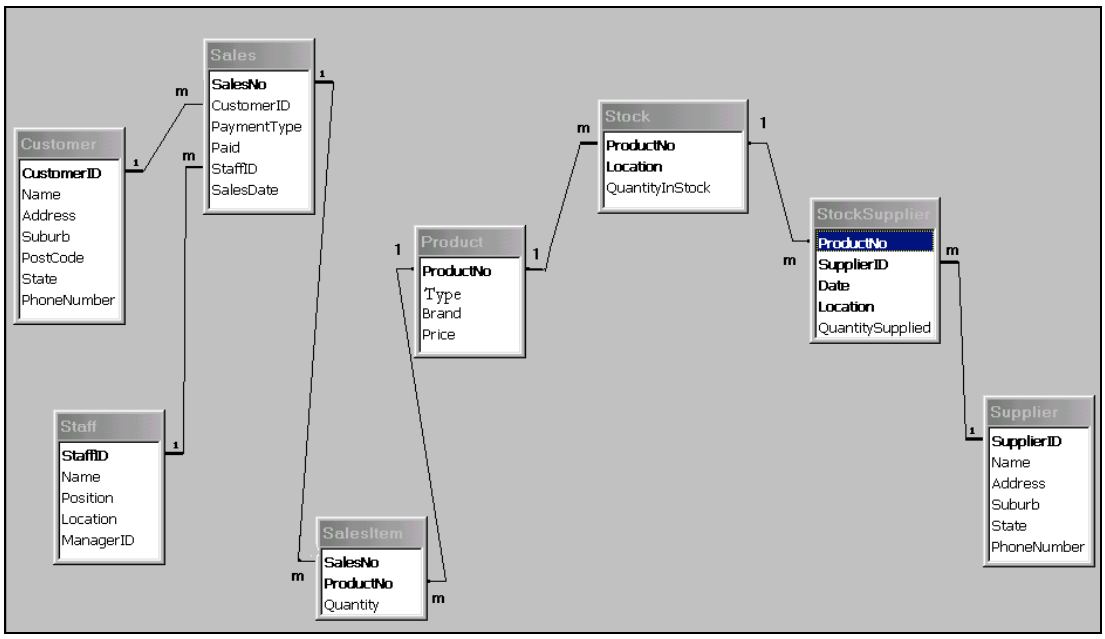
Bridge Tables
Case Study #4 (PRODUCT SALES SYSTEM)

1. Case Study Description

The following is an E/R diagram of a product sales system.



The management of the company would like to analyse the statistics of its product sales history. The analysis is needed to identify popular products, suppliers supplying those products, the best time to purchase more stock, etc. Hence, a small data warehouse is to be built to keep track of the statistics. The management is particularly interested in analysing the *total sales* (Quantity * Price) by *product*, *customer locations (suburbs and postcodes)*, *sales time periods* (monthly and yearly), and *supplier*.

2. The Solutions

We start the data warehouse by designing 2-column tables. Here are some possible scenarios (note that the total sales figures are imaginary):

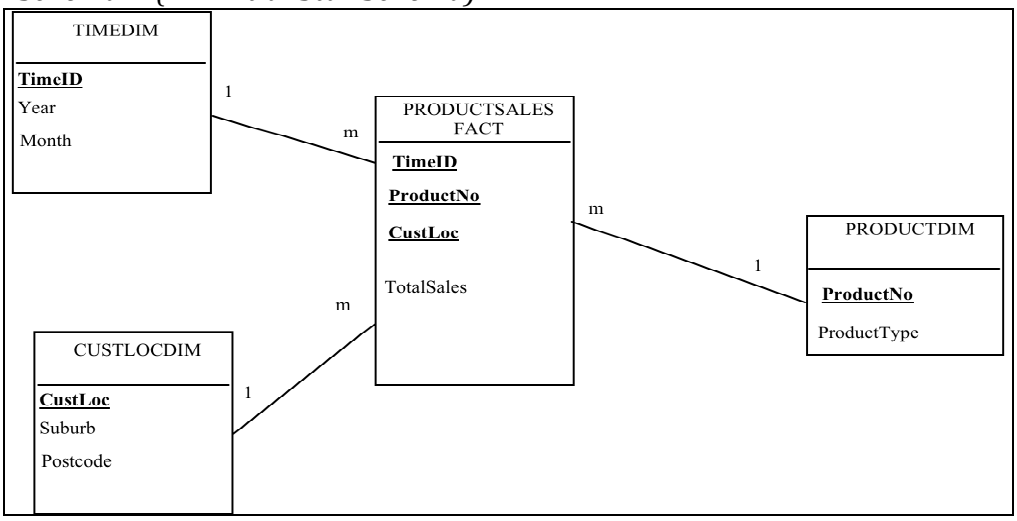
ProductNo	TotalSales
A3	$130,000
B2	$15,900
C7	$2,500,000
...	...

