AIS Star Schema Presentation

- **Leverage the** Star Schema **for Star Query Transformation & Optimization**
- **Presented to the New York Oracle Users Group Data Warehouse Special Interest Group by;**

  ***Mike Richards,* Advanced Information Systems**
  - **For further information contact: mrichards@advanis.com**

Overview

- **Utilize the Star Schema to leverage 'Star Query Transformation'**

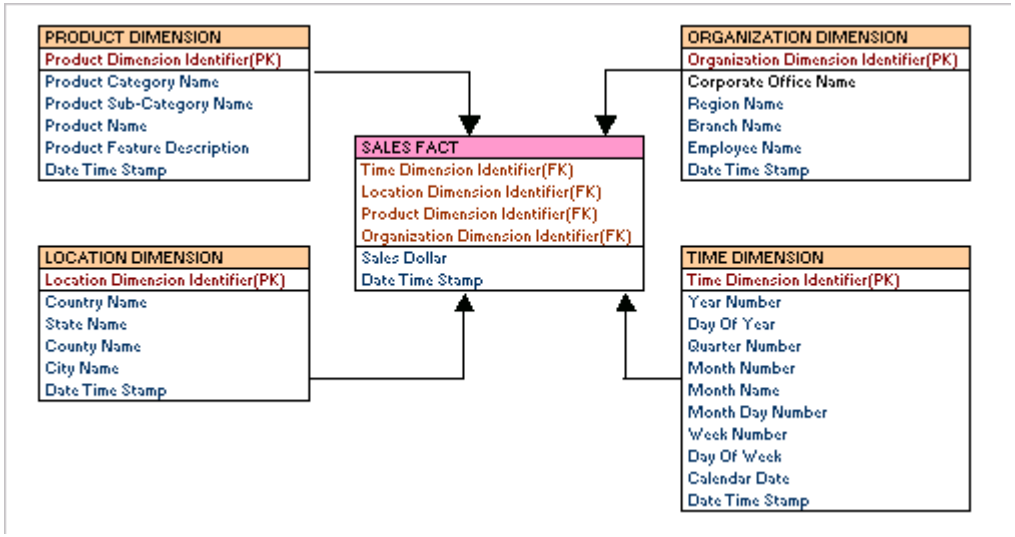
- **Oracle optimizer prunes a query's results set with it's conversion of many logical joins into a single operation with Bitmap Indexes**
- **Bitmap Indexes are up to 100 times smaller in size and hence up to 100 times faster**
- **Bitmaps are not just for low cardinality**

- **Central table in a star schema is called a FACT table**
- **A fact table typically has two types of columns: those that contain facts and those that are foreign keys to dimension tables**
- **In a star schema every dimension will have a primary key**
- **In a star schema, a dimension table will not have any parent table**

- **Whereas in a snow flake schema, a dimension table will have one or more parent tables**

Logical Star Schema Model

- **In the example figure 1.6, sales fact table is connected to dimensions; Product, Location, Organization and Time**



Star Query

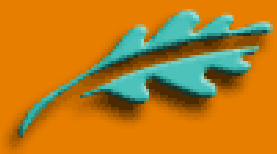
- **A Star query is a join between a fact table and a number of dimension tables.**
- **Each dimension table is joined to the fact table using a primary key to foreign key join**
- **The dimension tables are not joined to each other**
- **The Oracle 10*g* cost-based optimizer recognizes star queries and generates the most efficient execution plans for them**

Star Transformation

- **For *star_transformation* join plans, the following parameters must also be considered:**

- ***star_transformation_enabled*= TRUE**
- **No hint STAR: So forcing a *star_query* excludes *star_transformation***
- **No BIND VARIABLE in SELECT statement**
- **No CONNECT BY and *start with***
- **Fact table columns in EQUIJOIN predicate, must have bitmap index defined on them**
- **More than 2 bitmap indexes on fact table**
- **Fact table must have more than 15,000 rows**
- **Fact table cannot be a view or a remote table**
- **No hint FULL on fact table**

Successful *star_transformation*



- **A successful *star_transformation* join execution (explain) plan looks like:**
- **0 SELECT STATEMENT Optimizer=CHOOSE**
- **1 0 NESTED LOOPS**
- **2 1 HASH JOIN**
- **3 2 HASH JOIN**
- **4 2 TABLE ACCESS (FULL) OF 'MINUTE_DIMENSION'**
- **5 2 PARTITION CONCATENATED**
- **6 2 TABLE ACCESS BY ROWID**
- **7 2 BITMAP CONVERSION TO ROWIDS**
- **8 2 BITMAP AND**
- **9 2 BITMAP MERGE**
- **10 2 BITMAP KEY ITERATION**
- **11 2 SORT BUFFER**
- **12 2 TABLE ACCESS (FULL) OF 'MINUTE_DIMENSION'**

