Task 1:

Create mat view log

create materialized view log on agr\_trans

with primary key

including new values;

Create mat view:

create materialized view v\_agr\_trans

build immediate as

SELECT event\_dt,

company\_name

,ROUND ( AVG ( avg\_pct )

, 2 ) avg\_pct

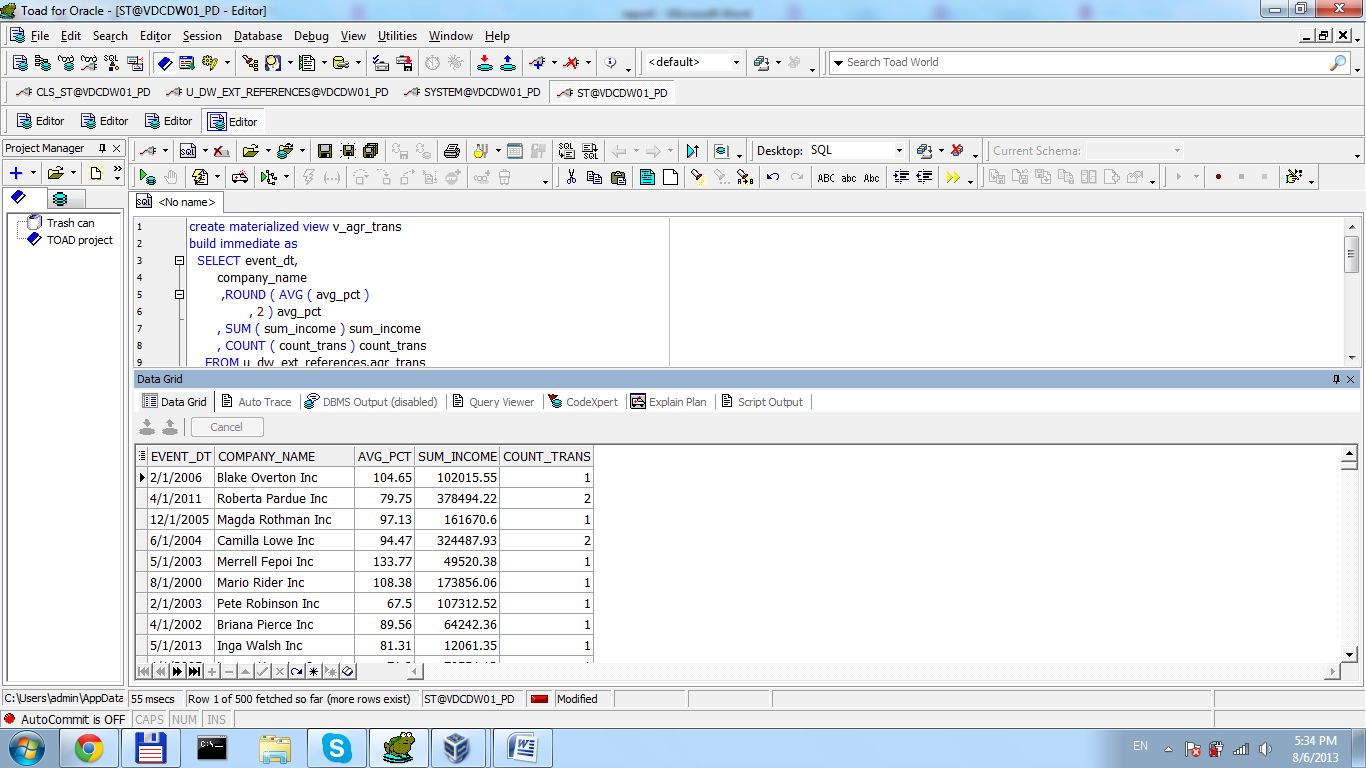
, SUM ( sum\_income ) sum\_income

, COUNT ( count\_trans ) count\_trans

FROM u\_dw\_ext\_references.agr\_trans

GROUP BY event\_dt, company\_name;

Screenshot:



Check:

update agr\_trans

set company\_name = 'CHECK\_VALUE'

where a\_id = 5;