TimeID	TotalSales
201001	$25,000
201002	$4,700
201003	$3,500
...	...

CustLoc	TotalSales
Caulfield	$6,500
Chadstone	$12,000
Clayton	$1,800
…	…

The star schema is shown in Star Schema-1 below:

Star Schema-1 (An Initial Star Schema):



One possible report from the above fact table is:

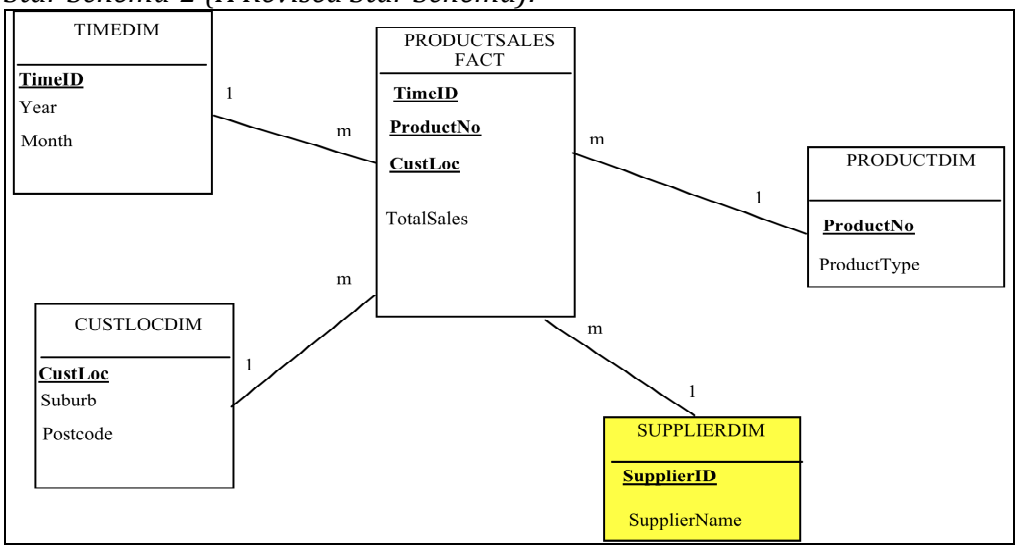
Report 1:

TimeID	CustLoc	ProductNo	TotalSales
201001	Caulfield	A3	$450
201001	Caulfield	B2	$100
201001	Caulfield	C7	$320
201001	Caulfield	…	
201001	Chadstone	A3	$75
201001	Chadstone	B2	$600
201001	Chadstone	C7	$55
201001	Chadstone	…	
201001	Clayton	A3	$130
…	…	…	
201003	Caulfield	A3	$500
201003	Caulfield	B2	$430
201003	Caulfield	C7	$120
…	…	…	

However, the management would also like information about the supplier, which is still missing from the above report. Therefore, one could have another 2-column table for suppliers, which may look like the following, and the star schema is shown in Star Schema-2:

SupplierID	TotalSales
S4	$77,000
S5	$5,700
S6	$12,500
…	…

Star Schema-2 (A Revised Star Schema):



And therefore, Report 1 is modified to include information about the supplier (see Report 2 below):

Report 2:

