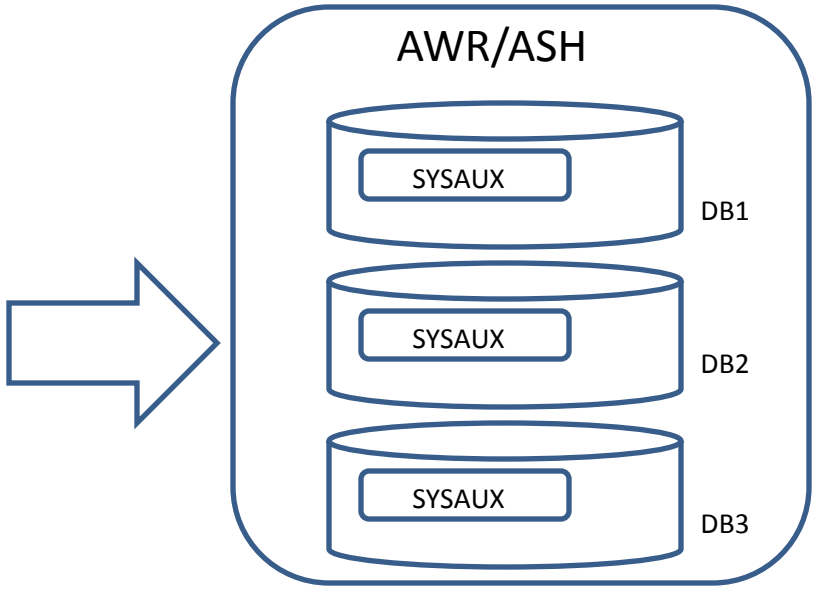
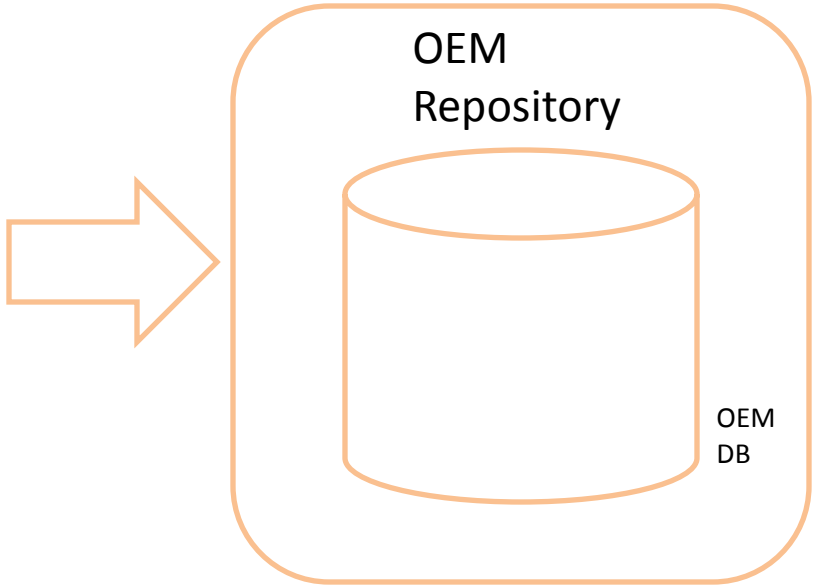
Unleash the Value of the Data in Your OEM Repository

Iordan K. Iotzov

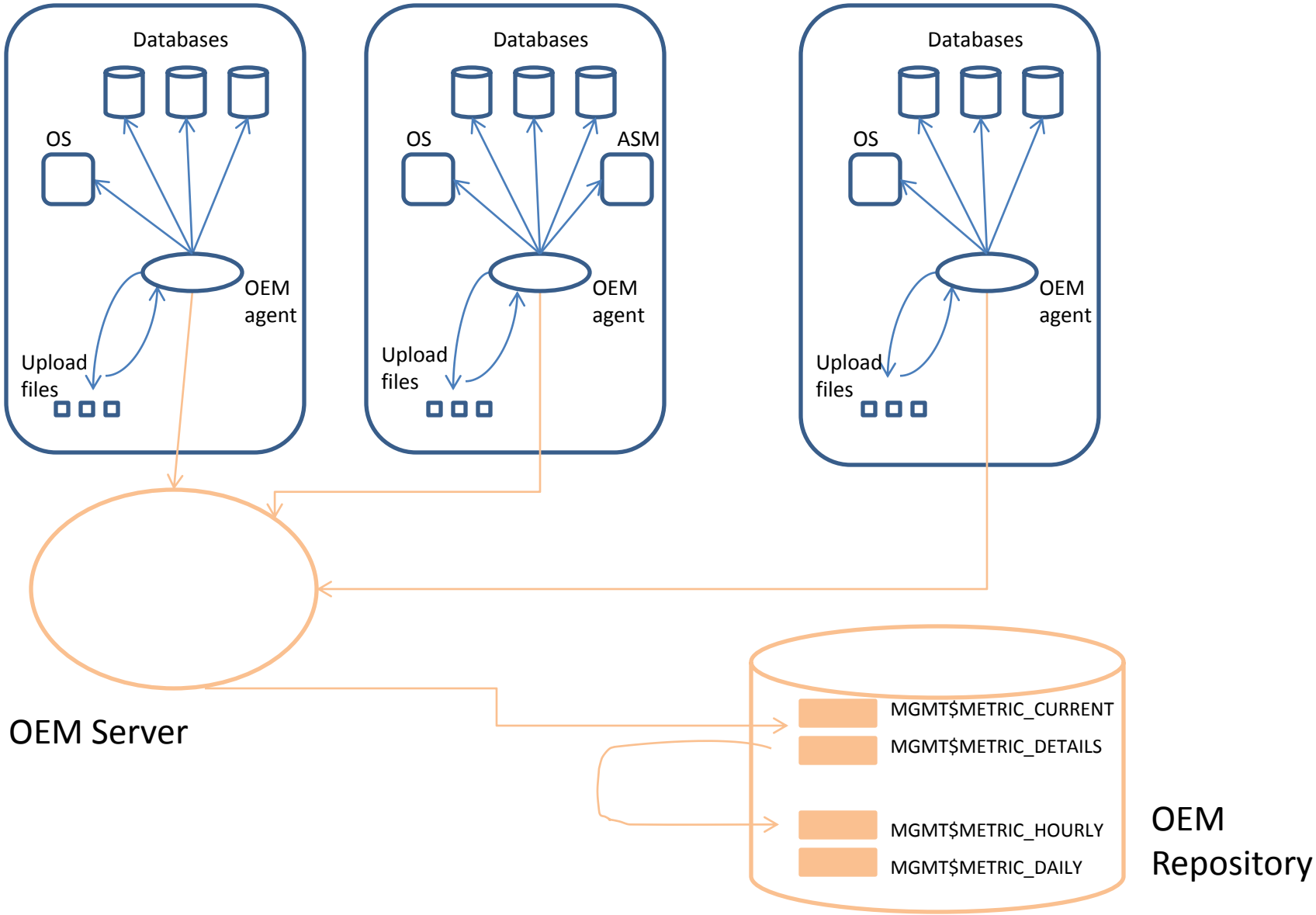
iiotzov@newsamerica.com
News America Marketing
(NewsCorp)