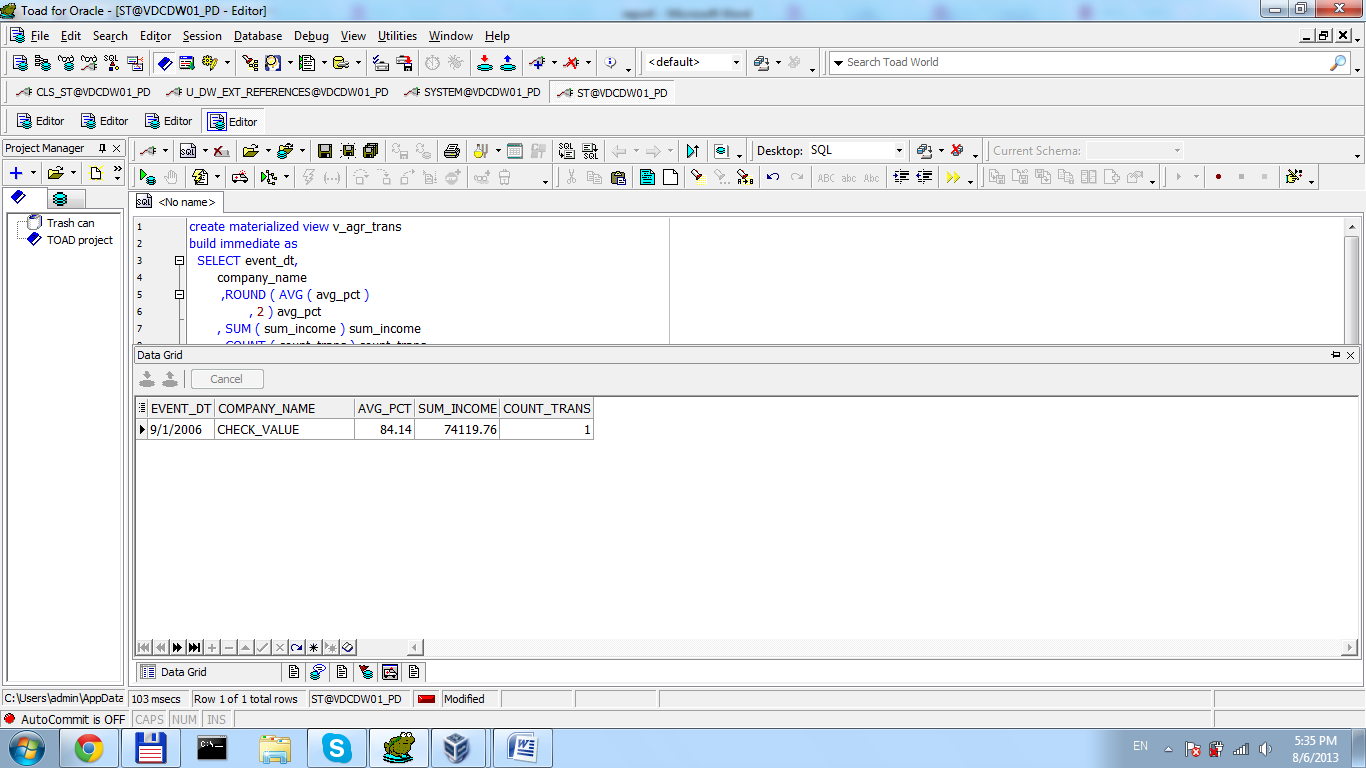
begin

***dbms\_mview.refresh***('v\_agr\_trans', 'c');

end;

select \* from v\_agr\_trans

where company\_name = 'CHECK\_VALUE';



Task 2:

create materialized view v\_agr\_trans

build immediate

refresh on commit

as

SELECT a\_id,event\_dt,

company\_name

, AVG ( avg\_pct )

