

**BI Solution Concept**

**Mercedes-Benz Dealers**

|  |  |
| --- | --- |
| Requested By | Kyril Bucha |
| Business Owner | Kanstantsin Arzhakhouski |
| Contact Info. | Kanstantsin\_Arzhakhouski@epam.com |

**CONTENTS**

Logical Diagram ……………………………………………………………………………………………………….………………2

Physical diagram………………………………………………………………………………………………………………………3

Dimension ……………………………………………………………………………………………………………………………… 4

Dimension Types …………………………………………………………………………………………………………. 4

Dimension Hierarchies………………………………………………………………………………………………….. 5

Facts ………………………………………………………………………………………………………………………………………. 6

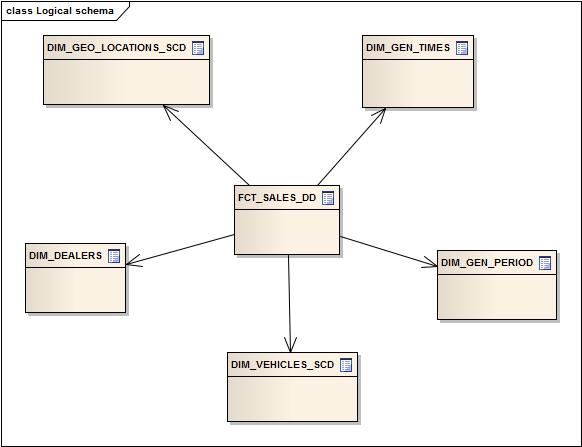
Facts Aggregations …………………………………………………………………………………………………………6

Dataflow Diagram ………………………………………………………………………………………………………………….. 7

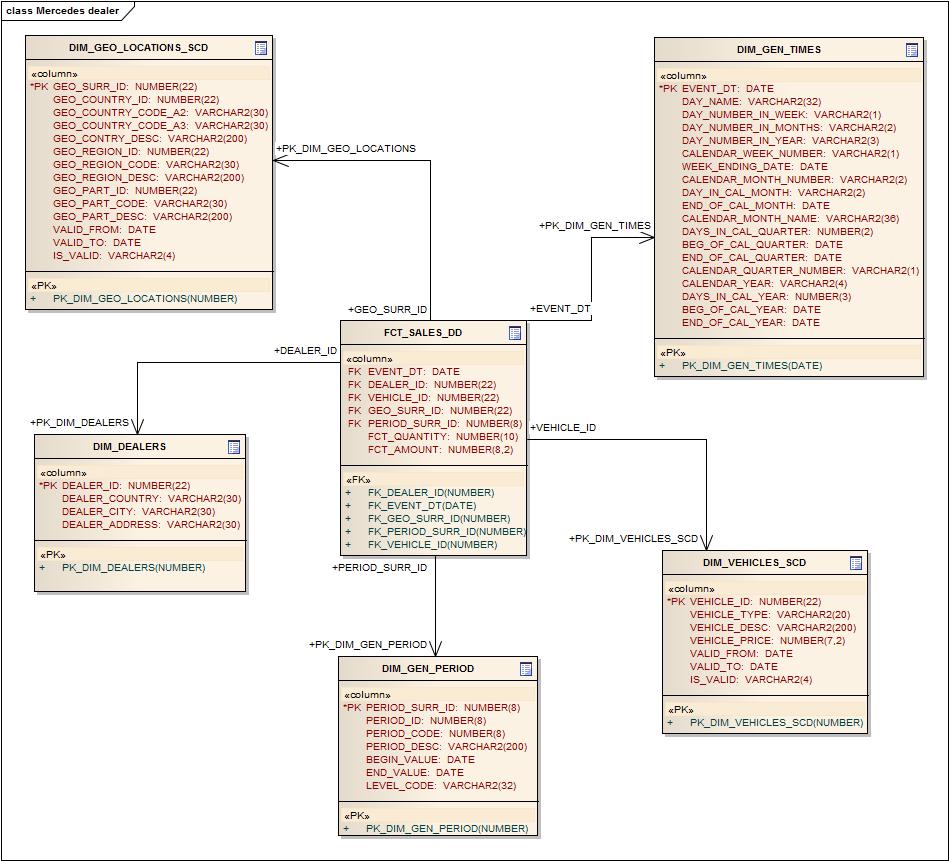
Partitioning rules ……………………………………………………………………………………………………………………. 8