Sources of Information in OEM

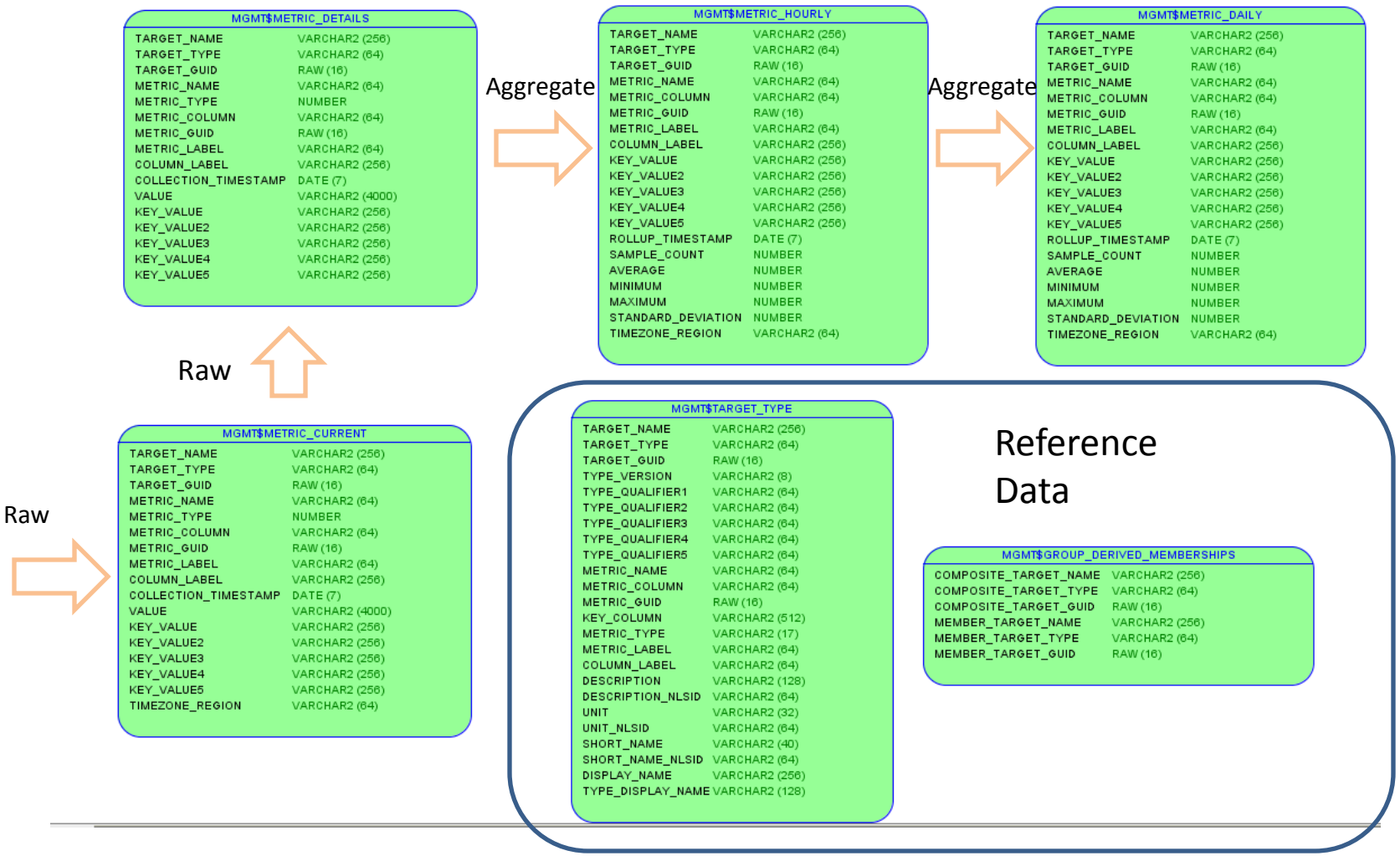




Process of Metrics Gathering in OEM

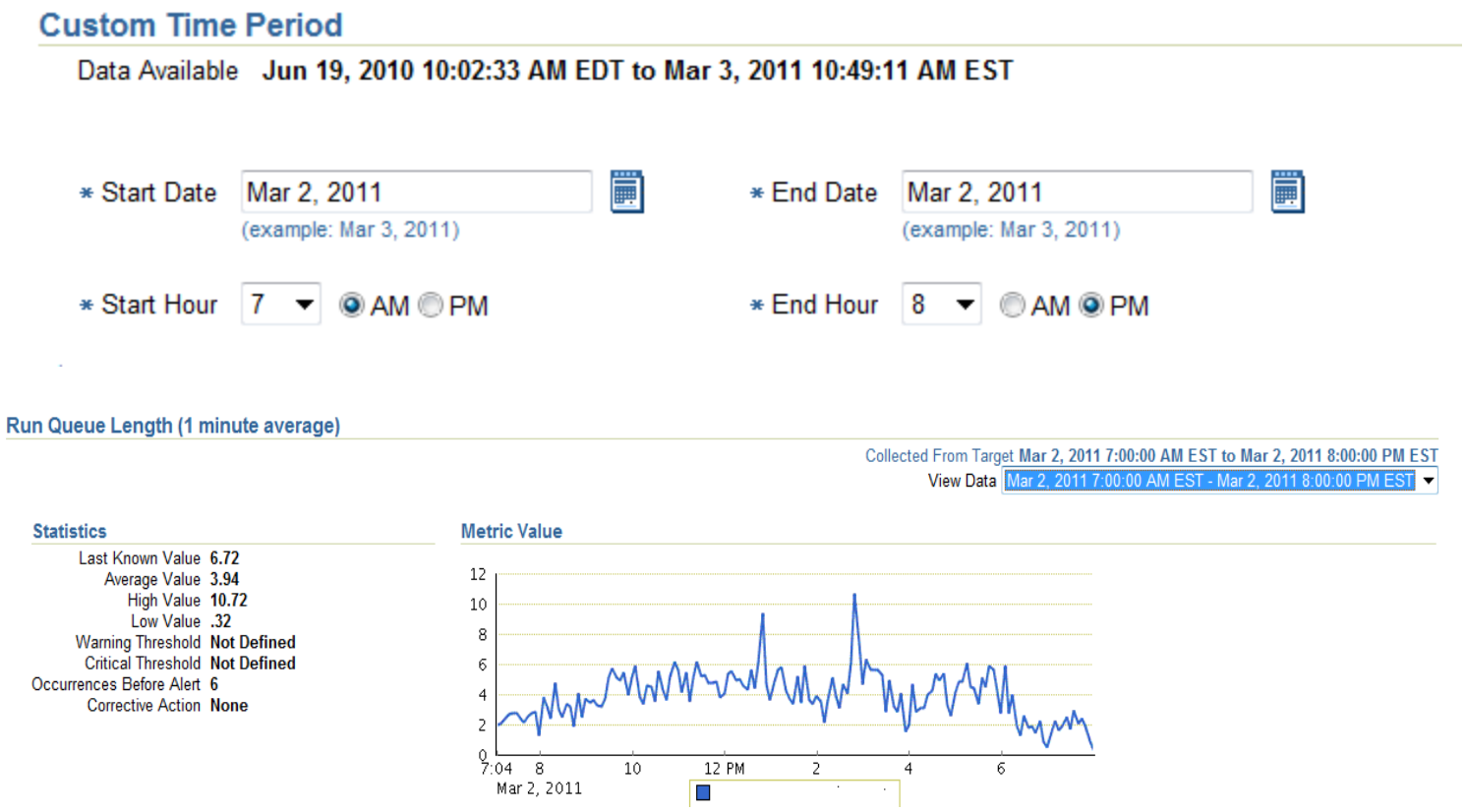


Some Metrics-related OEM Repository Tables



Useful Applications – averages over any period of time

Find the average run queue length during business hours(M-F, 7 am – 8 pm) for 30 days



Useful Applications – averages over any period of time

Find the average run queue length during business hours(M-F, 7 am – 8 pm) for 30 days

Query:

```
select
        avg(average)
from
        mgmt$metric_hourly
where
        rollup_timestamp        > sysdate – 30
