CREATE VIEW ' + dictionaryname + ' (' AS SQLColumn,
  'A 0' AS DestinationColumnName
FROM dm_fdr_dictionarymetadata
WHERE dictionaryname = 'MemoryDetails'
UNION
SELECT ' ' + dictionaryattributename, 'A ' + DestinationColumnName
FROM dm_fdr_dictionarymetadata
WHERE dictionaryname = 'MemoryDetails'
  AND DestinationColumnName = 'FIELD1'
UNION
SELECT ', ' + dictionaryattributename, 'A ' + DestinationColumnName
FROM dm_fdr_dictionarymetadata
WHERE dictionaryname = 'MemoryDetails'
  AND DestinationColumnName <> 'FIELD1'
UNION
SELECT ', REQUISITIONENTRYID, REQUISITIONID, SERVICEID', 'A Y'
UNION
SELECT ') AS SELECT' , 'A Z'
UNION
SELECT ' ' + DestinationColumnname, 'B ' + DestinationColumnName
FROM dm_fdr_dictionarymetadata
WHERE dictionaryname = 'MemoryDetails'
  AND DestinationColumnName = 'FIELD1'
UNION
SELECT ', ' + DestinationColumnname, 'B ' + DestinationColumnName
FROM dm_fdr_dictionarymetadata
WHERE dictionaryname = 'MemoryDetails'
  AND DestinationColumnName <> 'FIELD1'
UNION
SELECT ', REQUISITIONENTRYID, REQUISITIONID, SERVICEID', 'B Y'
UNION
SELECT distinct 'FROM ' + DestinationTableName, 'B Z'
FROM dm_fdr_dictionarymetadata
WHERE dictionaryname = 'MemoryDetails'
ORDER BY DestinationColumnName
```

Executing that SQL Statement yields a SQL Command like:

```
CREATE VIEW MemoryDetails (
  CurrentMemorySize
, MemoryType
, MemorySizeNeeded
, Reason
, REQUISITIONENTRYID, REQUISITIONID, SERVICEID
) AS SELECT
  FIELD1
```

```
, FIELD2
, FIELD3
, FIELD4
, REQUISITIONENTRYID, REQUISITIONID , SERVICEID
FROM DM_FDR_DICTIONARYTABLE_18
```

## Data Mart Administration

This section is intended for use by people who need to know technical implementation details of the Service Catalog reporting solution. These include report administrators, responsible for the reporting environment; support personnel who may need to report issues to Cisco TAC; and analysts and designers who want to investigate the options of customizing or enhancing components of the reporting options.

## Data Mart Source Data

The accuracy and completeness of the data in the data mart is critical. In order to gauge these qualities, it is critical for the source of the data in the data mart to be documented. The table below shows the tables in the OLTP database which contributed to data found in the query subjects available in the Custom Data Mart.

**Table 1: DataMart Source Data Table**

| Data Mart Query Subject                                 | OLTP Database Tables   |
|---|--|
| Dictionary Data (user-specific reportable dictionaries) | DefDataDictionary<br>DefObjectDictionaries<br>TxRequisitionEntry   |
| Service form Data (user-specific reportable services)   | DefService<br>DefServiceExtension<br>DefDataDictionary<br>DefObjectDictionaries<br>DefData<br>TxRequisitionEntry |
| Service   | DefService<br>DefArea<br>DirOrganizationalUnit   |
| Dictionary  | DefDataDictionary<br>DefDictionaryGroup  |
| Keyword   | DefKeyword   |
| Group   | DirGroup   |
| Organization  | DirOrganizationalUnit  |

| Data Mart Query Subject | OLTP Database Tables  |
|-------------------------|---|
| Customer                | DirPerson   |
| Requestor               | DirOrganizationalUnit   |
| Performer               | DirNetworkInfo  |
| Person                  | DirLocation   |
| Queue                   | DirAddress  |
| ServiceRequestFact      | TxRequisitionEntry<br>TxRequisition   |
| ServiceTaskFact         | TxRequisitionEntry<br>TxActivity  |
| RequisitionTaskFact     | TxRequisition<br>TxActivity   |
| TaskEffortEntryFact     | TxBilling<br>DefExpenditureType<br>DefBillingClass<br>DefUnitType<br>DirOrganizationalUnit<br>DirPerson |

## Service-Form Reporting Metadata

The data mart ETL processes use a set of tables in the data mart database to configure the ServiceData and DictionaryData dimensions that are exposed to users in the Ad-Hoc Reports and Report Designer options. These tables are created when the Custom Reports Package is installed and populated when data is loaded into the data mart.

