www.sheepsqueezers.com

Star Schema Space Reduction Ideas

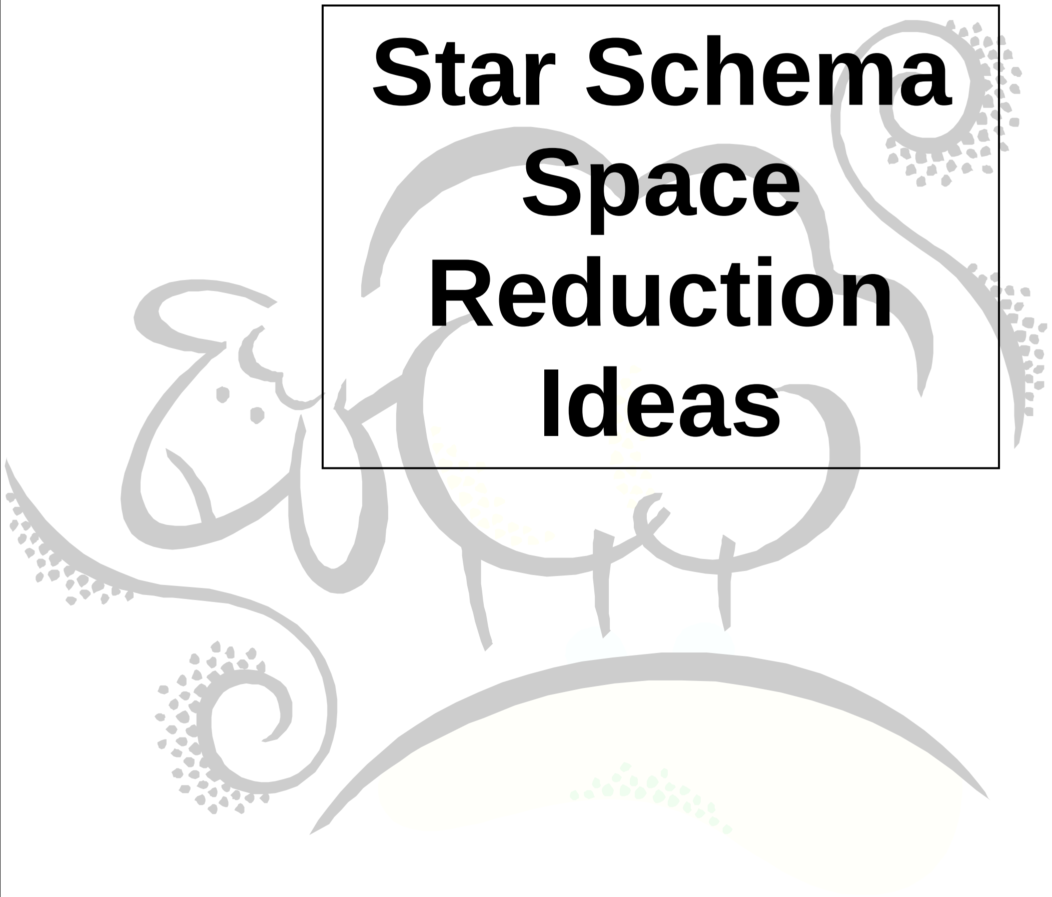