and     target_name             = 'dbtest01'
and     metric_name             = 'Load'
and     column_labal            = 'Run Queue Length (1 minute average)'
and     to_char(rollup_timestamp,'DAY') not in ('SATURDAY','SUNDAY')
and     rollup_timestamp        between trunc(rollup_timestamp,'DD') + 7/24
                                and trunc(rollup_timestamp,'DD') + 20/24
```

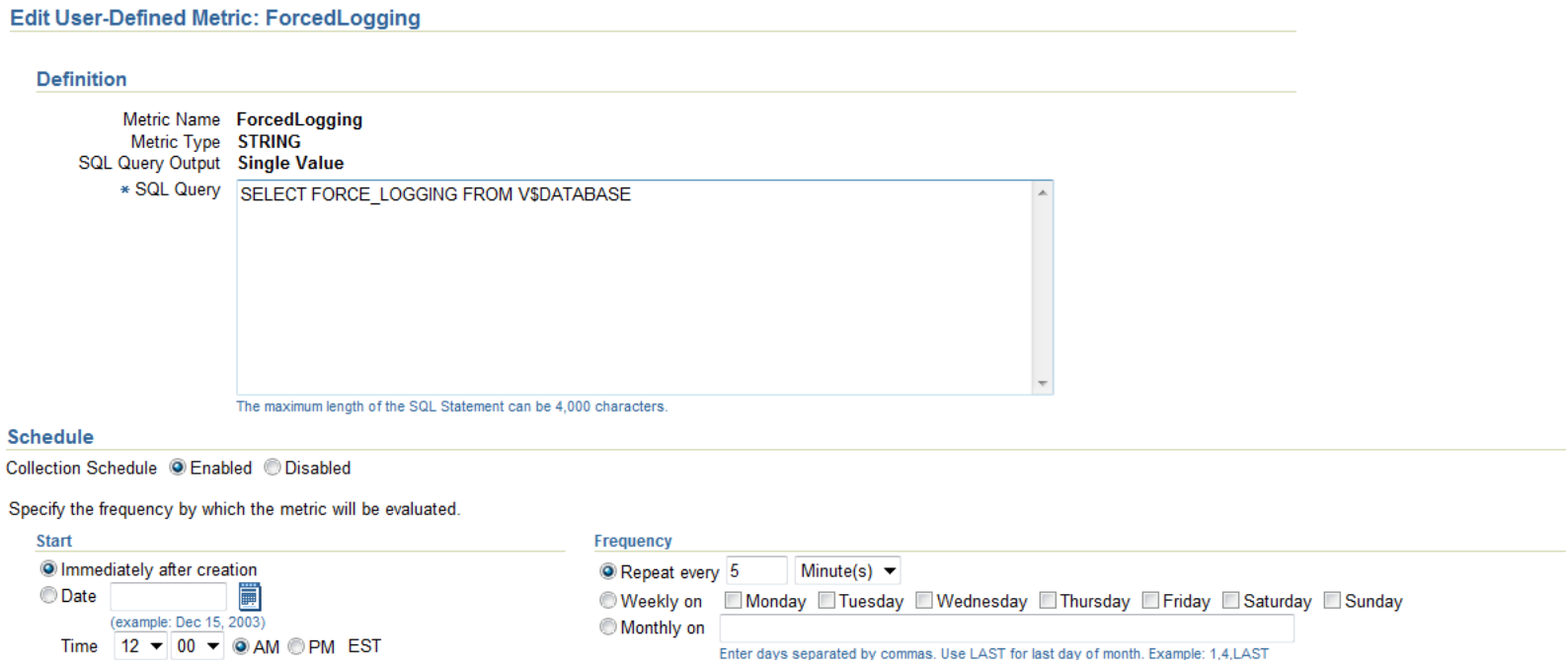
Output: 2.97053285256410256410256410256410256411

Opportunities: Unlimited flexibility in managing OEM metrics data

Useful Applications – enforcing enterprise-wide policies

Monitor Force Logging mode for all production databases.