avg\_pct

, count (avg\_pct) avg\_i

, SUM ( sum\_income ) sum\_income

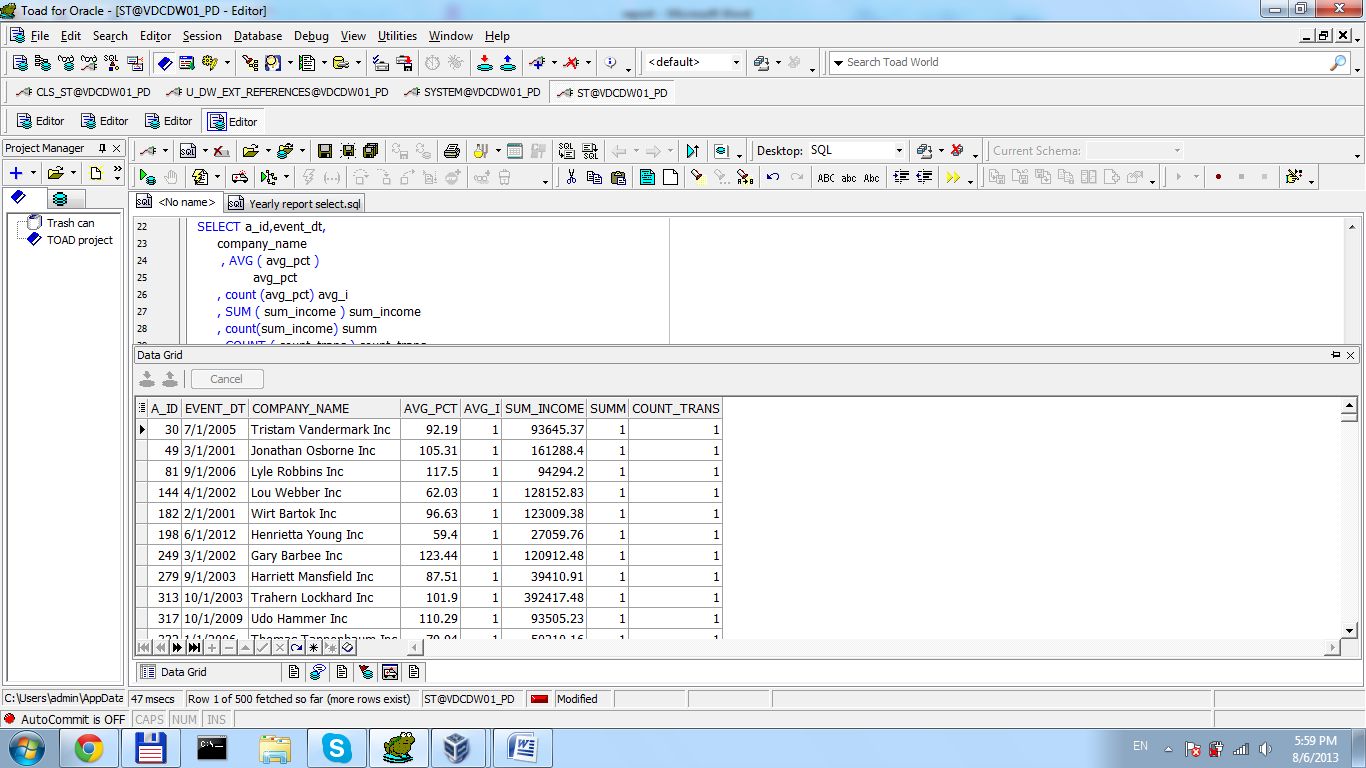
, count(sum\_income) summ

, COUNT ( count\_trans ) count\_trans

FROM u\_dw\_ext\_references.agr\_trans

GROUP BY a\_id,event\_dt, company\_name;

View has some changes because there are some constraints on ‘on commit refresh’



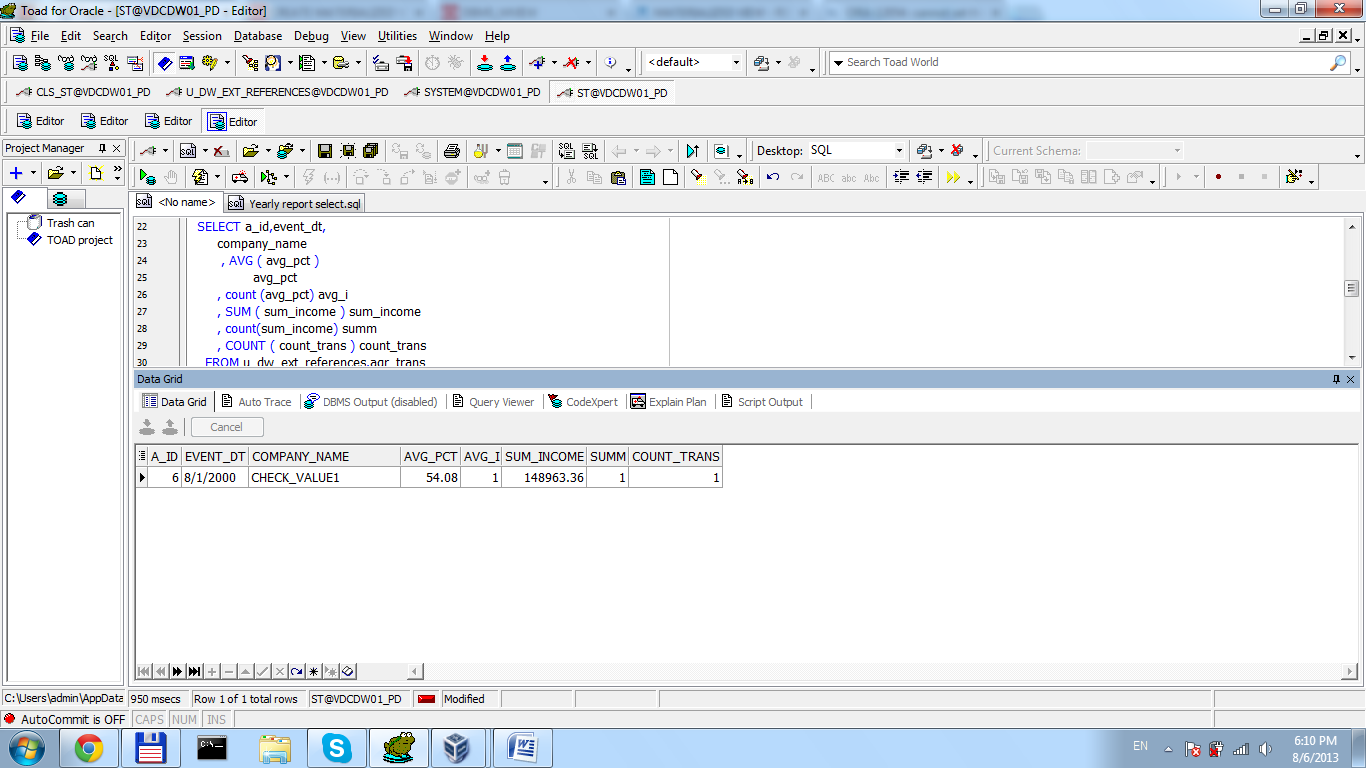
Check:

update agr\_trans

set company\_name = 'CHECK\_VALUE1'

where a\_id = 6;

commit;



Task 3:

create materialized view v\_agr\_trans

build immediate

refresh start with sysdate next (sysdate + 1/180)

as

*/\* Formatted on 8/6/2013 1:22:21 PM (QP5 v5.139.911.3011) \*/*

WITH temp AS ( SELECT TO\_CHAR ( TRUNC ( t.tran\_id

, 'MM' )

, 'Month' )

tran\_id

, cust.company\_name

, cust.cust\_city

, p.port\_identifier port\_id

, AVG ( t.dep\_goods / t.ar\_goods ) \* 100 pct

, SUM ( t.dep\_goods \* prod.income\_coef ) income

, COUNT ( t.dep\_goods ) cnt

FROM u\_dw\_ext\_references.t\_trans t

JOIN u\_dw\_ext\_references.ext\_products prod

ON ( t.prod\_id = prod.prod\_id )

JOIN u\_dw\_ext\_references.ext\_prod\_categories cat

ON ( prod.prod\_category\_id = cat.prod\_category\_id )

JOIN u\_dw\_ext\_references.ext\_ship sh

ON ( t.ship\_id = sh.ship\_unique\_identifier )

JOIN u\_dw\_ext\_references.ports p

ON ( t.dep\_port = p.port\_identifier )

JOIN u\_dw\_ext\_references.ports s

ON ( t.ar\_port = s.port\_identifier )

JOIN u\_dw\_ext\_references.ext\_insurances ins

ON ( t.ins\_id = ins.unique\_identifier )

JOIN u\_dw\_ext\_references.ext\_customers cust

ON ( t.cust\_id = cust.customer\_identifier )

WHERE TO\_CHAR ( TRUNC ( t.tran\_id

, 'YYYY' )

, 'YYYY' ) = '2011'

AND TO\_CHAR ( TRUNC ( t.tran\_id

, 'MM' )

, 'MM' ) = '01'

GROUP BY t.tran\_id

, cust.cust\_city

, p.port\_identifier

, cust.company\_name)

SELECT tran\_id month

, company\_name

, pct

, income

, cnt

FROM temp

MODEL

DIMENSION BY ( tran\_id, company\_name, cust\_city, port\_id )

MEASURES ( pct pct, income income, cnt cnt )

RULES

( pct [FOR tran\_id IN

(SELECT tran\_id

FROM temp), 'ALL\_COMPANIES', NULL, NULL] = AVG ( pct )[CV ( tran\_id ), company\_name, ANY, ANY],

pct ['TOTAL', NULL, NULL, NULL] = AVG ( pct )[ANY, ANY, ANY, ANY],

income [FOR tran\_id IN

(SELECT tran\_id

FROM temp), 'ALL\_COMPANIES', NULL, NULL] = SUM ( income )[CV ( tran\_id ), company\_name, ANY, ANY],

income ['TOTAL', NULL, NULL, NULL] = SUM ( income )[ANY, ANY, ANY, ANY],

cnt [FOR tran\_id IN

(SELECT tran\_id

FROM temp), 'ALL\_COMPANIES', NULL, NULL] = SUM ( pct )[CV ( tran\_id ), company\_name, ANY, ANY],

cnt ['TOTAL', NULL, NULL, NULL] = COUNT ( pct )[ANY, ANY, ANY, ANY] )

Screenshots:

