# Task

1.1. Prepare Report Layout

The Main Task is to create SQL for Report Layout that was developed in the concepts of your business in the last module (Use STAR schema objects for source of Data).

1.2. Compare Report Layout Performance

The Main Task is to create summarize table with comparison Performance of next Report Layout:

- Advancing Grouping (GROUP BY GROUPING SETs – Lab Work 02)

- 3NF (Staging)

- Star Schema (Lab Work 11)

Task Results:

• Prepare Document with description of results.

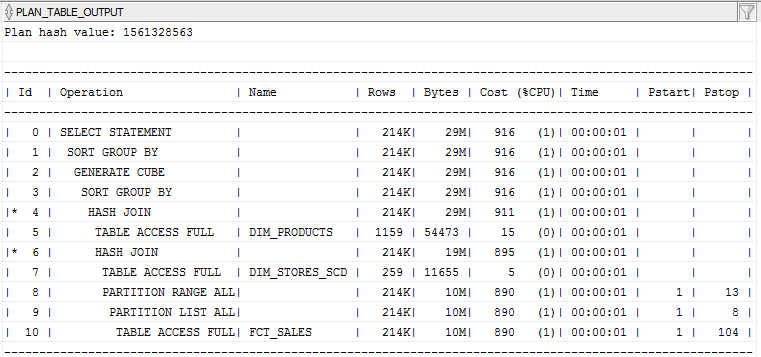
2. Advanced Loading

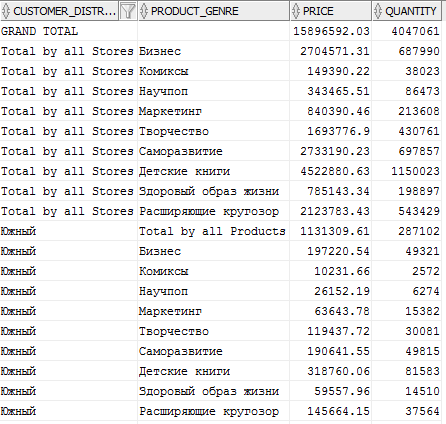
Implement Fact Table load with use of EXCHANGE PARTITION clause and logic of load for some rolling window period.

# Solution

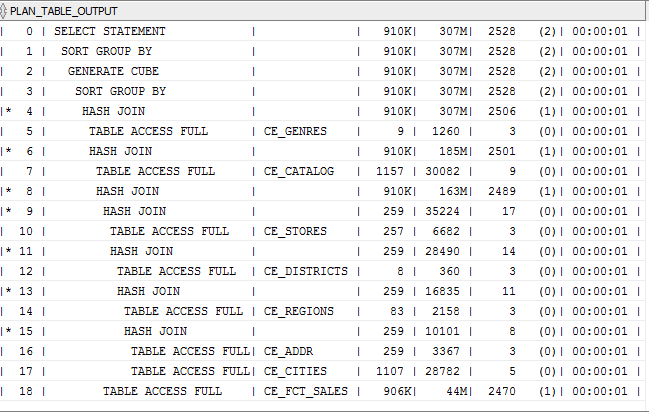
## Perfomance

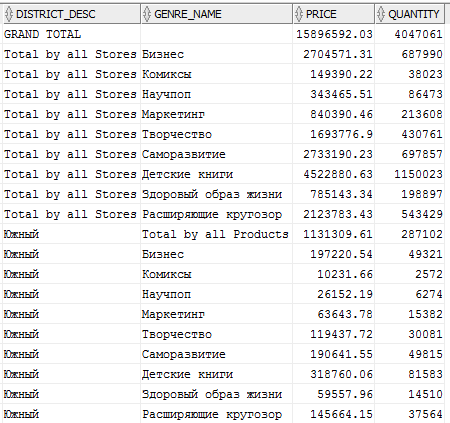
### Dim-layer





### 3NF layer





### Conclusion

Один и тот же запрос показывает разные результаты в зависимости от того, где он запущен:

* BL\_DM: cost=916
* BL\_3NF: cost=2528

Разница очевидна: на BL\_DM запрос выполняется практически в 3 раза эффективнее, чем на уровне BL\_3NF.