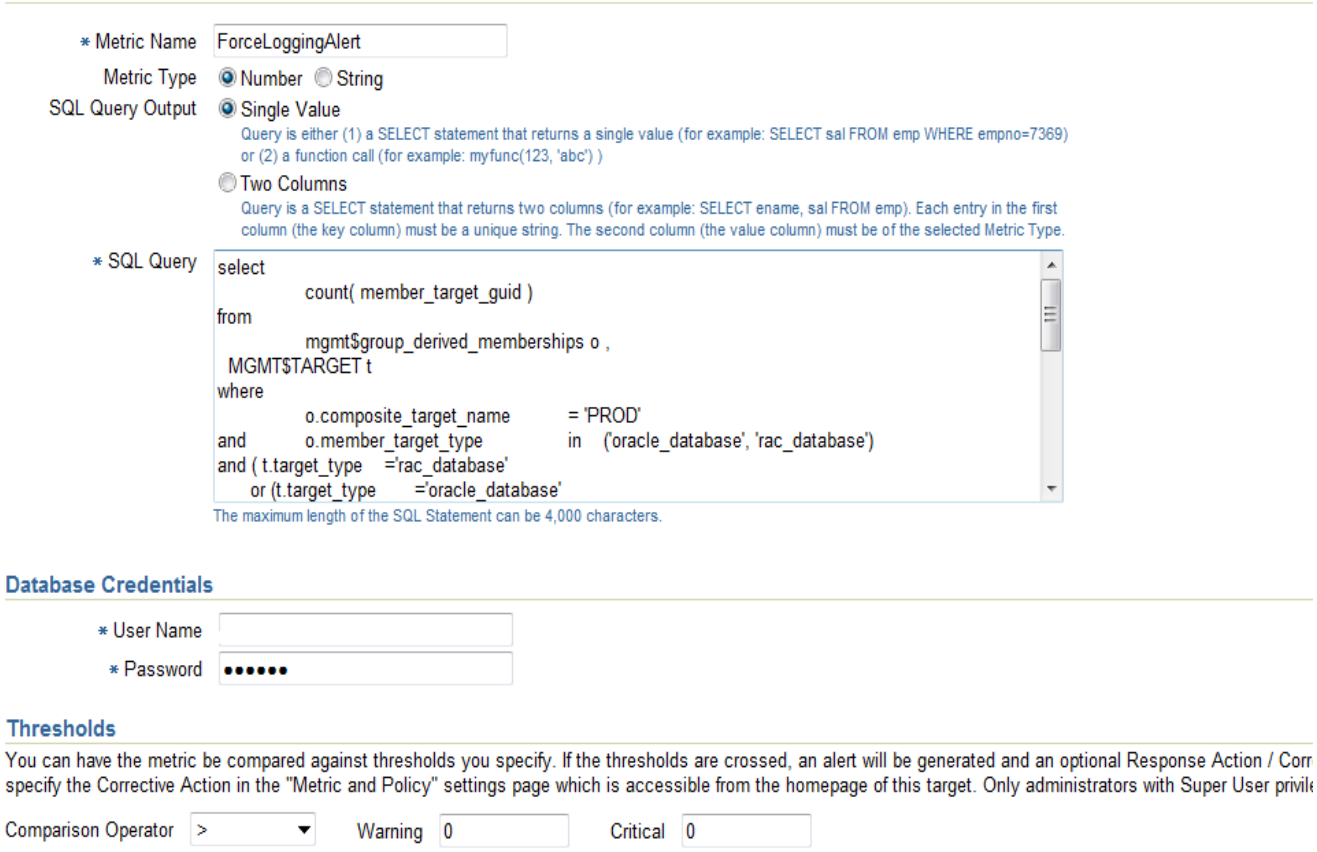
Create UDM in each production database. The sole purpose of this UDM would be to supply data to the OEM repository



Useful Applications – enforcing enterprise-wide policies

Monitor Force Logging mode for all production databases.

Create UDM in the OEM repository database. This UDM will notify/page DBAs if a DB is not in Force Logging mode



Useful Applications – enforcing enterprise-wide policies

Monitor Force Logging mode for all production databases.

The full query behind the OEM repository UDM:

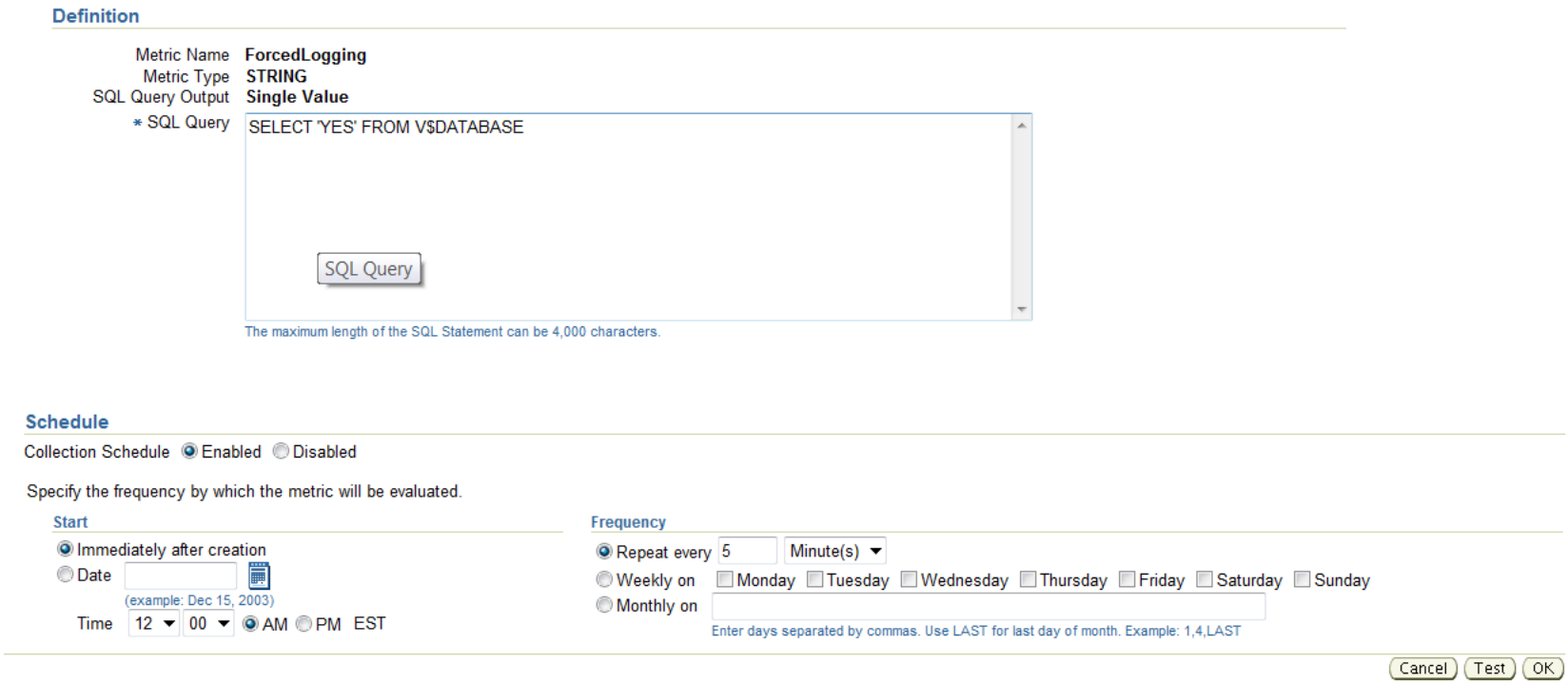
```
select
        count( member_target_guid )
from
        mgmt$group_derived_memberships o ,
        mgmt$target t
where
        o.composite_target_name     = ‘PROD'
and     o. member_target_guid       = t.target_guid
and     ( t.target_type             ='rac_database'
          or
         (t.target_type             ='oracle_database'
          and t.type_qualifier3     != 'RACINST‘
         )
        )

and     not exists (
                select
                        *
                from
                        mgmt$metric_current i
                where
                        i.target_guid           = o.member_target_guid
                and     metric_name             = 'SQLUDM'
                and     column_label            = 'ForcedLogging'
                and     Metric_Column           = 'StrValue'
                and     collection_timestamp    > sysdate - 20/1440
                and     value                   = 'YES‘
                 )
```

Useful Applications – enforcing enterprise-wide policies

Monitor Force Logging mode for all production databases.

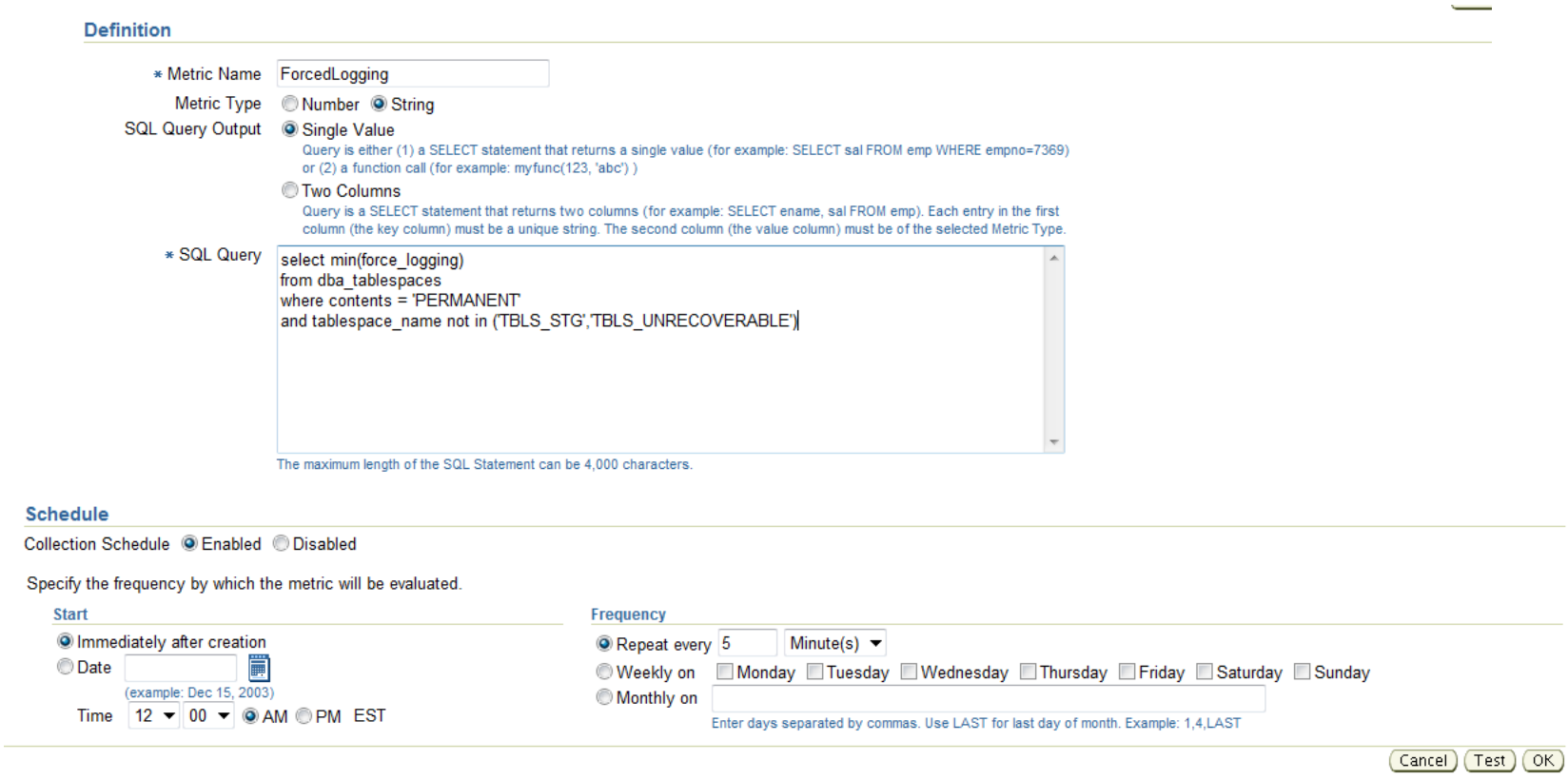
How about exempting a database? One need to do a concerted effort to exempt a database.