Strategy of Parallel execution …………………………………………………………………………….………………….. 9

**LOGICAL DIAGRAM**



**PHYSICAL DIAGRAM**



**DIMENSIONS**

**DIMENSION TYPES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | TYPE | SIZE | MERGED DIMENSIONS | DESCRIPTION |
| DIM\_GEN\_TIMES | SCD1 | BIG | dw.t\_days  dw.lc\_days  dw.t\_weeks  dw.t\_months  dw.lc\_months  dw.t\_quarters  dw.t\_years  dw.t\_localizations | Dimension consists of a calendar value list |
| DIM\_GEO\_LOCATIONS\_SCD | SCD2 | SMALL | dw.t\_countries  dw.lc\_countries  dw.t\_cntr\_groups  dw.lc\_cntr\_groups  dw.t\_geo\_regions  dw.lc\_geo\_regions  dw.t\_geo\_objects  dw.t\_geo\_types  dw.t\_geo\_object\_links  dw.t\_localizations  dw.t\_geo\_action | Dimension with locatons information and hierarchy of countries, groups of countries and regions. |
| DIM\_GEN\_PERIODS | SCD1 | SMALL | dw.gen\_periods  dw.lc\_gen\_periods | Dimension with information about business periods of dealer’s activity |
| DIM\_VEHICLE\_SCD | SCD2 | BIG | dw.t\_vehicle\_type  dw.t\_vehicle\_desc  dw.lc\_vehicle\_type  dw.t\_vehicle\_price  dw.t\_localizations | Dimension with information about vehicles, and types. It includes the history of changing of price |
| DIM\_DEALERS | SCD1 | SMALL | dw.t\_dealers  lc.t\_dealers | Contain information about dealers |

**DIMENSIONS HIERARCHIES**

DIM \_TIMES:

Hierarchy DAY-MONTH-QUARTER-YEAR

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| DAYs | DAY | Store all day at the calendar | DAY\_ID |
| MONTHs | MONTH | Store all months at the calendar year | MONTH\_ID |
| QUARTERs | QUARTER | Store all quarters at the calendar  year | QUARTER\_ID |
| YEARs | YEAR | Store all years | YEAR\_ID |

DIM\_VEHICLE\_SCD:

Hierarchy CAR-TYPE

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| CARs | CAR | Store vehicle number | VEHICLE\_ID |
| TYPEs | TYPE | Store description of types of vehicle | VEHICLE\_TYPE |

**FACTS**

**FACTS AGGREGATIONS**

One-day granularity. This level consist of two aggregations (by number, by amount sold)

|  |  |
| --- | --- |
| Table Name | Aggregation Level |
| FCT\_SALES\_DD | Number of sold vehicles and amount sales. Every day calculation. |

**DATAFLOW DIAGRAM**

 Source files

**Star cleansing area**

**Data Mart**

**Star area**

DW area

DW Cleansing area

ETL procedure

3NF

Stage area

Cleansing area

**PARTITIONING RULES**

Main rule is separate stored data in different tablespace.

CREATE TABLE FCT\_INCOME\_PRODUCT\_DAYLY

(

event\_dt DATE NOT NULL,

...

)

PARTITION BY RANGE(event\_dt)

(

PARTITION 2000

VALUES LESS THAN (to\_date('12/31/2000', 'mm/dd/yyyy')) tablespace SALES\_2000,

PARTITION 2005

VALUES LESS THAN (to\_date('12/31/2005', 'mm/dd/yyyy')) tablespace SALES\_2005,

PARTITION 2010

VALUES LESS THAN (to\_date('12/31/2010', 'mm/dd/yyyy')) tablespace SALES\_2010,

PARTITION 2011

VALUES LESS THAN (to\_date('12/31/2011', 'mm/dd/yyyy')) tablespace SALES\_2011,

PARTITION 2012

VALUES LESS THAN (to\_date('12/31/2012', 'mm/dd/yyyy')) tablespace SALES\_2012,

PARTITION 2013

VALUES LESS THAN (to\_date('12/31/2013', 'mm/dd/yyyy')) tablespace SALES\_2013,

PARTITION INC\_OTHERS

VALUES LESS THAN (MAXVALUE) tablespace SALES\_OTHER

);

**STRATEGY OF PARALLEL EXECUTION**

For increase performance, use all types of parallel execution. Such as parallel query, parallel DML, parallel DML. Pay special attention to the use of parallel query. Parallel query allows a single SQL SELECT statement to be divided into many smaller queries, with each component query being run concurrently, and then the results from each combined to provide the final answer with highly performance.