Это обусловлено тем, что в схеме Звезда данные уже сагрегированы, потому количество join-ов значительно меньше.

## Partition exchange

Cоздала таблицу fct\_sales1 такую же по структуре, как fct\_sales + range partition по event\_dt.

EXECUTE pckg\_drop.drop\_proc(object\_name=>'Fct\_Sales1',object\_type=>'table');

CREATE TABLE Fct\_Sales1

(

Event\_dt DATE NOT NULL,

Fct\_customer\_id NUMBER(**8**) NOT NULL,

Fct\_employee\_id NUMBER(**8**) NOT NULL,

Fct\_store\_id NUMBER(**8**) NOT NULL,

Fct\_store\_dist\_id NUMBER(**8**) NOT NULL,

Fct\_product\_id NUMBER(**8**) NOT NULL,

Fct\_payment\_id NUMBER(**8**) NOT NULL,

Fct\_check\_id NUMBER(**8**) NOT NULL,

Fct\_quantity NUMBER(**5**) NOT NULL,

Fct\_discount NUMBER(**5**,**3**) NOT NULL,

Fct\_unit\_price\_BYN NUMBER(**10**,**4**) NOT NULL,

Fct\_unit\_price\_disc\_BYN generated always AS (Fct\_unit\_price\_BYN\*Fct\_discount),

Fct\_sales\_Amount\_BYN generated always AS (Fct\_unit\_price\_BYN\*Fct\_discount \*Fct\_quantity ) ,

Fct\_BYN\_USD NUMBER(**8**,**3**) ,

Fct\_unit\_price\_USD generated always AS (Fct\_BYN\_USD \* Fct\_unit\_price\_BYN),

Fct\_unit\_price\_disc\_USD generated always AS (CEIL(Fct\_BYN\_USD \* Fct\_unit\_price\_BYN\*Fct\_discount)) ,

Fct\_sales\_Amount\_USD generated always AS (CEIL(Fct\_BYN\_USD \* Fct\_unit\_price\_BYN\*Fct\_discount \*Fct\_quantity)) ,

Insert\_DT DATE DEFAULT(sysdate) NOT NULL,

CONSTRAINT fk\_date1 FOREIGN KEY (Event\_dt) REFERENCES dim\_time\_day(full\_date\_dt),

CONSTRAINT fk\_customer1 FOREIGN KEY (Fct\_customer\_id) REFERENCES dim\_customers(customer\_sur\_id),

CONSTRAINT fk\_employee1 FOREIGN KEY (Fct\_employee\_id) REFERENCES dim\_employees(employee\_sur\_id),

CONSTRAINT fk\_store1 FOREIGN KEY (Fct\_store\_id) REFERENCES dim\_stores\_scd(store\_sur\_id),

CONSTRAINT fk\_product1 FOREIGN KEY (Fct\_product\_id) REFERENCES dim\_products(product\_sur\_id),

CONSTRAINT fk\_payment1 FOREIGN KEY (Fct\_payment\_id) REFERENCES dim\_payments(payment\_sur\_id)

)

PARTITION BY RANGE (event\_dt)

(

PARTITION sales\_2005\_2010 VALUES LESS THAN (TO\_DATE('01-JAN-2010','dd-Mon-yyyy'))tablespace tbs1, --2005-2020

PARTITION sales\_2010\_2011 VALUES LESS THAN (TO\_DATE('01-JAN-2011','dd-Mon-yyyy'))tablespace tbs2,

PARTITION sales\_2011\_2012 VALUES LESS THAN (TO\_DATE('01-JAN-2012','dd-Mon-yyyy')) tablespace tbs3 ,

PARTITION sales\_2012\_2013 VALUES LESS THAN (TO\_DATE('01-JAN-2013','dd-Mon-yyyy')) tablespace tbs4,

PARTITION sales\_2013\_2014 VALUES LESS THAN (TO\_DATE('01-JAN-2014','dd-Mon-yyyy')) tablespace tbs5,

PARTITION sales\_2014\_2015 VALUES LESS THAN (TO\_DATE('01-JAN-2015','dd-Mon-yyyy')) tablespace tbs6,