Useful Applications – enforcing enterprise-wide policies

Monitor Force Logging mode for all production databases.

Ability to deliver customized solutions – all but two tablespaces (TBLS_STG, TBLS_UNRECOVERABLE) must be in Force Logging mode



Useful Applications – integration with other data sources

Combining ASH/AWR and OEM repository

Where does the variation in single block read time come from?

```
select
        corr(db.time_waited , oem.value) , count(*)
from
        (select
                sample_time ,
                time_waited
        from
                dba_hist_active_sess_history@prod_db
        where
                event           = 'db file sequential read'
        and     session_state   = 'WAITING'
        and     sample_time     > sysdate - 1 ) db ,
        ( select
                collection_timestamp ,
                value
        from
                mgmt$metric_details
        where
                target_name     = '+ASM_PROD'
         and    metric_name      = 'Single_Instance_DiskGroup_Performance'
         and    metric_column   = '<ASM Metric>' ) oem
 where
         oem.collection_timestamp       between db.sample_time - 15/1440
                                        and db.sample_time
```

ASM Metric	Correlation
writes_ps	0.04
reads_ps	-0.04
write_throughput	0.04
read_throughput	0.01

Very different sampling rates usually result in lower correlation

Useful Applications – the power of Oracle's own SQL

Why is the latency of Streams Apply so high? Could it be the disk IO? It is often the issue…

Check:

```
select
        corr(streams_latency.value , IO_load.value)
from
        ( select *
        from
                mgmt$metric_details
        where
                target_name = 'STRDEST'
        and     metric_name = 'streams_latency_throughput'
        and     column_label = 'Latency' ) streams_latency ,
        (select *
        from
                mgmt$metric_details
        where
                target_name = 'STRDEST'
        and     metric_name = 'instance_throughput'
        and     column_label = 'I/O Megabytes (per second)') IO_load
where
        streams_latency.collection_timestamp between IO_load.collection_timestamp
                                                 and IO_load.collection_timestamp  + 5/(60*24)
and     streams_latency.collection_timestamp  > sysdate – 1
```

Output: -0.0010434366348630079753544589947484440986 31

Interpretation: Hypothesis rejected! Need to look for other explanation…

Useful Applications – the power of Oracle's own SQL

Why is the latency of Streams Apply so high? Maybe it is the redo volume on the Capture side…

Check:

```
select corr(streams_latency.value , redo_source_1.value + redo_source_2.value)
from
                ( select *
                  from
                                mgmt$metric_details
                  where
                                target_name = 'STRDEST'
                  and           metric_name = 'streams_latency_throughput'
                  and column_label = 'Latency' ) streams_latency ,
                (select *
                from
                                mgmt$metric_details
                where
                                target_name = 'STRSRC_STRSRC 1'
                and             metric_name = 'instance_throughput'
                and             column_label = 'Redo Generated (per second)' ) redo_source_1 ,
                 (select *
                from
                                mgmt$metric_details
                where
                                target_name = 'STRSRC_STRSRC 2'
                and             metric_name = 'instance_throughput'
                and             column_label = 'Redo Generated (per second)' ) redo_source_2
where
                streams_latency.collection_timestamp  between redo_source_1.collection_timestamp
                                                        and     redo_source_1.collection_timestamp  + 5/(60*24)
and             streams_latency.collection_timestamp  between redo_source_2.collection_timestamp
                                                        and redo_source_2.collection_timestamp  + 5/(60*24)
and             streams_latency.collection_timestamp > sysdate – 1
```

Output: 0.764129749129402128979924364231927912862 9

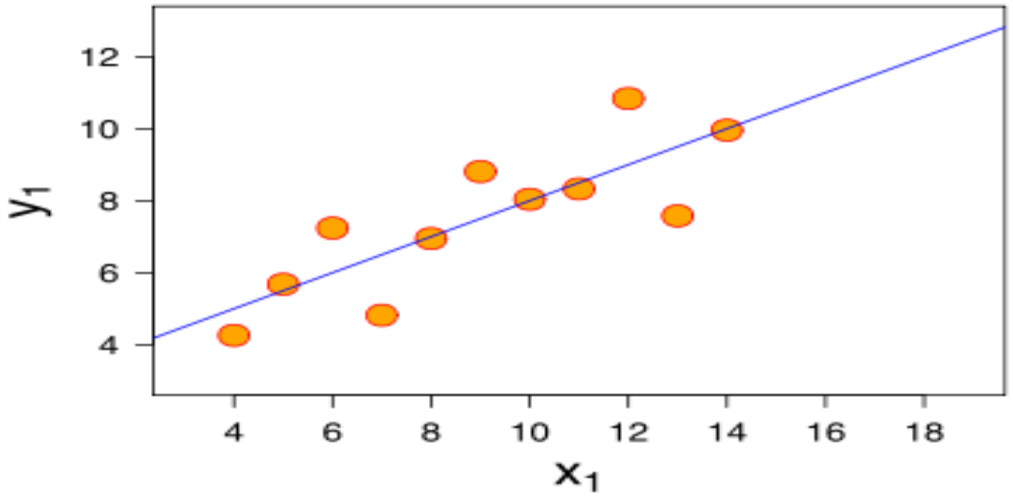
Interpretation: The redo volume on the Capture side is likely the reason for increased latency on the Apply side