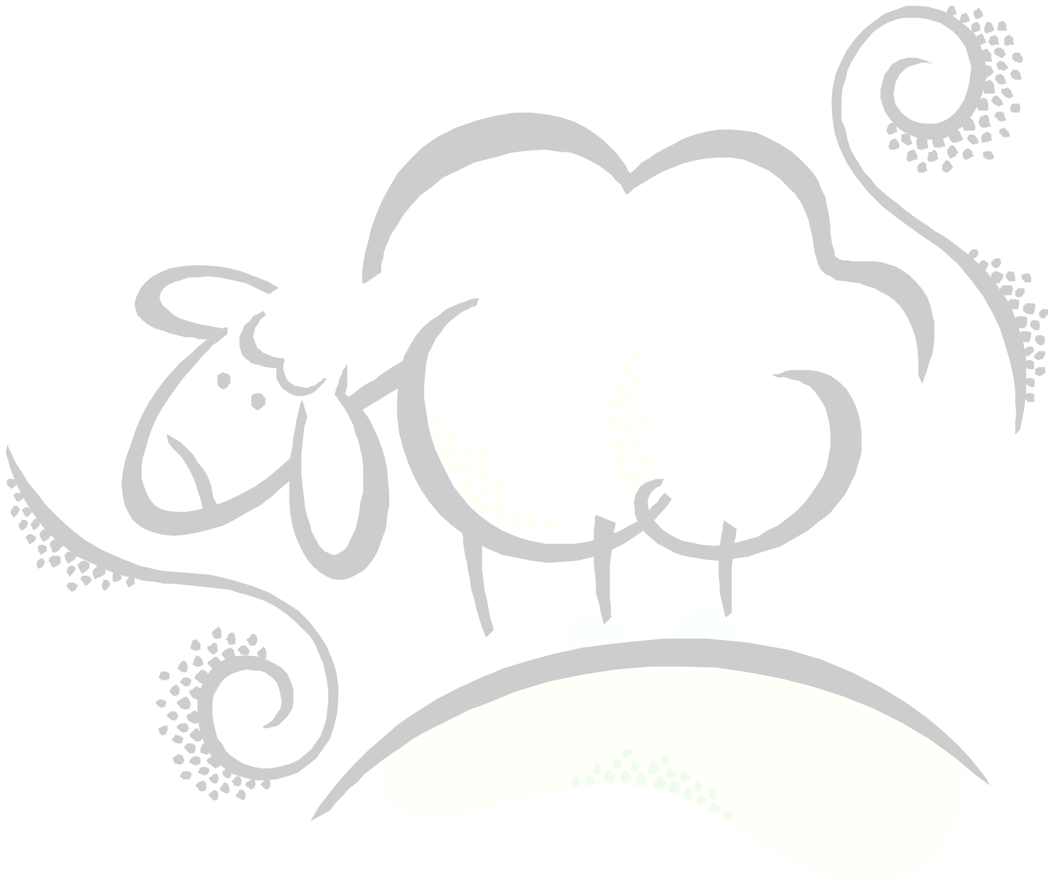
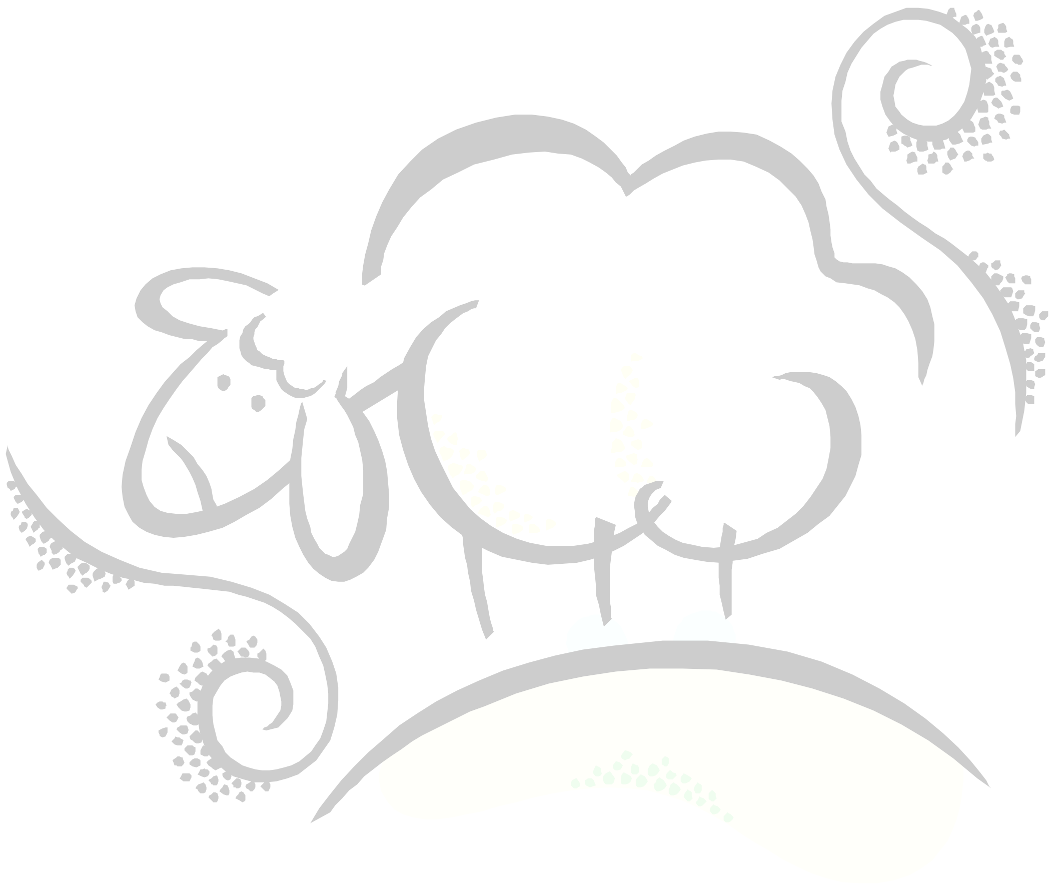