These metadata tables are not exposed in the Cognos framework. The contents of these tables are used to specify the dynamically defined dictionary and service-based dimensions which appear as the business view of the data in the Custom Reports Project.

These tables are described below. The description for each column uses an abstract data type; the actual data types will vary, depending on the database (Oracle or SQLServer) in which the data mart resides.

### DM\_FDR\_ETLDictionaryMetadata

The DM\_FDR\_ETLDictionaryMetadata table maps a particular reportable dictionary (identified by its DictionaryID) to the DM\_FDR\_DICTIONARY\_n table in which dictionary data is stored (DictionaryTableName). It also tracks how many date, numeric, and varchar fields are used within that dictionary and consequently, within the corresponding data mart table.



**Table 2: DM\_FDR\_ETLDICTIONARYMETADATA Table**

| Column Name                | Data Type   | Description   |
|----------------------------|-------------|---|
| DictionaryID               | Integer     | Unique identifier for the dictionary                            |
| DictionaryTableName        | Varchar(50) | Name of the data mart table where the dictionary data is stored |
| LastDictionaryDateField    | Integer     | Last datetime type field used in the dictionary table           |
| LastDictionaryNumericField | Integer     | Last numeric type field used in the dictionary table            |
| LastDictionaryVarcharField | Integer     | Last character type field used in the dictionary table          |

## DM\_FDR\_DICTIONARYMETADATA

The DM\_FDR\_DICTIONARYMETADATA table maps individual dictionary fields (attributes) (identified by DictionaryID and DictionaryAttributeName) to specific columns of the dictionary tables. For example, the attribute "LastName" in the dictionary RC\_REQUESTEDBY may be mapped to (that is, actually stored in) the data mart table DM\_FDR\_DICTIONARYTABLE\_10, in the field "Field2".

**Table 3: DM\_FDR\_DICTIONARYMETADATA Table**

| Column Name             | Data Type    | Description   |
|-------------------------|--------------|---|
| DestinationColumnName   | Varchar(100) | Column name of the table where this attribute is stored |
| DestinationTableName    | Varchar(200) | Table name where the dictionary information is stored   |
| DictionaryAttributeName | Varchar(100) | Name of the attribute in the dictionary                 |
| DictionaryAttributeType | Varchar(100) | Data type of the attribute in the dictionary            |
| DictionaryID            | Integer      | Dictionary ID   |
| DictionaryName          | Varchar(200) | Dictionary name   |
| DictionaryAttributeID   | Integer      | Dictionary attribute ID                                 |

## DM\_FDR\_ETLMETADATA

The DM\_FDR\_ETLMETADATA table holds information on configuring the Custom Reports Package that was specified when the Reporting options was installed, as well as data about the data extraction process (ETL) and last update information to for scheduled run of the ETL Process.

**Table 4: DM\_FDR\_ETLMETADATATable**

| Column Name                 | Data Type   | Description   |
|-----------------------------|-------------|---|
| DictionaryTablePattern      | Varchar(50) | Pattern for the name of the dictionary tables; by default DM_FDR_DICTIONARY_; specified via Advanced Reporting installation. Dictionary tables will have the specified name followed by a number. |
| Field Pattern               | Varchar(50) | Field pattern for both dictionary and service tables; by default FIELD_.  |
| LastDictionaryTableSequence | Integer     | Sequence number of the last dictionary table currently used in the data mart.   |
| LastDictRequisitionID       | Integer     | Last requisition ID for which data was written to the dictionary table.   |
| LastProcessedTime           | Datetime    | Date and time when the ETL to load the custom Reports Package was most recently run.  |
| LastServiceTableSequence    | Integer     | Sequence number of the last service table used in the data mart.  |
| LastSvcRequisitionID        | Integer     | Last requisition ID of the service data.  |
| ServiceTablePattern         | Varchar(50) | Pattern for the name of the service tables; by default DM_FDR_SERVICE_; specified via Advanced Reporting installation.  |
| DictionaryTotalDateField    | Integer     | The total number of date fields used for dictionary and service tables.   |
| DictionaryTotalNumericField | Integer     | The total number of numeric fields used for dictionary and service tables.  |
| DictionaryTotalVarcharField | Integer     | The total number of character fields used for dictionary and service tables.  |
| ServiceTotalDateField       | Integer     | The total number of date fields used for service tables.  |
| ServiceTotalNumericField    | Integer     | The total number of numeric fields used for service tables.   |
| ServiceTotalVarcharField    | Integer     | The total number of character fields used for service tables.   |

## DM\_FDR\_ETLSERVICEMETADATA

The DM\_FDR\_ETLSERVICEMETADATA table holds the information about the tables that are used to store information about each reportable service.