Advanced Forecasting Example - tablespace size forecast

Simple linear regression:

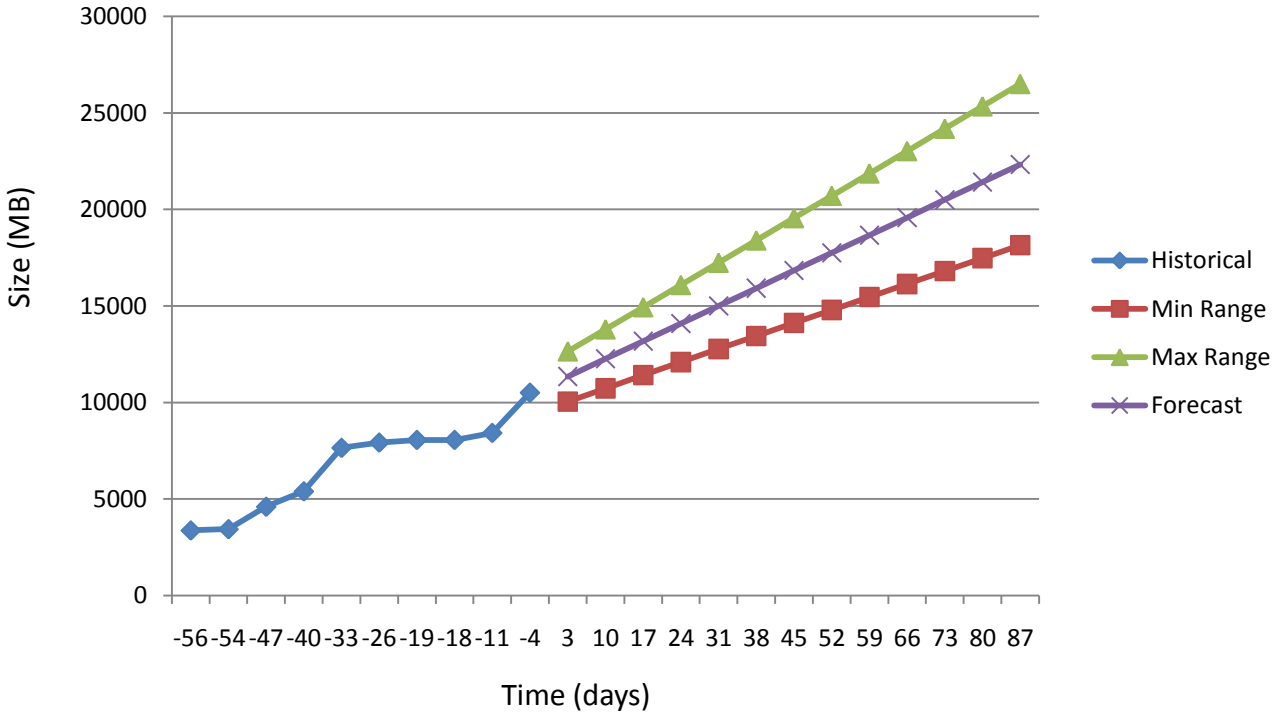
$$tbls(t) = tbls0 + incr*t$$

tbls0 (Intercept)	– tablespace size for time 0
incr (Slope)	– how fast a tablespace grows
t	– time
tbls(t)	– tablespace size at time t



Advanced Forecasting Example - tablespace size forecast

Quality of forecasting - 95% prediction interval of the regression line



Advanced Forecasting Example - tablespace size forecast

Getting raw historical tablespace sizing data from the OEM repository

```
insert into raw_data
select
        m.rollup_timestamp ,
        m.average
from
        mgmt$metric_daily m ,
        mgmt$target_type t
where
        (t.target_type    ='rac_database'
           or (t.target_type      ='oracle_database'
                and t.type_qualifiers3 != 'RACINST'))
and     m.target_guid                   =p_target_guid    (Database)
and     m.target_guid                   = t.target_guid
and     m.metric_guid                    =t.metric_guid
and     t.metric_name                   ='tbspAllocation‘
and     t.metric_column                  = 'spaceUsed'
and     m.rollup_timestamp              >= sysdate - p_period_hist
and     m.rollup_timestamp              <= sysdate
and     m.key_value                     = p_tablespace_name;
```

Advanced Forecasting Example - tablespace size forecast

Utilize Oracle's built-in packages and functions to compute most of the statistics in this project

```
select
        regr_intercept (average , rollup_timestamp -  sysdate  ) ,
        regr_slope (average , rollup_timestamp -  sysdate  ),
        regr_r2 (average , rollup_timestamp -  sysdate  ),
        regr_count (average ,rollup_timestamp -  sysdate  ),
        regr_avgx (average ,rollup_timestamp -  sysdate  ),
        regr_sxx (average ,rollup_timestamp -  sysdate  )  ,
        regr_syy (average ,rollup_timestamp -  sysdate  ),
        regr_sxy (average ,rollup_timestamp -  sysdate  )
into
        …..
from
        raw_data
```

Import data and create functions that are not available in Oracle

```
create or replace
FUNCTION f_forecast_t_dist (p_degree_freedom NUMBER , p_conf NUMBER) RETURN NUMBER IS
l_res           NUMBER;
l_one_tail_conf NUMBER;
BEGIN
        --- Custom Code
END f_forecast_t_dist;
```

