 Oracle OCP 11G 051答案解析

1. View the Exhibit and examine the structure of the SALES, CUSTOMERS, PRODUCTS, and TIMES

tables.

The PROD\_ID column is the foreign key in the SALES table, which references the PRODUCTS table.

Similarly, the CUST\_ID and TIME\_ID columns are also foreign keys in the SALES table referencing the

CUSTOMERS and TIMES tables, respectively.

Evaluate the following CREATE TABLE command:

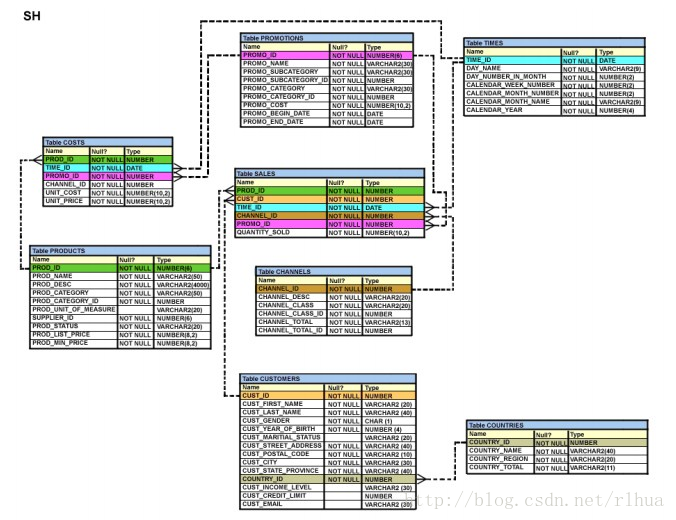
CREATE TABLE new\_sales(prod\_id, cust\_id, order\_date DEFAULT SYSDATE)

AS

SELECT prod\_id, cust\_id, time\_id

FROM sales;

Which statement is true regarding the above command?



A. The NEW\_SALES table would not get created because the DEFAULT value cannot be specified in the

column definition.

B. The NEW\_SALES table would get created and all the NOT NULL constraints defined on the specified

columns would be passed to the new table.

C. The NEW\_SALES table would not get created because the column names in the CREATE TABLE

command and the SELECT clause do not match.

D. The NEW\_SALES table would get created and all the FOREIGN KEY constraints defined on the

specified columns would be passed to the new table.

Answer: B

答案解析：

A答案：默认sysdate可以在列出被指定。

sh@TEST0924> CREATE TABLE new\_sales(prod\_id, cust\_id, order\_date DEFAULT SYSDATE) AS

  2  SELECT prod\_id, cust\_id, time\_id FROM sales;

Table created.

B答案：NULL约束会传递。由以下可以看出，not null是可以传递的。B正确

第一种方法，查看表的结构，可见not null约束是可以传递的。

sh@TEST0924> desc SALES

 Name                                                  Null?    Type

----------------------------------------------------- -------- ------------------------------------

 PROD\_ID                                               NOT NULL NUMBER

 CUST\_ID                                               NOT NULL NUMBER

 TIME\_ID                                               NOT NULL DATE

 CHANNEL\_ID                                       NOT NULL NUMBER

 PROMO\_ID                                          NOT NULL NUMBER

 QUANTITY\_SOLD                                NOT NULL NUMBER(10,2)

 AMOUNT\_SOLD                                  NOT NULL NUMBER(10,2)

sh@TEST0924> desc new\_sales

 Name                                                  Null?    Type

----------------------------------------------------- -------- ------------------------------------

 PROD\_ID                                               NOT NULL NUMBER

 CUST\_ID                                               NOT NULL NUMBER

 ORDER\_DATE                                     NOT NULL DATE

第二种方法：由下面的查询看出，not null也是传递的。B正确

sh@TEST0924> select a.owner,a.TABLE\_NAME,b.COLUMN\_NAME,a.CONSTRAINT\_NAME,a.CONSTRAINT\_TYPE,a.SEARCH\_CONDITION

  2  from USER\_CONSTRAINTS a,USER\_CONS\_COLUMNS b

  3  where a.CONSTRAINT\_NAME=b.CONSTRAINT\_NAME and A.table\_name='SALES';

OWNER      TABLE\_NAME COLUMN\_NAME     CONSTRAINT\_NAME                CONSTRAINT SEARCH\_CONDITION

---------- ---------- --------------- ------------------------------ ---------- ------------------------------

SH         SALES      CUST\_ID                    SALES\_CUSTOMER\_FK              R

SH         SALES      PROD\_ID                  SALES\_PRODUCT\_FK               R

SH         SALES      TIME\_ID                   SALES\_TIME\_FK                          R

SH         SALES      CHANNEL\_ID          SALES\_CHANNEL\_FK               R

SH         SALES      PROD\_ID                  SYS\_C0011009                           C                  "PROD\_ID" IS NOT NULL

SH         SALES      CUST\_ID                   SYS\_C0011010                           C                  "CUST\_ID" IS NOT NULL

SH         SALES      TIME\_ID                  SYS\_C0011011                           C                  "TIME\_ID" IS NOT NULL

SH         SALES      CHANNEL\_ID          SYS\_C0011012                       C              "CHANNEL\_ID" IS NOT NULL

SH         SALES      PROMO\_ID              SYS\_C0011013                       C              "PROMO\_ID" IS NOT NULL

SH         SALES      QUANTITY\_SOLD   SYS\_C0011014                       C          "QUANTITY\_SOLD" IS NOT NULL

SH         SALES      AMOUNT\_SOLD     SYS\_C0011015                       C          "AMOUNT\_SOLD" IS NOT NULL

SH         SALES      PROMO\_ID            SALES\_PROMO\_FK                 R

12 rows selected.

sh@TEST0924> l

  1   select a.owner,a.TABLE\_NAME,b.COLUMN\_NAME,a.CONSTRAINT\_NAME,a.CONSTRAINT\_TYPE,a.SEARCH\_CONDITION

  2  from USER\_CONSTRAINTS a,USER\_CONS\_COLUMNS b

  3\*  where a.CONSTRAINT\_NAME=b.CONSTRAINT\_NAME and A.table\_name='NEW\_SALES'

sh@TEST0924> /

OWNER      TABLE\_NAME COLUMN\_NAME     CONSTRAINT\_NAME                CONSTRAINT SEARCH\_CONDITION

---------- ---------- --------------- ------------------------------ ---------- ------------------------------

SH         NEW\_SALES  PROD\_ID         SYS\_C0011428                   C          "PROD\_ID" IS NOT NULL

SH         NEW\_SALES  CUST\_ID         SYS\_C0011429                   C          "CUST\_ID" IS NOT NULL

SH         NEW\_SALES  ORDER\_DATE      SYS\_C0011430                   C          "ORDER\_DATE" IS NOT NULL

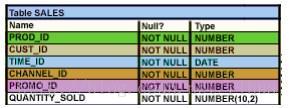
C答案：

由A中所见，此表是可以创建的，虽然字段名不一样，但数据库类型一样，就可以创建表。C错误。

D答案：

由B第二种方法看出，仅仅只是传递了not null约束，而SALES\_PRODUCT\_FK，SALES\_CUSTOMER\_FK，SALES\_TIME\_FK这三个外键约束是没有被传递的。

2. View the Exhibit to examine the description for the SALES table.



Which views can have all DML operations performed on it? (Choose all that apply.)

A. CREATE VIEW v3

AS SELECT \* FROM SALES

WHERE cust\_id = 2034

WITH CHECK OPTION;

B. CREATE VIEW v1

AS SELECT \* FROM SALES

WHERE time\_id <= SYSDATE - 2\*365

WITH CHECK OPTION;

C. CREATE VIEW v2

AS SELECT prod\_id, cust\_id, time\_id FROM SALES

WHERE time\_id <= SYSDATE - 2\*365

WITH CHECK OPTION;

D. CREATE VIEW v4

AS SELECT prod\_id, cust\_id, SUM(quantity\_sold) FROM SALES

WHERE time\_id <= SYSDATE - 2\*365

GROUP BY prod\_id, cust\_id

WITH CHECK OPTION;

Answer: AB

答案解析：

参考：

C不正确的原因为有其它列不能为空

D出错，sum(quantity\_sold)需要加个别名

sh@TESTDB> create view v4

  2  as select prod\_id,cust\_id,sum(quantity\_sold) from sales

  3  where time\_id<=sysdate-2\*365

  4  group by prod\_id,cust\_id

  5  with check option

  6  ;

as select prod\_id,cust\_id,sum(quantity\_sold) from sales

                          \*

ERROR at line 2:

ORA-00998: must name this expression with a column alias

sh@TESTDB> create view v4

  2  as select prod\_id,cust\_id,sum(quantity\_sold) sumqty from sales

  3   where time\_id<=sysdate-2\*365

  4  group by prod\_id,cust\_id

  5  with check option

  6  ;

3. You need to extract details of those products in   the SALES table where the PROD\_ID column

contains the string '\_D123'.