**Table 5: DM\_FDR\_ETLSERVICEMETADATA Table**

| Column Name             | Data Type | Description   |
|-------------------------|-----------|---|
| LastServiceDateField    | Integer   | Last datetime type field used in the service table                    |
| LastServiceNumericField | Integer   | Last numeric type field used in the service table                     |
| LastServiceVarcharField | Integer   | Last character type field used in the service table                   |
| ServiceID               | Integer   | Unique Identifier assigned to the service                             |
| ServiceTableName        | Varchar   | Name of the data mart database table where the service data is stored |

## DM\_FDR\_SERVICEMETADATA

The DM\_FDR\_SERVICEMETADATA table holds the Metadata information about which service attributes are populated in which columns of the service tables and, also the usage name of each of the columns.

**Table 6: DM\_FDR\_SERVICEMETADATA Table**

| Column Name           | Data Type    | Description   |
|-----------------------|--------------|---|
| DestinationColumnName | Varchar(100) | Column name of the table where this attribute is stored |
| DestinationTableName  | Varchar(200) | Table name where the service information is stored      |
| ServiceAttributeName  | Varchar(100) | Name of the attribute in the service                    |
| ServiceAttributeLabel | Varchar(200) | Caption of the attribute in the service                 |
| ServiceAttributeType  | Varchar(100) | Attribute type  |
| ServiceAttributeID    | Integer      | Identifier for the attribute of the service             |
| ServiceID             | Integer      | Service ID  |
| ServiceName           | Varchar(200) | Name of the service                                     |

## Dynamically Defined Dimensions

The nature and number of the dictionaries and their attributes which are added to the Custom Reports Package is dynamically determined and may differ greatly at each Service Catalog installation. To support the required flexibility, the database which supports the Custom Reports Package includes a set of abstract data structures, which hold the dimensional data corresponding to the dictionaries, their attributes, and service configuration using dictionary (form) data. Dictionary contents are mapped to these tables via the DM\_FDR metadata tables explained above.

### DM\_FDR\_DICTIONARYTABLE\_n

A set of tables captures attributes (fields) for each reportable dictionary in the application. The number of these tables is configurable as part of the application installation, as well as the number of columns of each data type (character, numeric, or datetime).

Each table has the name DM\_FDR\_DICTIONARYTABLE (or alternate pattern supplied via the installation procedure), followed by a numeric suffix, \_n. Each table is numbered sequentially, starting with 1. Each instance of this table represents a reportable dictionary.

The DM\_FDR\_DICTIONARYTABLEs appear in the reporting tools as a set of dimensions within the DictionaryData folder. The name of each dimension is the caption of the corresponding dictionary. (For dictionaries with no caption, the dictionary name is used.) The attributes of the dimension are the fields which comprise the dictionary. The fields are numbered sequentially, starting with 1. The number of each type of field (character, numeric, or datetime) is specified via the application installation procedure.

**Table 7: DM\_FDR\_DICTIONARYTABLE\_n Table**

| Column Name           | Data Type    | Description                             |
|-----------------------|--------------|---|
| DictionaryID          | Integer      | Dictionary ID                           |
| RequisitionID         | Integer      | Requisition ID                          |
| RequisitionEntryID    | Integer      | Requisition Entry ID                    |
| Field1 through Fieldn | Varchar(200) | Varchar fields to hold dictionary data  |
| Fieldn+1 through n+m  | Numeric      | Numeric fields to hold dictionary data  |
| Fieldn+m+1 through..  | Datetime     | Datetime fields to hold dictionary data |

### DM\_FDR\_SERVICETABLE\_n

A set of tables captures data for each service which has been designated as reportable. The tables contain all fields in all dictionaries used in the service. The number of these tables is configurable as part of the application

installation, as well as the number of columns of each data type (character, numeric, or datetime). Each table is numbered sequentially, starting with 1.

Each table has the name DM\_FDR\_SERVICETABLE (or alternate pattern, as designated via the installation procedure), followed by a numeric suffix, \_n. Each instance of this table represents a reportable service.

The DM\_FDR\_SERVICETABLEs appear in the reporting tools as a set of dimensions within the ServiceData folder. The name of each dimension is the name of the corresponding service. The attributes of the dimension are the fields which comprise all dictionaries in the service. Fields are added to this table in the order in which their dictionary occurs in the service. Since the number of fields that may be accommodated in each table is limited (specified via the installation procedure, but physically limited by database constraints), the service table may not be complete—some fields, indeed some dictionaries, may be truncated. Therefore, DM\_FDR\_SERVICETABLEs should be used with care, especially if dictionaries with large numbers of fields are designated as reported, or if a great number of dictionaries are used in the same service.