PARTITION sales\_2015\_2016 VALUES LESS THAN (TO\_DATE('01-JAN-2016','dd-Mon-yyyy')) tablespace tbs7,

PARTITION sales\_2016\_2017 VALUES LESS THAN (TO\_DATE('01-JAN-2017','dd-Mon-yyyy')) tablespace tbs8,

PARTITION sales\_2017\_2018 VALUES LESS THAN (TO\_DATE('01-JAN-2018','dd-Mon-yyyy'))tablespace tbs8,

PARTITION sales\_2018\_2019 VALUES LESS THAN (TO\_DATE('01-JAN-2019','dd-Mon-yyyy'))tablespace tbs8,

PARTITION sales\_2019\_2020 VALUES LESS THAN (TO\_DATE('01-JAN-2020','dd-Mon-yyyy'))tablespace tbs8,

PARTITION sales\_2020\_2021 VALUES LESS THAN (TO\_DATE('01-JAN-2021','dd-Mon-yyyy'))tablespace tbs8,

PARTITION sales\_2021\_more VALUES LESS THAN (maxvalue)

) ;

Затем создала таблицу fct\_sales\_2018, куда, предположительно, будут заливать данные за продажи 2018 года (она не партиционирована).

EXECUTE pckg\_drop.drop\_proc(object\_name=>'Fct\_Sales\_2018',object\_type=>'table');

CREATE TABLE Fct\_Sales\_2018

(

Event\_dt DATE NOT NULL,

Fct\_customer\_id NUMBER(**8**) NOT NULL,

Fct\_employee\_id NUMBER(**8**) NOT NULL,

Fct\_store\_id NUMBER(**8**) NOT NULL,

Fct\_store\_dist\_id NUMBER(**8**) NOT NULL,

Fct\_product\_id NUMBER(**8**) NOT NULL,

Fct\_payment\_id NUMBER(**8**) NOT NULL,

Fct\_check\_id NUMBER(**8**) NOT NULL,

Fct\_quantity NUMBER(**5**) NOT NULL,

Fct\_discount NUMBER(**5**,**3**) NOT NULL,

Fct\_unit\_price\_BYN NUMBER(**10**,**4**) NOT NULL,

Fct\_unit\_price\_disc\_BYN generated always AS (Fct\_unit\_price\_BYN\*Fct\_discount),

Fct\_sales\_Amount\_BYN generated always AS (Fct\_unit\_price\_BYN\*Fct\_discount \*Fct\_quantity ) ,

Fct\_BYN\_USD NUMBER(**8**,**3**) ,

Fct\_unit\_price\_USD generated always AS (Fct\_BYN\_USD \* Fct\_unit\_price\_BYN),

Fct\_unit\_price\_disc\_USD generated always AS (CEIL(Fct\_BYN\_USD \* Fct\_unit\_price\_BYN\*Fct\_discount)) ,

Fct\_sales\_Amount\_USD generated always AS (CEIL(Fct\_BYN\_USD \* Fct\_unit\_price\_BYN\*Fct\_discount \*Fct\_quantity)) ,

Insert\_DT DATE DEFAULT(sysdate) NOT NULL,

CONSTRAINT fk\_date12 FOREIGN KEY (Event\_dt) REFERENCES dim\_time\_day(full\_date\_dt),

CONSTRAINT fk\_customer21 FOREIGN KEY (Fct\_customer\_id) REFERENCES dim\_customers(customer\_sur\_id),

CONSTRAINT fk\_employee12 FOREIGN KEY (Fct\_employee\_id) REFERENCES dim\_employees(employee\_sur\_id),

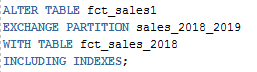
CONSTRAINT fk\_store12 FOREIGN KEY (Fct\_store\_id) REFERENCES dim\_stores\_scd(store\_sur\_id),

CONSTRAINT fk\_product12 FOREIGN KEY (Fct\_product\_id) REFERENCES dim\_products(product\_sur\_id),

CONSTRAINT fk\_payment21 FOREIGN KEY (Fct\_payment\_id) REFERENCES dim\_payments(payment\_sur\_id)

);

Осуществила partition exchange:



Проверка:

Было:



Стало:



Операция осуществлена успешно.