Which WHERE clause could  be used in the SELECT statement to get the required output?

A. WHERE prod\_id LIKE '%\_D123%' ESCAPE '\_'

B. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\'

C. WHERE prod\_id LIKE '%\_D123%' ESCAPE '%\_'

D. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\\_'

Answer: B

答案解析：

ESCAPE子句为指定转译字符，因为‘\_’下滑线在LIKE子句中指的是任意一个字符，所以需要把\_’下滑线进行转译。

4. Which two statements are true regarding single row functions? (Choose two.)

A. They a ccept only a single argument.

B. They c an be nested only to two levels.

C. Arguments can only be column values or constants.

D. They a lways return a single result row for every row of a queried table.

E. They c an return a data type  value different from the one that is referenced.

Answer: DE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

A. 只接受一个参数，错误，可以接受一个或多个参数

B. 只可以被嵌套两级，错误，可以被全套多级

C. 参数只能是列值或者常量.，参数还可以是变量和表达式

D. 对于查询表的每一行总是返回一个结果行.正确。

E. 返回的数据类型值可以与之前引用的数据类型不同，正确。

5. Which SQL statements would display the value 1890.55 as $1,890.55? (Choose three .)

A. SELECT TO\_CHAR(1890.55,'$0G000D00')

FROM DUAL;

B. SELECT TO\_CHAR(1890.55,'$9,999V99')

FROM DUAL;

C. SELECT TO\_CHAR(1890.55,'$99,999D99')

FROM DUAL;

D. SELECT TO\_CHAR(1890.55,'$99G999D00')

FROM DUAL;

E. SELECT TO\_CHAR(1890.55,'$99G999D99')

FROM DUAL;

Answer: ADE

答案解析：

G表示分组分隔符group

D表示小数点decimal point

V表示移动N位（小数点），乘以10的n次方后面跟几位就是几次方

如：

to\_char(12,'99V999') ' 12000'

to\_char(12.4,'99V999') ' 12400'

to\_char(12.45, '99V9') ' 125'

0表示补齐位数。

6. Examine the structure of the SHIPMENTS table:

name           Null       Type

PO\_ID          NOT NULL    NUMBER(3)

PO\_DATE        NOT NULL    DATE

SHIPMENT\_DATE NOT NULL    DATE

SHIPMENT\_MODE             VARCHAR2(30)

SHIPMENT\_COST             NUMBER(8,2)

You want to generate a report that displays the PO\_ID and the penalty amount to be paid if the

SHIPMENT\_DATE is later than one month from the PO\_DATE. The penalty is $20 per day.

Evaluate the following two queries:

SQL> SELECT po\_id, CASE

WHEN MONTHS\_BETWEEN (shipment\_date,po\_date)>1 THEN

TO\_CHAR((shipment\_date - po\_date) \* 20) ELSE 'No Penalty' END PENALTY

FROM shipments;

SQL>SELECT po\_id, DECODE

(MONTHS\_BETWEEN (po\_date,shipment\_date)>1,

TO\_CHAR((shipment\_date - po\_date) \* 20), 'No Penalty') PENALTY

FROM shipments;

Which statement is true regarding the above commands?

A. Both execute successfully and give correct results.

B. Only the first query executes successfully but gives a wrong result.

C. Only the first query executes successfully and gives the  correct result.

D. Only the second query executes successfully but gives a wrong result.

E. Only the second query executes successfully and gives the correct result.

Answer: C

答案解析：

题意要求： 显示PO\_ID和支付的罚款总金额，SHIPMENT\_DATE与PO\_DATE进行比较，SHIPMENT\_DATE如果比PO\_DATE晚一个月，则每天罚款$20。

MONTHS\_BETWEEN(date1, date2): 返回date1和date2两个日期之间间隔几个月，结果可以为正或为负。如果date1晚于date2则结果为正;如果date1早于date2则结果为负;结果的非整数部分也代表月间隔的一部分。

DECODE(expr,search1,result1[,search2,result2……,default])：比较expr与search，如果等于search1则返回result1，如果等于search2则返回result2，依次类推，如果都不等于，如果有default则返回default，否则返回NULL.