**Table 8: DM\_FDR\_SERVICETABLE\_n Table**

| Column Name               | Data Type    | Description                          |
|---------------------------|--------------|--------------------------------------|
| ServiceID                 | Integer      | Service ID                           |
| RequisitionID             | Integer      | Requisition ID                       |
| RequisitionEntryID        | Integer      | Requisition Entry ID                 |
| Field1 through Field $n$  | Varchar(200) | Varchar fields to hold service data  |
| Field $n+1$ through $n+m$ | Numeric      | Numeric fields to hold service data  |
| Field $n+m+1$ through..   | Datetime     | Datetime fields to hold service data |

For a service configured with no reportable grid dictionaries, each request for the service (that is, requisition entry) is captured by the ETL process and inserted into the corresponding DM\_FDR\_SERVICETABLE as one row of data. However, for a service configured with one or more reportable grid dictionaries, the ETL process inserts multiple rows of data into the DM\_FDR\_SERVICETABLE table. The number of rows inserted corresponds to the greatest number of rows in any of the reportable grid dictionaries.

For example, consider a service with a reportable nongrid dictionary (Employee) and two reportable grid dictionaries (Contact, Address). Assume a request for this service has three rows of data in Contact, two rows of data in Address, and some data in the Employee dictionary. The form data captured in the service table for this service would look the following:

**Table 9: Form Data in Service Table**

| RequisitionEntryID | Employee.<br>FirstName | Employee.<br>LastName | Contact.<br>Type | Contact. Details | Address.<br>Line | Address.<br>City | Address.<br>State | Address.<br>Country |
|--------------------|------------------------|-----------------------|------------------|------------------|------------------|------------------|-------------------|---------------------|
| NNN                | John                   | Smith                 | Cell             | 650-123-4567     | 3333 Third St.   | San Mateo        | CA                | USA                 |
| NNN                | (NULL)                 | (NULL)                | Work             | 408-765-4321     | 1111 First St.   | San Jose         | CA                | USA                 |

| RequisitionEntryID | Employee.<br>FirstName | Employee.<br>LastName | Contact.<br>Type | Contact. Details    | Address.<br>Line | Address.<br>City | Address.<br>State | Address.<br>Country |
|--------------------|------------------------|-----------------------|------------------|---------------------|------------------|------------------|-------------------|---------------------|
| NNN                | (NULL)                 | (NULL)                | Email            | j.smith@company.com | (NULL)           | (NULL)           | (NULL)            | (NULL)              |

## Data Mart Database Objects

The following table lists the database tables and views used for the Service Catalog data mart and exposed to users via the Custom Reports package. The tables/views are mapped directly to corresponding query subjects.

**Table 10: Data Mart Database Objects Table**

| Data Mart Table/View     | Primary Key        | Query Subject/Description   |
|--------------------------|--------------------|---|
| DM_DEFSERVICE            | ServiceID          | Service information (which includes Service Group and Service Team information)   |
| DM_DEFDICTIONARY         | DictionaryID       | Dictionary information  |
| DM_PERSON                | PersonID           | Person information  |
| DM_DATE                  | DateID             | Calendar information  |
| VIEW_CUSTOMER            | CustomerID         | Customer information  |
| VIEW_REQUESTOR           | RequestorID        | Requestor (initiator) information   |
| VIEW_PERFORMER           | PerformerID        | Information about the person who performs a task  |
| VIEW_QUEUE               | QueueID            | The queue to which a task is assigned   |
| VIEW_CALENDARCLOSEDDATE  | ClosedDateID       | The date a task or requisition was closed and the accompanying date hierarchy   |
| VIEW_CALENDARDUEDATE     | DueDateID          | The date a task or requisition was due and the accompanying date hierarchy  |
| VIEW_CALENARSCHEDULEDATE | ScheduledDateID    | The date a task or requisition was scheduled to start and the accompanying date hierarchy   |
| VIEW_CALENDARSTARTEDDATE | StartedDateID      | The date a task or requisition was started and the accompanying date hierarchy  |
| DM_REQUISITIONENTRYFACT  | RequisitionEntryID | Individual requisition entries (services) ordered   |
| DM_SERVICETASKFACT       | ServiceTaskID      | Tasks performed at the service (requisition entry) level, including delivery tasks, ad-hoc tasks and service group authorizations and reviews |

| Data Mart Table/View                    | Primary Key        | Query Subject/Description   |
|---|--------------------|---|
| DM_REQUISITIONTASKFACT                  | RequisitionTaskID  | Tasks performed at the requisition level, including financial authorizations and organizational unit authorizations and reviews |
| DM_TASKEFFORTENTRYFACT                  | EffortEntryID      | Effort expended in the performance of a delivery task   |
| DM_FDR_DICTIONARY_n<br>(n=1, 2, 3, ...) | RequisitionEntryID | Corresponding dictionary attribute information  |
| DM_FDR_SERVICE_n<br>(n=1, 2, 3, ...)    | RequisitionEntryID | Corresponding service form data attribute information   |

The tables which comprise the data mart have been indexed to optimize the performance of queries and reports that retrieve data from multiple query subjects. Because of the dynamic nature of the dictionary and service-based dimensions, no additional indexes have been added to these tables.