TimeID	CustLoc	ProductNo	SupplierID	TotalSales
201001	Caulfield	A3	S4	$450
201001	Caulfield	A3	S5	
201001	Caulfield	A3	S6	
201001	Caulfield	A3	…	
201001	Caulfield	B2	S4	$100
201001	Caulfield	B2	S5	
201001	Caulfield	B2	S6	
201001	Caulfield	B2	…	
201001	Caulfield	C7	S4	$320
201001	Caulfield	C7	S5	
201001	Caulfield	C7	S6	
201001	Caulfield	C7	…	
201001	Caulfield	…		
201001	Chadstone	A3		$75
201001	Chadstone	B2		$600
201001	Chadstone	C7		$55
201001	Chadstone	…		
201001	Clayton	A3		$130
…	…	…		
201003	Caulfield	A3		$500
201003	Caulfield	B2		$430
201003	Caulfield	C7		$120
…	…	…		

Report 2 seems to be accurate. Actually, it is impossible to produce such a report, because in the E/R diagram (in the operational database), because one supplier supplies many products (or one product is supplied by many suppliers), the sales do not record from which supplier a product is sold. In other words, there is no direct relationship between suppliers and total sales. Therefore, the 2-column table for the supplier, although seems to be correct, is actually **wrong**.

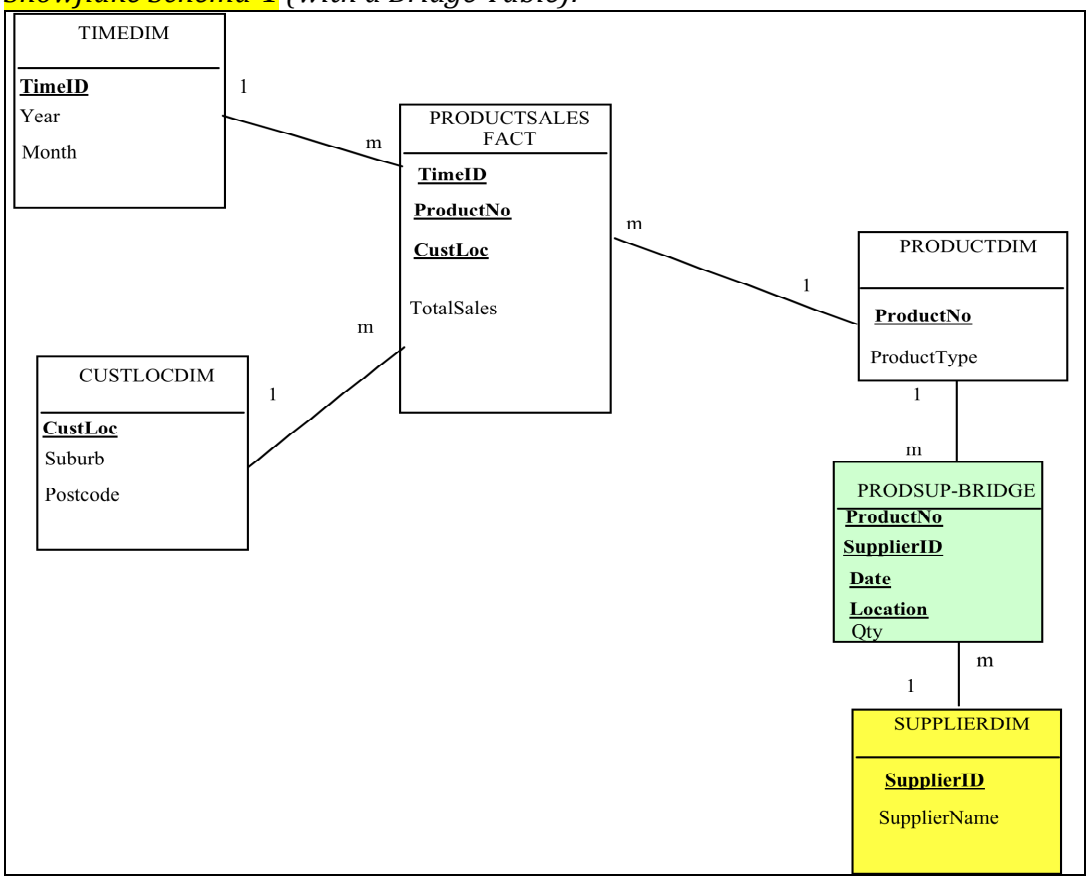
The problem will even be bigger, if we add the date of the supplies (e.g. Date attribute from the StockSupplier table – see the E/R diagram). Imagine that in Report 2, you have one additional column next to the SupplierID, called SupplyDate. This will create a conflict between the TimeID column (1st column in Report 2), and the SupplyDate column. On top of this, it is also impossible to have a SupplyDate column in Report 2, because one product-supplier pair may have multiple supply dates.