ORACLE在比较之前，会自动把expr和每一个search隐式转换成第一个search（search1）的数据类型。自动把返回值转换成第一个result（result1）的数据类型。如果第一个result的数据类型为CHAR或者值是null，则Oracle转换返回值为VARCHAR2.

在DECODE函数中,NULL是相等的，如果expr为空，则Oracle将会返回第一个为NULL的search所对应的result。DECODE列表中的最大表达式个数为255个。

第一个SQL的表达是正确，可以正确执行的。其实我觉得这道题的答案有点出入，MONTHS\_BETWEEN返回的是月数，每天罚款$20，相乘起来，因为一个是天的单位，一个是月的单位，如果题中改成每月罚款20，则才是正确的。

 DECODE 的表达是错误的。

*sh@TEST0924> SELECT po\_id, DECODE (MONTHS\_BETWEEN (po\_date,shipment\_date)>1,*

*2  TO\_CHAR((shipment\_date - po\_date) \* 20), 'No Penalty') PENALTY*

*3  FROM shipments;*

*SELECT po\_id, DECODE (MONTHS\_BETWEEN (po\_date,shipment\_date)>1,*

*\**

*ERROR at line 1:*

*ORA-00907: missing right parenthesis*

7. Which two  statements are true regarding the USING and ON clauses in table joins? (Choose two.)

A. Both USING and ON clauses can be used for equijoins and nonequijoins.

B. A maximum of one pair of columns can be joined between two tables using the ON clause.

C. The  ON clause can be used to join tables on columns that have different names but compatible data

types.

D. The  WHERE clause can be used to apply additional conditions in SELECT statements containing the

ON or the  USING clause.

Answer: CD

答案解析：

A, USING和ON子句可以用于等值连接和非等值连接,USING不能用于非等值连接，ON可以。

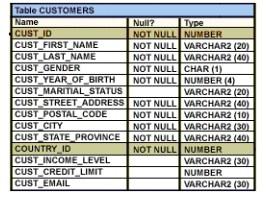
B, 使用ON子句最大只能使用两个列连接两个表.错误，可以连接多个列

C,ON子句用于连接表的列可以是不同的名字，但是数据类型要兼容，正确。ON子句还可以用于联接同一表内或者不同表中具有不同名称的列。

D,在包含ON或USING子句的SELECT命令中，WHERE子句可以做为附加的条件，正确。

8. View the Exhibit and examine the structure of the CUSTOMERS table.

Which two tasks would require subqueries or joins to be executed in a single statement? (Choose two.)



A. listing of customers who do not have a credit limit and were born before 1980

B. finding the number of customers, in each city, whose marital status is 'married'

C. finding the average credit limit of male customers residing in 'Tokyo' or 'Sydney'

D. listing of those customers whose credit limit is the same as the credit limit of customers residing in the

city 'Tokyo'

E. finding the number of customers, in each city, whose credit limit is more than the average credit limit of

all the customers

Answer: DE

答案解析：

ABC不用子查询就可以查询出，题意选择两个，排除ABC,选择DE

A. 列出没有信贷限额并且1980年以前出生的客户。

SELECT 客户 from 表名 where 信贷限额 is null and 出生日期>1980;

B. 查找每个城市的已婚客户的数量。

SELECT 城市名，COUNT(\*) FROM 表名 where 客户婚否=‘结婚’ group by 城市名;

C. 查找属于'Tokyo'或者 'Sydney'两个城市的男性客户的平均信贷限额。

SELECT 城市名， AVG(信贷限额) from 表名 where 性别=‘男’ and 城市 in（'Tokyo'， 'Sydney'） group by 城市名