The indexes provided in the statically defined fact and dimension tables are summarized below.

**Table 11: Index in fact and Dimensions Table**

| Data Mart Table/View    | Primary Key        | Additional Indexes  |
|-------------------------|--------------------|---|
| DM_DEFSERVICE           | ServiceID          | SERVICENAME   |
| DM_DEFDICTIONARY        | DictionaryID       | DICTIONARYNAME  |
| DM_PERSON               | PersonID           | PERSONFIRSTNAMEPERSONLASTNAME<br>ISQUEUE  |
| DM_DATE                 | DateID             | (none)  |
| DM_REQUISITIONENTRYFACT | RequisitionEntryID | SERVICEID<br>REQUESTORID<br>CUSTOMERID<br>STARTEDDATE<br>CLOSEDDATE<br>DUE DATE |

| Data Mart Table/View  | Primary Key       | Additional Indexes   |
|-----------------------|-------------------|--|
| DM_SERVICETASKFACT    | ServiceTaskID     | REQUISITIONENTRYID<br>PERFORMERID<br>SERVICEID<br>STARTEDDATE<br>COMPLETEDDATE<br>DUEDATE<br>QUEUEID |
| DM_REQUISITONTASKFACT | RequisitionTaskID | PERFORMERID<br>STARTEDDATE<br>COMPLETEDDATE<br>DUEDATE<br>QUEUEID                                    |

## Relationships between the Facts and Dimensions

Requisitions (ServiceRequestFact) are joined to all relevant dimensions (shown in the star schema previously included) via “inner joins”. That means that any attempt to use query items from both the requisition and a dimension will show only those requisitions which have a corresponding row in the dimension. This is generally not a factor for all of the statically defined dimensions, since these are always required for all requisitions. For example, by definition a requisition must have a customer and initiator as well as a requested service and all dates associated with the delivery of that requisition.

This does have implications for writing reports. For example, if you start defining a report by choosing a set of customers, then add requisition data filtered for a particular period, those customers who did not order a service in that period will vanish from the report.

It is critically important for the dynamically defined, dictionary-based dimensions. If a dictionary was not used in a particular service, any requisitions for that service will not appear on a report that includes query items from that dictionary-based dimension.

Similarly, for delivery tasks (ServiceTaskFact) and service-level authorizations (RequisitionTaskFact), inner joins relate the fact to all dimension tables except the queue. These facts are joined to the Queue dimension via an “outer join”, which supports optional relationships. This allows the service designer to assign the task to a specific person or functional position, rather than to a queue. If a task was not assigned to a queue, it still appears on the report, but the queue will be blank.

For request-level authorizations (AuthTaskFact), too, the queue is optional. In addition, the service is not relevant, since the authorization is performed at the request level, rather than for any individual services which comprise the request.

## Relationships in Organizational Data

The Organizations folder allows you to compose reports on people, organizations, and groups. Service Catalog supports many-to-many relationships between these entities. For example, a person may be a member of many



organizations (a business unit and multiple service teams); an organization comprises many people. These relationships are reflected in the data mart design, so you can combine two of these entities on a report and group by either entity. For example, you could report on all organizations, listing their members; or you could choose a person, and list all the organizations to which that person belongs.

## Fact to Fact Relationships

**Table 12: Fact to Fact Relationships Table**

| Fact-1             | Fact-2              | Type of Relationship |
|--------------------|---------------------|----------------------|
| ServiceRequestFact | ServiceTaskFact     | Left Outer Join      |
| ServiceRequestFact | DeliveryTaskFact    | Inner Join           |
| ServiceRequestFact | AuthTaskFact        | Left Outer Join      |
| ServiceRequestFact | AllTaskFact         | Left Outer Join      |
| AllTaskFact        | TaskEffortEntryFact | Left Outer Join      |
| DeliveryTaskFact   | TaskEffortEntryFact | Left Outer Join      |
| ServiceTaskFact    | TaskEffortEntryFact | Left Outer Join      |