The reasons for this problem to occur are: (1) There is no direct relationship between supplier and product sales, and (2) the supplier information is not available in the sales of a particular product.

One solution is to move the Supplier Dimension from being attached to the Fact (see Star Schema-2), and to create a relationship with the Product Dimension – resulting a Bridge Table between Product Dimension and Supplier Dimension (see Snowflake Schema-1 below). Note that three additional pieces of information are added: location, date, and quantity of the supplies.

The aim for using a bridge table is for "*drilling down*" information on products. For example, the management, after examining the fact report, might be interested in drilling down further on a particular product. Perhaps the product performs well in the sales, or for any other reasons. Hence, we can drill down that product and find out, in this example, the details of the supplies history and the suppliers as well.

Snowflake Schema-1 (with a Bridge Table):



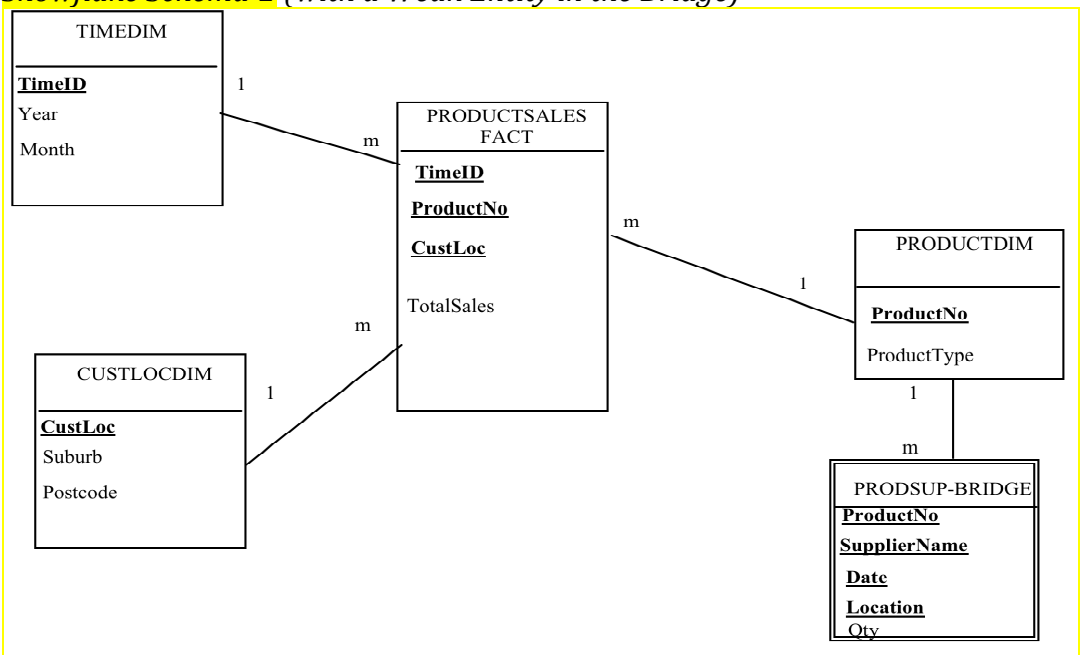
The SQL commands to create fact and dimension tables are as follows:

```
create table TimeDim as
select
   distinct to_char(salesDate, 'YYYYMM') as TimeID,
   to_char(salesDate, 'YYYY') as Year,
   to_char(salesDate, 'MM') as Month
from Sales;

create table CustLocDim as
select distinct Suburb||Postcode as LocID, Suburb, Postcode
from Customer;

create table ProductDim as
select distinct ProductNo,type
from Product;


create table ProductSupplierBridge as
select *
from stockSupplier;

create table SupplierDim as
Select supplierID, Name as Suppliername
from supplier;
```

```
Create table ProductSalesFact as
Select
   to_char(s.salesDate, 'YYYYMM') as TimeID,
   p.productNo as ProductNo,
   c.Suburb||c.Postcode as LocID,
   sum(si.quantity*p.price) as TotalSales
from sales s, product p, customer c, salesitem si
where s.salesNo = si.salesNo
and si.ProductNo= p.ProductNo
and c.CustomerID = s.CustomerID
Group By to_char(s.salesDate, 'YYYYMM'), p.productNo,
c.Suburb||c.Postcode;
```