D 列出与'Tokyo'城市的客户的信贷限额相等的客户

E. 查找每个城市的客户数量，这些客户的信贷限额大于所有客户的平均信贷限额。

9. Which statement is true regarding the INTERSECT operator?

A. It ignores NULL values.

B. Reversing the order of the intersected tables alters the result.

C. The names of columns in all SELECT statements must be identical.

D. The number of columns and data   types must be identical for all SELECT statements in the query.

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12883007>

A. 它忽略空值，错误，不会忽略空值

B. 交换交集表的前后顺序可以改变交集结果，错误，不会改变结果

C. 所有SELECT查询语句中的列的名字必须相同。错误，列名可以不必相同

D. 对于所有SELECT查询语句，列的数量和数据类型必须相同。

使用INTERSECT运算符可以返回多个查询的所有共同行。

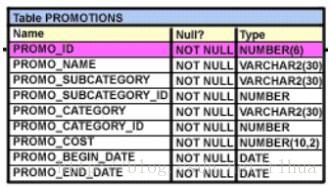
准则

• 在查询中使用的所有SELECT语句中，由查询中的SELECT语句选定的列数和列的数据类型必须相同。不过，列名不必相同。

• 使相交的表按反方向排序不会更改结果。

• INTERSECT不会忽略NULL值。

10. View the Exhibit; e xamine the structure of the PROMOTIONS table.



Each promotion has a duration of at least seven days .

Your manager has asked you to generate a report,  which provides the weekly cost for each promotion

done to l date.

Which query would achieve the required result?

A. SELECT promo\_name, promo\_cost/promo\_end\_date-promo\_begin\_date/7

FROM promotions;

B. SELECT promo\_name,(promo\_cost/promo\_end\_date-promo\_begin\_date)/7

FROM promotions;

C. SELECT promo\_name, promo\_cost/(promo\_end\_date-promo\_begin\_date/7)

FROM promotions;

D. SELECT promo\_name, promo\_cost/((promo\_end\_date-promo\_begin\_date)/7)

FROM promotions;

Answer: D

答案解析：

提要要求:报表提供促销期间每周的成本

D,正确，符合题意。

AB C语法错误

sh@TEST0924> SELECT promo\_name, promo\_cost/promo\_end\_date-promo\_begin\_date/7 FROM promotions;

SELECT promo\_name, promo\_cost/promo\_end\_date-promo\_begin\_date/7 FROM promotions

                              \*

ERROR at line 1:

ORA-00932: inconsistent datatypes: expected NUMBER got DATE

sh@TEST0924> SELECT promo\_name,(promo\_cost/promo\_end\_date-promo\_begin\_date)/7 FROM promotions;

SELECT promo\_name,(promo\_cost/promo\_end\_date-promo\_begin\_date)/7 FROM promotions

                              \*

ERROR at line 1:

ORA-00932: inconsistent datatypes: expected NUMBER got DATE

sh@TEST0924> SELECT promo\_name, promo\_cost/(promo\_end\_date-promo\_begin\_date/7) FROM promotions;

SELECT promo\_name, promo\_cost/(promo\_end\_date-promo\_begin\_date/7) FROM promotions

                                              \*

ERROR at line 1:

ORA-00932: inconsistent datatypes: expected NUMBER got DATE

11. View the Exhibit and examine the structure of the PRODUCTS table.  
All products have a list price.   
You issue the following command to display the total price of each product after a discount of 25% and a

tax of 15% are  applied on it. Freight charges of $100 have to be applied to all the products.  
SQL>SELECT prod\_name, prod\_list\_price -(prod\_list\_price\*(25/100))                   
+(prod\_list\_price -(prod\_list\_price\*(25/100))\*(15/100))+100                                   
AS "TOTAL PRICE"   
FROM products;   
What would be the outcome if all the parenthese s are removed from the above statement?

A. It produces a syntax error.  
B. The result remains unchanged.   
C. The total price value would be lower than the correct value.   
D. The total price value would be higher than the correct value.

