THE REVERSE STAR SCHEMA

Use of a central dimensional table to facilitate 'one change' row
level security in Cognos Framework Manager

Document History

Date	Version	Summary of Changes
10/Feb/2007	Draft v1.0	Draft
26/Feb/2007	WIP v1.1	Rework following review
12/Mar/2007	WIP v2.0	Rewordings
30/Apr/2007	Final v2.1	Approval by P.R. for public release

Distribution
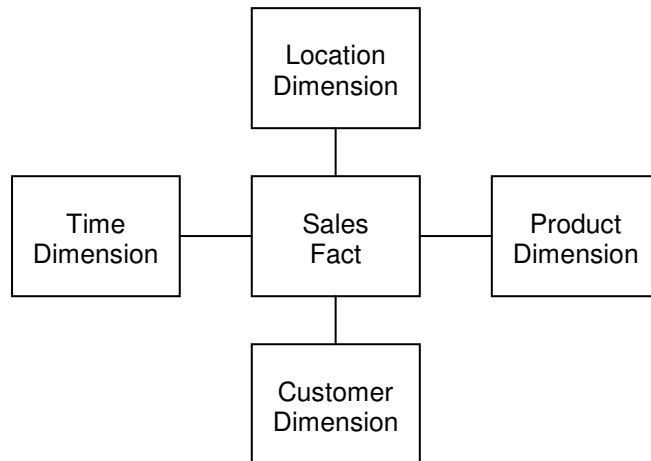
Name	Date of Issue	Version

1   THE REVERSE STAR SCHEMA

The standard star schema (Figure1) as described by Ralph Kimball has one fact table and is surrounded by a number of dimensions. These dimensions have normally been created by 'denormalising' (squashing a number of smaller tables containing descriptive data into one big table).
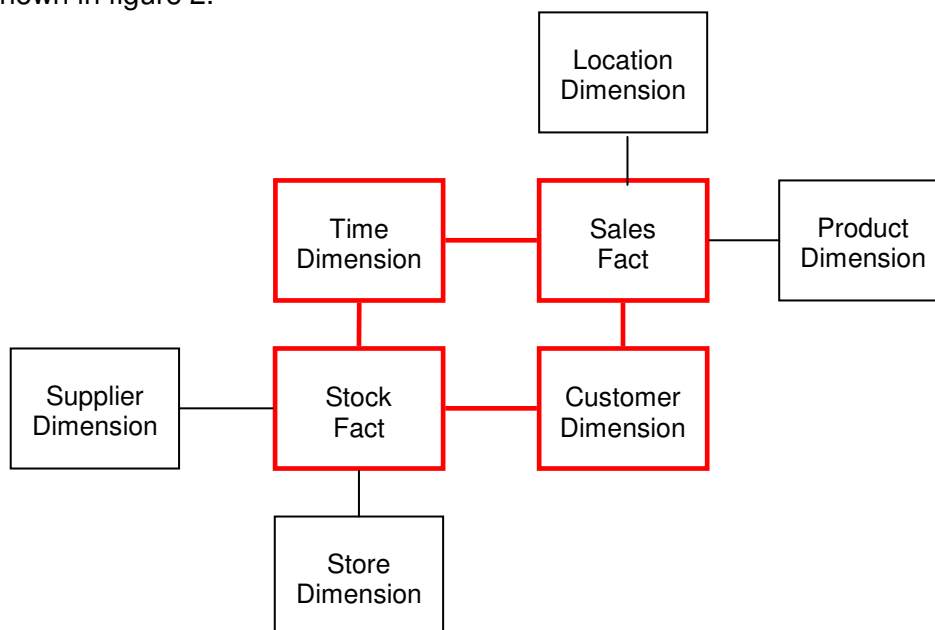
Figure 1:

```
                      ┌──────────────┐
                      │   Location   │
                      │  Dimension   │
                      └──────┬───────┘
                             │
┌──────────────┐      ┌──────┴───────┐      ┌──────────────┐
│     Time     │      │    Sales     │      │   Product    │
│  Dimension   │──────│    Fact      │──────│  Dimension   │
└──────────────┘      └──────┬───────┘      └──────────────┘
                             │
                      ┌──────┴───────┐
                      │   Customer   │
                      │  Dimension   │
                      └──────────────┘
```

This design has proved to be invaluable to those working in reporting, BI and MIS. The design means that we can make selections of millions of rows from the central fact table, pull in some attributes from a dimension table and still only use one join (meaning a fast query result time)

This is great, but most businesses have more than one focus of measurement (fact table). When multiple fact tables are in a data warehouse life becomes more complex. The reason for this is that we want to use all of our fact tables, and we also want to use common dimensions (such as product or customer). By joining fact tables to shared dimension tables can create loops as shown in figure 2.