Advanced Forecasting Example - tablespace size forecast

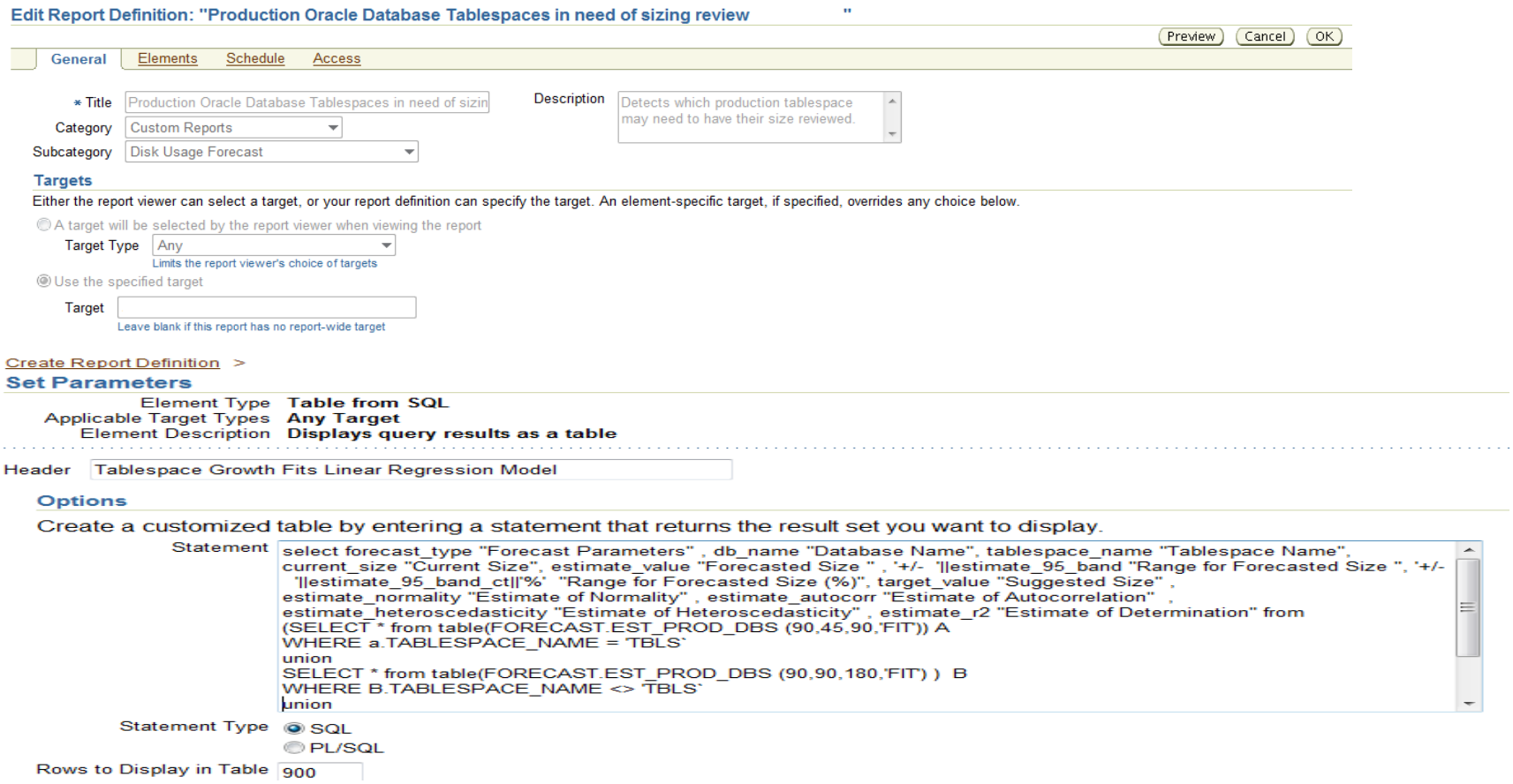
Final computations

```
l_ci:=sqrt ( 1/ l_cnt
              +
                   power((p_period_forecast - l_avgx),2)
                   /
                   l_sxx
              );

l_sigma:= sqrt(
                   abs(
                                 (l_syy
                                 -
                                 (power(l_sxy,2)/l_sxx)
                                 )
                        )
                        /(l_cnt-2)
                   ) ;
p_out_estimate_95_range:=(l_ci
                                 *
                                 f_forecast_t_dist(l_cnt-2,0.95)
                                 *
                                 l_sigma) ;
```

Advanced Forecasting Example - tablespace size forecast

Delivery Methods - Various



Advanced Forecasting Example - tablespace size forecast

Sample Output:

Tablespace Growth Fits Linear Regression Model

Forecast Parameters	Database Name	Tablespace Name	Current Size	Forecasted Size	Range for Forecasted Size	Range for Forecasted Size (%)	Suggested Size	Estimate of Normality	Estimate of Autocorrelation	Estimate of Heteroscedasticity	Estimate of Determination
90 days look back - 90 days forecast - 180 days sizing goal	DRRP	DRRP_DATA	83,100	98,199	+/- 26654	+/- 27%	128,354	0	0.65	0.27	0.54
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	LFD_DATA	16,000	17,645	+/- 3182	+/- 18%	25,518	0.83	0.56	0.05	0.87
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	LFD_INDX	3,100	3,589	+/- 783	+/- 21%	5,282	0.87	0.41	0.01	0.83
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	POSI_DATA	344,177	610,715	+/- 77604	+/- 12%	903,460	0.21	0.54	0.23	0.94
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	DMWW_DATA	19,000	20,224	+/- 3147	+/- 15%	25,342	0.25	0.71	0.28	0.74
90 days look back - 90 days forecast - 180 days sizing goal	ZFFO	SYSTEM	20,520	34,678	+/- 14119	+/- 40%	54,144	0.23	0.74	0.08	0.67
90 days look back - 90 days forecast - 180 days sizing goal	WPPT	TXCD	150	153	+/- 33	+/- 21%	197	0.55	0.54	0.08	0.64

Tablespace Growth Does not Fit Linear Regression Model

Forecast Parameters	Database Name	Tablespace Name	Current Size
90 days look back - 45 days forecast - 180 sizing goal	LSPPR	TEMP	1,000
90 days look back - 45 days forecast - 180 sizing goal	LSPPR	UNDOTBS1	516
90 days look back - 45 days forecast - 180 sizing goal	JBOB	TEMP	58
90 days look back - 45 days forecast - 180 sizing goal	FQQ	TEMP	14,300

References

- *Metalink Note 831243.1 (Examples: Creating Custom Reports)*.

- *Oracle® Enterprise Manager Extensibility 10g Release 2 (10.2) for Windows or UNIX*

- Liang's Blog: http://lianggang.wordpress.com/category/grid-control/

OEM 11g appears to have the same OEM repository structures as OEM 10g