**[html]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10913485)

1. **<span** style="font-size:12px;"**>**sh@TESTDB**>** SELECT prod\_name, prod\_list\_price -(prod\_list\_price\*(25/100))
2. 2  +(prod\_list\_price -(prod\_list\_price\*(25/100))\*(15/100))+100
3. 3  AS "TOTAL PRICE"
4. 4  FROM products where rownum**<6**
5. 5  order by "TOTAL PRICE";
7. PROD\_NAME                                          TOTAL PRICE
8. -------------------------------------------------- -----------
9. Y Box                                               613.732875
10. 5MP Telephoto Digital Camera                        1641.23288
11. 17" LCD w/built-in HDTV Tuner                       1812.48288
12. Envoy 256MB - 40GB                                  1812.48288
13. Mini DV Camcorder with 3.5" Swivel LCD              1983.73288**</span>**

去除括号后：

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10913485)

1. <span style="font-size:12px;">sh@TESTDB>  **SELECT** prod\_name, prod\_list\_price -prod\_list\_price\*25/100
2. 2  +prod\_list\_price -prod\_list\_price\*25/100\*15/100+100
3. 3  **AS** "TOTAL PRICE"
4. 4  **FROM** products **where** rownum<6
5. 5  **order** **by** "TOTAL PRICE";
7. PROD\_NAME                                          TOTAL PRICE
8. -------------------------------------------------- -----------
9. Y Box                                               613.732875
10. 5MP Telephoto Digital Camera                        1641.23288
11. 17" LCD w/built-in HDTV Tuner                       1812.48288
12. Envoy 256MB - 40GB                                  1812.48288
13. Mini DV Camcorder **with** 3.5" Swivel LCD              1983.73288</span>

虽然计算顺序不一样，但计算结果是一样的

**此题答案选：B**

12. You need to produce a report where each customer's credit limit has been incremented by $1000. In  
the output, t he customer's last name should have the heading Name and the   incremented credit limit  
should be labeled New Credit Limit. The column headings should have only the first letter of each word in  
uppercase .   
Which statement would accomplish this requirement?   
A.

SELECT cust\_last\_name Name, cust\_credit\_limit + 1000        
"New Credit Limit"   
FROM customers;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10914215)

1. sh@TESTDB> **SELECT** cust\_last\_name **Name**, cust\_credit\_limit + 1000
2. 2  "New Credit Limit"
3. 3  **FROM** customers **WHERE** ROWNUM<6;
5. **NAME**                                               New Credit Limit
6. -------------------------------------------------- ----------------
7. Ruddy                                                          2500
8. Ruddy                                                          8000
9. Ruddy                                                         12000
10. Ruddy                                                          2500
11. Ruddy                                                         10000

B.

SELECT cust\_last\_name AS Name, cust\_credit\_limit + 1000         
AS New Credit Limit   
FROM customers;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10914215)

1. sh@TESTDB> **SELECT** cust\_last\_name **AS** **Name**, cust\_credit\_limit + 1000
2. 2  **AS** New Credit Limit
3. 3  **FROM** customers **WHERE** ROWNUM<6;
4. **AS** New Credit Limit
5. \*
6. ERROR **at** line 2:
7. ORA-00923: **FROM** keyword not found **where** expected

C.

SELECT cust\_last\_name AS "Name", cust\_credit\_limit + 1000         
AS "New Credit Limit"   
FROM customers;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10914215)

1. <a target="\_blank" href="mailto:sh@TESTDB">sh@TESTDB</a>> **SELECT** cust\_last\_name **AS** "Name", cust\_credit\_limit + 1000
2. 2  **AS** "New Credit Limit"
3. 3  **FROM** customers **WHERE** ROWNUM<6;
5. **Name**                                               New Credit Limit
6. -------------------------------------------------- ----------------
7. Ruddy                                                          2500
8. Ruddy                                                          8000
9. Ruddy                                                         12000
10. Ruddy                                                          2500
11. Ruddy                                                         10000

D.

SELECT INITCAP(cust\_last\_name) "Name", cust\_credit\_limit + 1000         
INITCAP("NEW CREDIT LIMIT")   
FROM customers;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10914215)

1. sh@TESTDB> **SELECT** INITCAP(cust\_last\_name) "Name", cust\_credit\_limit + 1000
2. 2  INITCAP("NEW CREDIT LIMIT")
3. 3  **FROM** customers **WHERE** ROWNUM<6;
4. INITCAP("NEW CREDIT LIMIT")
5. \*
6. ERROR **at** line 2:
7. ORA-00923: **FROM** keyword not found **where** expected

此题要求选出列标题为Name和New Credit Limit，只有C满足。

A的Name应加双引号，原型输出

B/D语法错误  
  
**此题答案选C**

13. View the Exhibit and examine the structure of the PRODUCTS table.

You need to generate a report in the following format:

CATEGORIES

5MP Digital Photo Camera's category is Photo

Y Box's category is Electronics

Envoy Ambassador's category is Hardware

Which two queries would give the required output? (Choose two.)