Star_transformation Execution Plan

- **The Star Transformation Execution Plan looks like:**

- SELECT STATEMENT C=301
- NESTED LOOPS
- HASH JOIN
- HASH JOIN
- TABLE ACCESS ... D1
- PARTITION CONCATENATED
- TABLE ACCESS BY ROWID F
- BITMAP CONVERSION TO ROWIDS
- BITMAP AND
- BITMAP MERGE
- BITMAP KEY ITERATION
- SORT BUFFER
- TABLE ACCESS ... D1
- BITMAP INDEX RANGE SCAN I_C1
- BITMAP MERGE
- BITMAP KEY ITERATION
- SORT BUFFER

Bitmap Star Transformation

- **The main components are a large *fact* table and a surrounding collection of *dimension* tables.**

- **A row in the *fact* table consists of a number of useful data elements, and a set of identifiers - typically short codes making up the concatenated key to the Dimensions**
- **Each dimension table is joined to the fact table using a Primary Key to Foreign Key join**
- **The identifying Primary Key fact or Foreign Key dimension column, have a *single column bitmap index* created over it on both the Fact and Dimension tables**
- **Single Column Bitmaps are best used by the optimizer for any combination of column order in your query; i.e. 123, 321, 213, 312, etc**
- **An identifying column on the *fact* table corresponds to one of the *dimension* tables, and the short codes that appear in that column must be selected from the (declared) *primary key* of that table**

Using Trusted Constraints

- **Keyword RELY or USING TRUSTED CONSTRAINTS**

- **Regarding materialized views and partitioned tables, such foreign key constraints in DSS databases are quite likely to be declared as *disabled*, *not validated*, and *rely*. (on keyword integrity)**
- **Oracle 10*g* adds the ability of a materialized view to choose more query rewrite options, generally resulting in better and more efficient execution of refreshes, via the USING TRUSTED CONSTRAINTS clause**
- **The USING TRUSTED CONSTRAINTS clause tells Oracle to use dimension and constraint information that the DBA has declared trustworthy, but that the database has not yet validated**
- **if Oracle determines that this information is invalid, the refresh procedure may instead corrupt the materialized view *even though the refresh operation itself returns a successful status***
- **If USING TRUSTED CONSTRAINTS is not specified, Oracle will use the default method, USING ENFORCED CONSTRAINTS, during the refresh operation**

Load Data, Build Indexes & Compute Statistics

- **Data loading**

- **Having created the tables, loaded them with suitable data, and then enabled the feature by issuing:**

```
alter session set
    star_transformation_enabled = true;
```

- **Following the data load we build the Bitmap indexes on the Primary Keys of the fact and Foreign Keys of the dimension tables**
- **This optimizes the fact join, referencing the dimension tables that are required for the Star Transformation of our query**

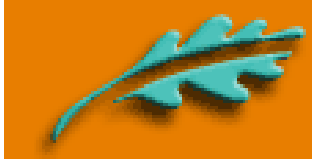
- **Estimate or compute statistics on the new objects and our star schema is now ready for action**

Star Query

```
select
    {pe.fact columns}
from
    towns   wt,
    towns   ht,
    people  pe
where
    wt.name = 'Coventry'
and ht.name = 'Birmingham'
and pe.id_town_home = ht.id
and pe.id_town_work = wt.id
;


select
    wt.name,
    ht.name,
    st.name
    {pe.fact columns}
from
    states  st,
    towns   wt,
    towns   ht,
    people  pe
where
    st.name = 'Alabama'
and wt.id_state = st.id
and ht.name = 'Birmingham'
and pe.id_town_home = ht.id
```

Bitmap Star Transformation Tips



- **The *bitmap star transformation* can be applied to partitioned tables, so extra steps relating to partitioning may appear, such as degree of parallelism.**
- **The path can execute in parallel, introducing extra levels of messy parallel distribution elements in the plan**
- **The join back of the *dimension* tables could be implemented as a series of hash joins, or sort merge joins instead of nested loop joins**

Recommended Books & Links

- ***Data Warehouse books & links for Ralph Kimball***

- **Books and links are available from:http://www.kimballgroup.com/html/books.html**

- **The Data Warehouse Lifecycle Toolkit, 2nd Edition: Practical Techniques for Building Data Warehouse and Business Intelligence Systems, John Wiley & Sons, 2008**

- **The world of data warehousing has changed remarkably since the first edition of *The Data Warehouse Lifecycle Toolkit* was published in 1998. The Kimball Group recently refined the original set of lifecycle methods and techniques. With significant amounts of new and updated material**

- ***The Data Warehouse Lifecycle Toolkit, 2nd Edition* will set the standard for DW/BI system design and development for the next decade**

Conclusion

- **The mind once expanded by a new idea, can never return to it's former Dimension**
  Justice Oliver Wendal Holmes
- **For further information contact *AIS*: mrichards@advanis.com**