If, for example, the operational database does not maintain a master list of all suppliers (assume that each supplier may supplies any product in an ad-hoc manner), then we will not have a Supplier Dimension in the Snowflake Schema-1. Instead, we will only have a Bridge Table, but for each Product-Supplier pair, we will have Location and Date implemented as weak entity, because for each product-supplier, there are multiple supplies. Therefore, the Bridge Table acts as a ***Temporal Dimension***. That means, for each product, there is a list of history of supplies.

Snowflake Schema-2 (with a Weak Entity in the Bridge)



The Product Dimension table and the Bridge Table may look something like this:

ProductDim Table:

ProductNo	ProductType
A3	Handbag
B2	Women Shoes
C7	T-Shirt
…	…

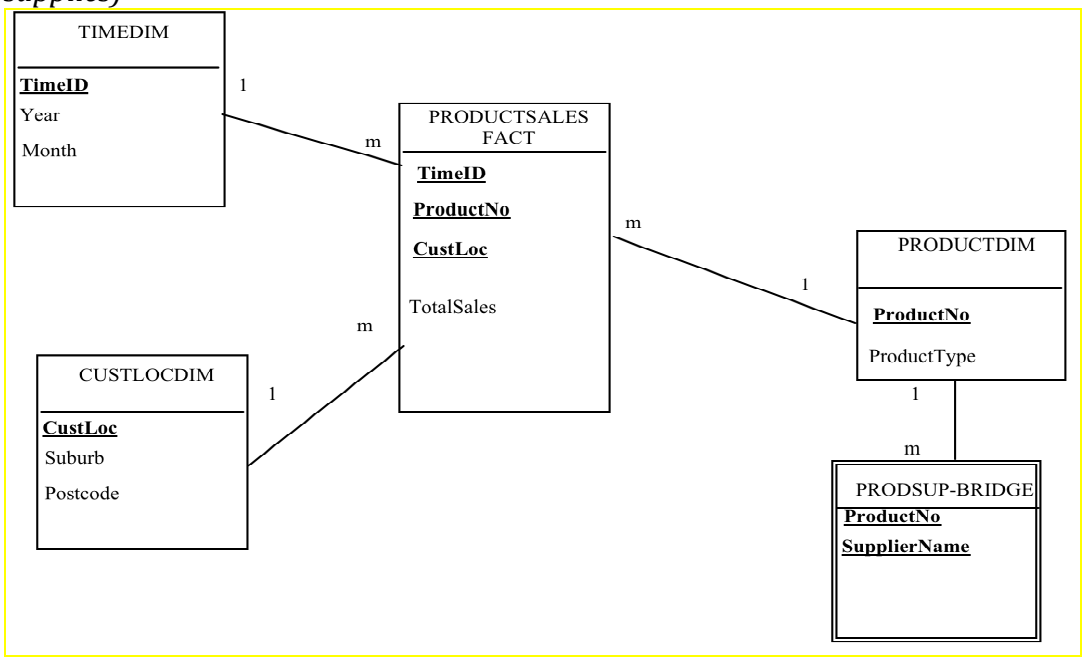
ProdSup-Bridge Table:

ProductNo	SupplierName	Date	Location	Qty
A3	CheapGoods	5-May-2010	Caulfield	100
A3	CheapGoods	15-Aug-2010	Caulfield	150
A3	CheapGoods	1-Dec-2010	Clayton	50
A3	CheapGoods	…	…	…
A3	JustBags	5-May-2010	Chadstone	200
A3	JustBags	30-June-2010	Clayton	80
A3	JustBags	…	…	…
A3	Baggy			
A3	…			
B2	CheapGoods			
…	…			

Hence, the *ProdSup-Bridge Table* acts as a ***Temporal Dimension***, since the history of supplies are recorded.