A. SELECT prod\_name  q'''s category is '  prod\_category CATEGORIES

FROM products;

B. SELECT prod\_name  q'['s ]'category is '  prod\_category CATEGORIES

FROM products;

C. SELECT prod\_name  q'\'s\'   ' category is '  prod\_category CATEGORIES

FROM products;

D. SELECT prod\_name  q'<'s >'   'category is '  prod\_category CATEGORIES

FROM products;

Answer: CD

答案解析：

在SQL字符串是以[单引号](http://zhidao.baidu.com/search?word=%E5%8D%95%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)作为分界符的，在字符串前面和后面各一个[单引号](http://zhidao.baidu.com/search?word=%E5%8D%95%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)。但是字符串中也能包含[单引号](http://zhidao.baidu.com/search?word=%E5%8D%95%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)，为了使语法分析器能够区分字符串中的单引号还是分界符。规定当字符串中出现单引号时，在其前面添加一个单引号作为区分。也就是说， 在单引号分隔的字符串中，两个连续的单引号''表示一个单引号字符。

sh@TEST1107> select '''' from dual;

'

-

'

四个连续单引号分界为：   
第一个单引号，字符串的开始分界符。  
接下来连续的两个单引号，表示字符串的值（一个单引号）  
第四个单引号，字符串的结束分界符。

sh@TEST1107> select 'today is ''FRIDAY''' from dual;

'TODAYIS''FRIDAY'

-----------------

today is 'FRIDAY'

在Oracle中，single-quote(‘)是一个表示字符串的关键字。所以在字符串中用两个”表示一个实际的单引号字符。所有才会有了上面第一条SQL的'today is ''FRIDAY'''。双引号“被识别为一个实际的的字符串。其实Oracle提供了一个Q-quote的表达式，用来简化SQL或PLSQL中字符串的表示。

sh@TEST1107>  select q'[today is ''FRIDAY'']' names from dual;

NAMES

-------------------

today is ''FRIDAY''

sh@TEST1107> select q'[today is 'FRIDAY']' names from dual;

NAMES

-----------------

today is 'FRIDAY'

看上面两个的差别可知，使用Q-quote的表达式，只需要将要表示的字符串用一对特殊字符括起来，这对字符必须一致。不一致会报错。

sh@TEST1107> select q'[today is 'FRIDAY'|' names from dual;

ERROR:

ORA-01756: quoted string not properly terminated

sh@TEST1107> select q'|today is 'FRIDAY'|' names from dual;

NAMES

-----------------

today is 'FRIDAY'

从以上实验来解答题中的答案。

A答案错误，使用‘作为特殊字符，is后面缺少单引号，应该为

sh@TEST1107> SELECT prod\_name || q'''s category is ''|| prod\_category CATEGORIES FROM products;

CATEGORIES

----------------------------------------------------------------------------------------------------

5MP Telephoto Digital Camera's category is Photo

17" LCD w/built-in HDTV Tuner's category is Peripherals and Accessories

Envoy 256MB - 40GB's category is Hardware

Y Box's category is Electronics

Mini DV Camcorder with 3.5" Swivel LCD's category is Photo

B. 错误。SELECT prod\_name  q'['s ]'category is '  prod\_category CATEGORIES FROM products;

category is ' 这里前面缺少一个单引号。缺少category is 开始的单引号。

应该为：

sh@TEST1107> SELECT prod\_name || q'['s ]'||'category is '||prod\_category CATEGORIES FROM products;

CATEGORIES

----------------------------------------------------------------------------------------------------

5MP Telephoto Digital Camera's category is Photo

17" LCD w/built-in HDTV Tuner's category is Peripherals and Accessories

Envoy 256MB - 40GB's category is Hardware

Y Box's category is Electronics

C. 正确SELECT prod\_name  q'\'s\'   ' category is '  prod\_category CATEGORIES FROM products;

sh@TEST1107> SELECT prod\_name||  q'\'s\' ||  ' category is ' || prod\_category CATEGORIES FROM products;