Figure 2:

```
                                      ┌──────────────┐
                                      │   Location   │
                                      │  Dimension   │
                                      └──────┬───────┘
                                             │
                ┌──────────────┐      ┌──────┴───────┐      ┌──────────────┐
                │     Time     │      │    Sales     │      │   Product    │
                │  Dimension   │──────│    Fact      │      │  Dimension   │
                └──────┬───────┘      └──────┬───────┘      └──────────────┘
                       │                     │
┌──────────────┐      ┌┴─────────────┐      ┌┴─────────────┐
│   Supplier   │      │    Stock     │      │   Customer   │
│  Dimension   │──────│    Fact      │──────│  Dimension   │
└──────────────┘      └──────┬───────┘      └──────────────┘
                             │
                      ┌──────┴───────┐
                      │    Store     │
                      │  Dimension   │
                      └──────────────┘
```

This can easily be resolved by using alias tables or views and effectively separates the fact tables from each other (figure 3). It should be noted that it is perfectly valid for two fact tables to be linked by a single dimension table.

Figure 3:
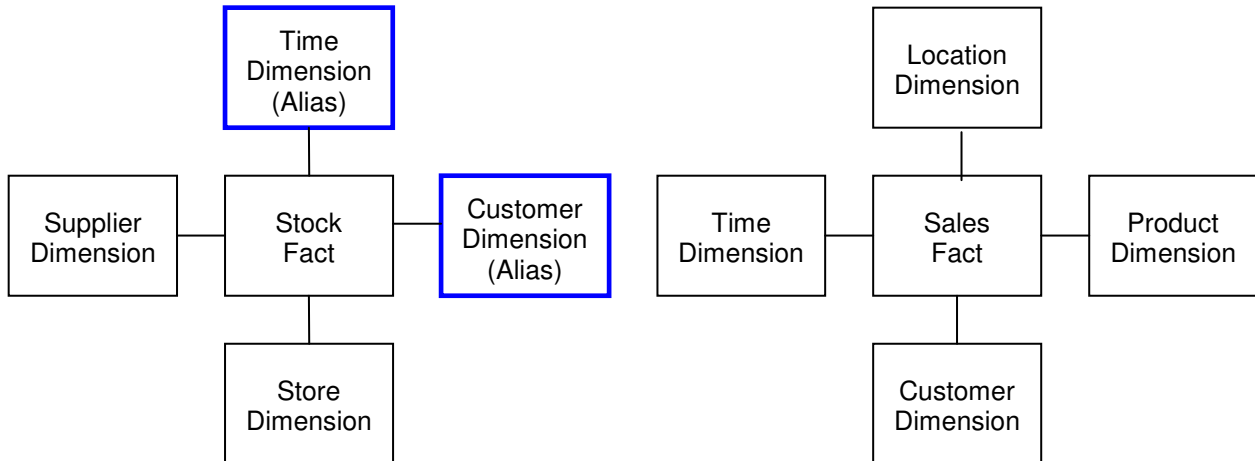
In order to report from these structures using Report Studio they must first be modelled in Framework Manager.

It is at this point where row level security can be implemented. Row level security allows the administrator to set up security against a query object (in our case a table) so that users within a specific user class are restricted to only access some of the information within the table.

Therefore if the administrator only wanted users within the French Marketing team to only see data referring to French customers, he could set the security so that the French Marketing team user class is limited to where Country (in the customers table) = France. By doing this the administrator can easily control access to the data by granting or denying users access to specific user classes.

Figure 4:

CustomerID	CustomerName	Contact	Country	Cust Type
1	Rawlings	Bill Withers	UK	Pharma
2	BTFA	Gregoire Beni	France	Minerals
3	Gutteridge	Francesco Tutti	Italy	Pharma
4	Apex	Fabienne Pelot	France	Transport

It is highly likely that there are many user classes that require this form of security building for the customers table and that there are many fact tables that these users wish to access. The solution is to use a logical design we refer to as the reverse star schema (Figure5).