Table of Contents



INTRODUCTION

This document outlines several methods to reduce the number of columns and/or rows in very large tables when the organized in a star schema format. **Note that these ideas have not been tested on real data and are only conceptual at this point and may not even work. Please let us know if you've had any luck with these ideas by emailing us at comments@sheepsqueezers.com. Thanks!**



METHOD #1 – Reducing the Number of Columns in the FACT Table

Let's assume that we have *d* dimension tables

$$d_1,\ d_2,\ \ldots,\ d_d$$

each with a single numeric primary key and at least one attribute column. Without loss of generality, assume that each primary key is numbered starting at 1 and increases by 1 for each row in the dimension table. Assume that each dimension table d_i contains c_i distinct values for each primary key. For example, the GENDER_DIM (d_1) table would contains 3 (c_1) distinct values (MALE, FEMALE and UNKNOWN).

Also, assume that the FACT table contains *d* foreign keys

$$f_1,\ f_2,\ \ldots,\ f_d$$

one associated with each dimension table's primary key as well as *v* additional (value) columns

$$v_1,\ v_2,\ \ldots,\ v_v.$$

Now, if our FACT table has a significant number of rows, it would behoove us to remove as many ***columns*** from our FACT table as possible to keep the table size on disk as small as possible. One way to do this is to remove the *d* foreign keys $f_1, f_2, \ldots, f_d$ from the FACT table and replace them with one column *r* which contains the number 1 for the first row in the FACT table, 2 for the second, 3 for the third, etc. Essentially, *r* is a row number across the FACT table.

Once you replace $f_1, f_2, \ldots, f_d$ with *r*, you will need to be able to compute the row number based on the selection of primary key values in the *d* dimension tables. Note that these *d* dimension tables are equivalent to *d* drop-down boxes on, say, an Excel spreadsheet. To get the row number from the primary key values selected in the d dimension table, employ this formula:

$$r = (c_2 * \ldots * c_{d-2} * c_{d-1} * c_d) * (u_1 - 1) + \ \ldots + (c_{d-2} * c_{d-1} * c_d) * (u_{d-3} - 1) + (c_{d-1} * c_d) * (u_{d-2} - 1) + (c_d) * (u_{d-1} - 1) + 1 * (u_d - 1) + 1$$

where u_i is the primary key value for the selected option for the dimension table d_i.