If the history is not needed (in a non-temporal data warehouse), then the ProdSup-Bridge Table will only have ProductNo and SupplierName attributes, without the history of supplies (see Snowflake Schema-3 below):

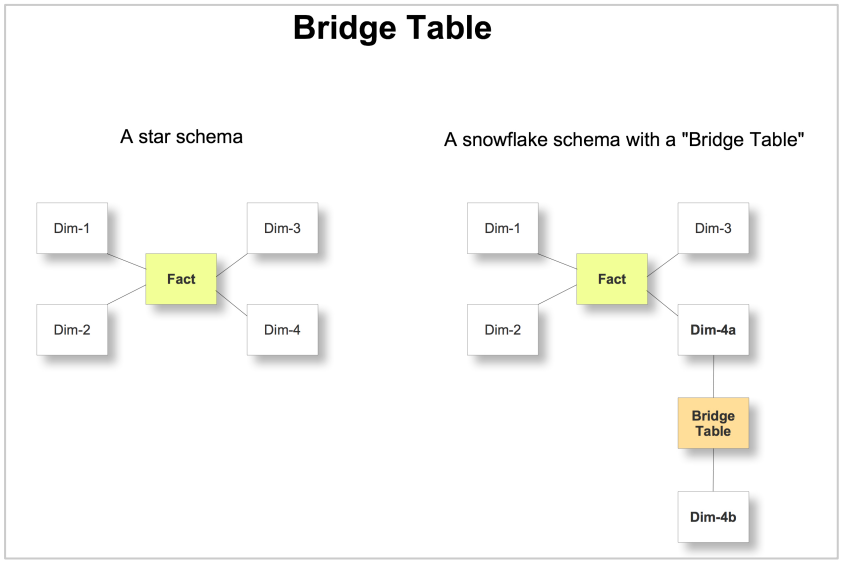
Snowflake Schema-3 *(Revised Snowflake Schema without maintaining the history of supplies)*



3. Bridge Table

A bridge table is a table that links between two dimensions; and only one of these two dimensions are linked to the fact. As a result, the star schema becomes a snowflake schema.

The following example compares a normal star schema with a snowflake schema; the latter has a bridge table. In the snowflake example below, Dim-4b is not linked directly to the fact, but is linked to another dimension (e.g. Dim-4a) through a Bridge Table.



Why a Bridge Table is needed?
It is needed because the dimension cannot be linked directly to the fact; it has to go through another dimension. For example, Dim-4b cannot be linked directly to the fact without going through Dim-4a.

Why can't a dimension be linked directly to the fact?
There are at least two reasons:

(i) The fact table has a measurable fact figure, and the dimension has a key identity. In order to link a dimension to the fact, the dimension's key identity must contribute directly to the calculation of the measurable fact. Unfortunately, this cannot happen if the data in the operational database does not have this data.

(ii) The operational database does not have this data if the relationship between two entities in the operational database that hold the information about dimension's key identity and the intended measurable fact is a *many-to-many* relationship.

4. Summary

In summary:

- Snowflake schema-1 contains a simple (and complete) bridge table
- Snowflake schema-2 contains a temporality feature (history of supplies; quantity and date of each supply)
- Snowflake schema-3 does not maintain any history of supplies (e.g. no temporality aspect)

In principal, a Bridge Table is used:

1. When it is impossible to have a dimension connected directed to the Fact table, because simply there is no relationship between this dimension and the Fact table (e.g. in star schema 2, it is impossible to have a direct link from SupplierDIM to ProductSalesFact)

2. When an entity (which will become a dimension) has a many-to-many relationship with another entity (dimension) in the E/R schema of the operational database (e.g. Supplier and Stock has a many-to-many relationship).

3. When temporality aspect (data history) is maintained in the operational database and the bridge table can be used to accommodate the dimension that has temporal attributes (e.g. product supply history is maintained in Snowflake Schema-2).