Figure 5:
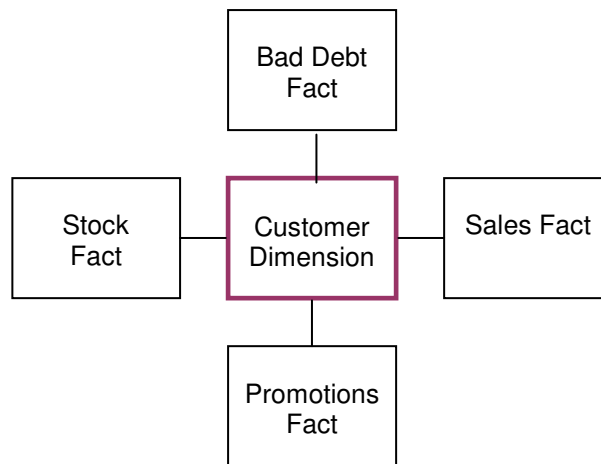
Using this design the row level security is set up in on one dimension that is common to all fact tables. It saves the administrator the time and expense of having to set up security separately for each fact table.

Of course other dimension tables are likely to be used in the design. Figure 6 shows a more realistic example of how the schema would appear:

Figure 6:
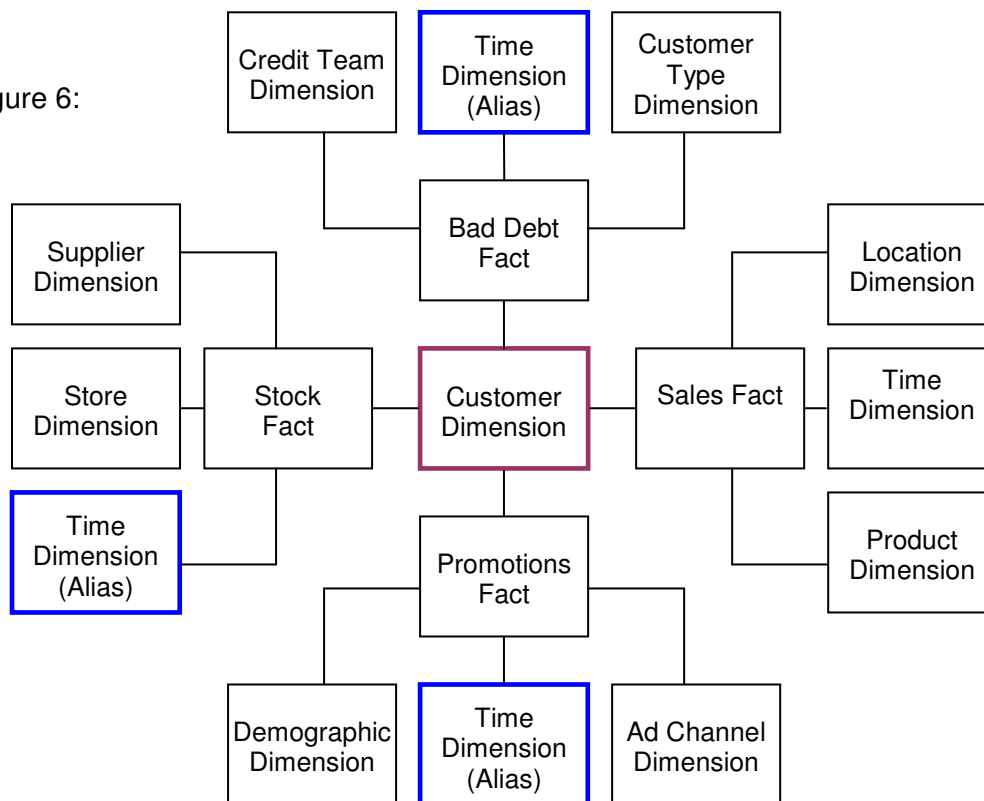
2 <u>SUMMARY</u>

This white paper demonstrates the modelling technique of placing a dimension table at the centre of a schema and surrounding it with fact tables thereby creating a 'reverse star schema'. As the document discusses the main driver for this is to facilitate 'one touch' security administration when using row level security.

3   COPYRIGHT AND USE

Please feel free to use the ideas and methods discussed in this white paper. All we ask in return that you add a link to www.digitalviper.co.uk

The 4 bars image is a trademark of Digital Viper and may not be used without express written permission.