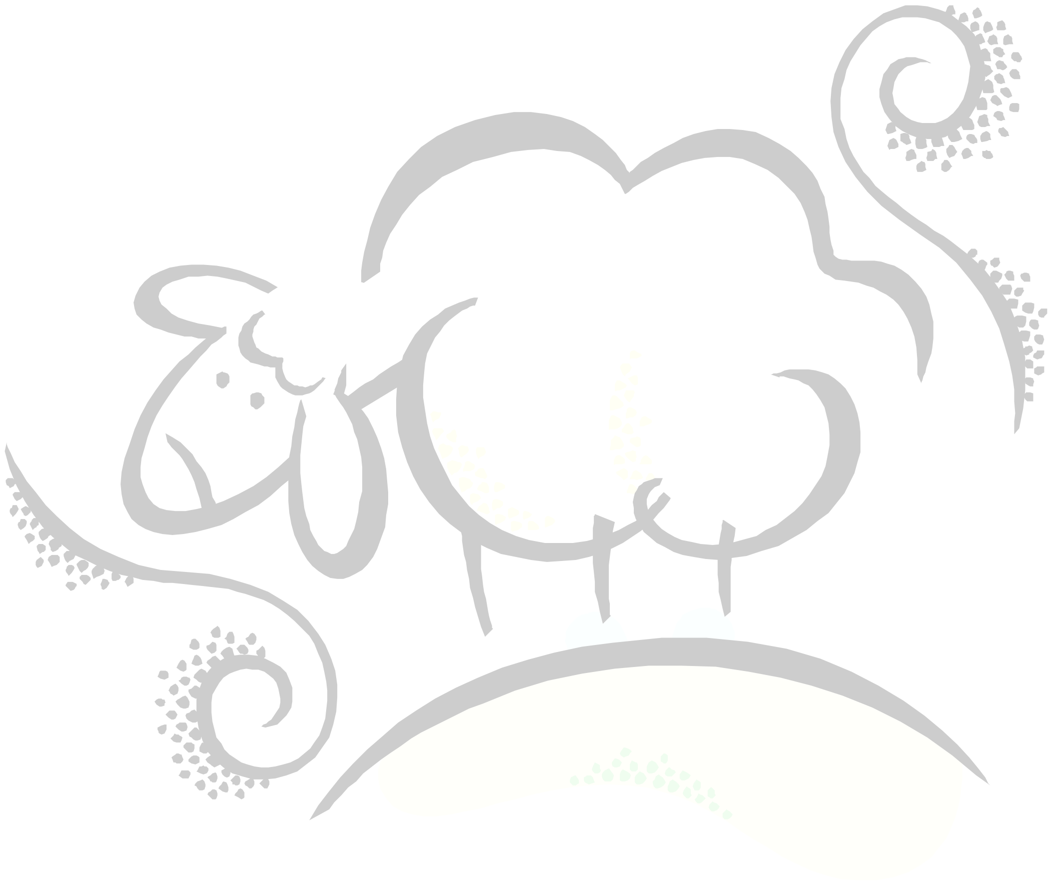
For example, given three dimension tables:

```
GENDER_DIM (with 3 values MALE/FEMALE/UNKNOWN)
AGE_DIM (with 2 values YOUNG/OLD)
PAYOR_DIM (with 4 values COMMERCIAL/MEDICARE/MEDCAID/OTHER)
```

The grand total number of distinct choices from these three dimension tables is 3*2*4 = 24. This would be the maximum number of rows in the FACT table. In order to compute the row number from the selected values, use this formula:

```
r = (2*4) * (GENDER_DIM-1) + (4) * (AGE_DIM-1) + (1) * (PAYOR_DIM-1) + 1
  = 8*(GENDER_DIM-1) + 4*(AGE_DIM-1) + (PAYOR_DIM-1) + 1
```

When GENDER_DIM=1, AGE_DIM=1 and PAYOR_DIM=1, then r=8*(0) + 4*(0) + 1*(0) + 1 = 1 and
when GENDER_DIM=3, AGE_DIM=2 and PAYOR_DIM=4, then r=8*2+4*1+1*3+1 = 16+4+3+1=24 which is the last row.



METHOD #2 – Reducing the Number of Rows in the FACT Table

As above, let's assume that we have *d* dimension tables

$$d_1,\ d_2,\ \ldots,\ d_d$$

each with a single numeric primary key and at least one attribute column. Without loss of generality, assume that each primary key is numbered starting at 1 and increases by 1 for each row in the dimension table. Assume that each dimension table d_i contains c_i distinct values for each primary key. For example, the GENDER_DIM (d_1) table would contains 3 (c_1) distinct values (MALE, FEMALE and UNKNOWN).

Also, assume that the FACT table contains *d* foreign keys

$$f_1,\ f_2,\ \ldots,\ f_d$$

one associated with each dimension table's primary key as well as *v* additional (value) columns

$$v_1,\ v_2,\ \ldots,\ v_v.$$

Now, if our FACT table has a significant number of rows, it would behoove us to remove as many ***rows*** (!) from our FACT table as possible to keep the table size on disk as small as possible. Unfortunately, our *d* foreign keys makes removing rows from the table impossible as the table stands now because we would be losing information. The goal in this section is to remove rows *without* losing information.

One approach to remove rows from the FACT table without losing information is to create another table containing the distinct rows from the FACT table but only on the columns v_1, v_2, ..., v_v ignoring the foreign keys. Note that if one or more of the columns v_1, v_2, ..., v_v contain fractional amounts, there won't be a significant reduction in rows, so the programmer should attempt to round any columns containing fractional amounts to 1 or two decimal places (zero, if at all possible). Along with the distinct rows of v_1, v_2, ..., v_v the programmer should also create a row number column *r*, thus the FACT table will be boiled down to distinct rows of v_1, v_2, ..., v_v along with a row number column *r*. Let's call this reduced table FACT_LITE.

At this point, the programmer has to create *another* table containing just the foreign keys from the FACT table (really, these are the primary keys from the dimension tables) along with the column *r* from the table we created above. This would require a merge back to the FACT table to attach the column r to it in order to create this table. Let's call this mapping table DIM_FACT_MAP.

Thus, based on the user's choices for each dimension table, you will have a very long, skinny table (DIM_FACT_MAP) mapping from the primary keys to row number r and then you can look up the appropriate row in the FACT_LITE table.

Unfortunately, DIM_FACT_MAP may contain many rows and searching through it may take a while even with indexes. One solution to this is to partition this table (in the Oracle sense) or in the naive sense; that is, by breaking this table into several tables with fewer rows.

METHOD #3 – Reducing the Number of Rows in the FACT Table (Redux)

Referring to METHOD #2 above, the reduction of the number of rows in a FACT table, we saw that the final result was a smaller FACT table called FACT_LITE, but a very large table, DIM_FACT_MAP, mapping the primary keys of the dimension tables to the row number in the FACT_LITE. This makes us sad since, if we don't want a huge FACT table, why would we want a huge DIM_FACT_MAP table?

Recall that DIM_FACT_MAP is made up of the primary keys in the dimension tables along with the row number r mapped to FACT_LITE:

$$f_1,\ f_2,\ \ldots,\ f_d\ ,r_{FL}$$

One **imperfect** method to remove/reduce the DIM_FACT_MAP table is to use statistical regression. Recall that DIM_FACT_MAP maps all of the primary keys in the dimension tables to the row in the reduced FACT_LITE table. In a similar idea to that of METHOD #1, create an additional column in DIM_FACT_MAP using the formula for "r" on Page 4. Then, use regression setting the column r_{FL} as the dependent variable, and r as the independent variable:

$$r_{FL} = b_0 + b_1 r$$

When the regression is completed, you should look at the Multiple R statistic. In the unlikely eventuality that it is exactly 1, you are finished. You now have a formula mapping the selected primary key values in the dimension tables directly to the row number in the FACT_LITE table.

But, your Multiple R will most likely be less than 1. You can fiddle with the regression like this:

$$r_{FL} = b_0 + b_1 r + b_2 r^2 + + b_3 r^3 + b_4 r^4 + \ldots$$

and so on. This may get you much closer to a Multiple R of 1. But...probably not. When you've had enough and life is slow and oh so mellow, compare the actual r_{FL} to the computed r_{FL} from the formula above. (Note that you will have to remove the fractional part of the *computed* r_{FL}.) Determine which rows do not match r_{FL}; hopefully, there will be very few of these. Place these "bad boys" into a lookup table, called RR_LOOKUP. At this point, you have a formula which works most of the time and a small lookup table called RR_LOOKUP.

This is how it goes down, bro': After the user chooses his or her options in the drop-down boxes on the view, first lookup the row in lookup table and, if found, get the row number into FACT_LITE; if not found, use the formula to get the row number to FACT_LITE.

Support sheepsqueezers.com

If you found this information helpful, please consider supporting sheepsqueezers.com. There are several ways to support our site:

- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: Buy Me A Cup Of Coffee?.
- ☐ Visit my Amazon.com Wish list at the following link and purchase an item: http://amzn.com/w/3OBK1K4EIWIR6

Please let me know if this document was useful by e-mailing me at `comments@sheepsqueezers.com`.