CATEGORIES

----------------------------------------------------------------------------------------------------

5MP Telephoto Digital Camera's category is Photo

17" LCD w/built-in HDTV Tuner's category is Peripherals and Accessories

Envoy 256MB - 40GB's category is Hardware

Y Box's category is Electronics

D. 正确SELECT prod\_name  q'<'s >'   'category is '  prod\_category CATEGORIES FROM products;

sh@TEST1107> SELECT prod\_name || q'<'s >'  || 'category is ' || prod\_category CATEGORIES FROM products;

CATEGORIES

----------------------------------------------------------------------------------------------------

5MP Telephoto Digital Camera's category is Photo

17" LCD w/built-in HDTV Tuner's category is Peripherals and Accessories

Envoy 256MB - 40GB's category is Hardware

Y Box's category is Electronics

**注意，题中答案需要添加||来连接字符串。**

14. Using the CUSTOMERS table,  you need to generate a report that shows 50% of each   credit   
amount in each income level. The report should NOT show any repeated credit amounts in each income   
level.  **--each income  level中没有重复的credit amounts**Which  query would give the required result?   
A.

SELECT  cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50      
AS "50% Credit Limit"   
FROM customers;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10915459)

1. sh@TESTDB> **SELECT**  cust\_income\_level, **DISTINCT** cust\_credit\_limit \* 0.50
2. 2  **AS** "50% Credit Limit"
3. 3  **FROM** customers;
4. **SELECT**  cust\_income\_level, **DISTINCT** cust\_credit\_limit \* 0.50
5. \*
6. ERROR **at** line 1:
7. ORA-00936: missing expression

语法错误，DISTINCT不应该放在第二列，应该放在第一列中

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10915459)

1. sh@TESTDB> **SELECT** **DISTINCT**  cust\_income\_level,cust\_credit\_limit \* 0.50
2. 2  **AS** "50% Credit Limit"
3. 3  **FROM** customers **where** rownum<6;
5. CUST\_INCOME\_LEVEL              50% Credit Limit
6. ------------------------------ ----------------
7. G: 130,000 - 149,999                       5500
8. G: 130,000 - 149,999                       4500
9. G: 130,000 - 149,999                       3500
10. G: 130,000 - 149,999                        750

B.

SELECT DISTINCT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50      
AS "50% Credit Limit"   
FROM customers;

同A，语法错误，DISTINCT不应该放在第二个

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10915459)

1. **SELECT** **DISTINCT** cust\_income\_level, **DISTINCT** cust\_credit\_limit \* 0.50
2. 2  **AS** "50% Credit Limit"
3. 3  **FROM** customers;
4. **SELECT** **DISTINCT** cust\_income\_level, **DISTINCT** cust\_credit\_limit \* 0.50
5. \*
6. ERROR **at** line 1:
7. ORA-00936: missing expression

C.

SELECT DISTINCT cust\_income\_level   '   '  cust\_credit\_limit \* 0.50   
AS "50% Credit Limit"   
FROM customers;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10915459)

1. sh@TESTDB> **SELECT** **DISTINCT** cust\_income\_level   '   '  cust\_credit\_limit \* 0.50
2. 2  **AS** "50% Credit Limit"
3. 3  **FROM** customers;
4. **SELECT** **DISTINCT** cust\_income\_level   '   '  cust\_credit\_limit \* 0.50
5. \*
6. ERROR **at** line 1:
7. ORA-00923: **FROM** keyword not found **where** expected

这道题愿意为用' '连接两个列，可能用错，应为||来连接，不过DISTINCT的位置 对的

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10915459)

1. sh@TESTDB> **SELECT** **DISTINCT** cust\_income\_level || cust\_credit\_limit \* 0.50
2. 2  **AS** "50% Credit Limit"
3. 3  **FROM** customers **where** rownum<6;
5. 50% Credit Limit
6. ----------------------------------------------------------------------
7. G: 130,000 - 149,999750
8. G: 130,000 - 149,9994500
9. G: 130,000 - 149,9993500
10. G: 130,000 - 149,9995500

D.

SELECT cust\_income\_level ' ' cust\_credit\_limit \* 0.50 AS "50% Credit Limit"   
FROM customers

D没有满足NOT show any repeated的条件。

**此题答案C**

15. View the Exhibit and examine the data in the CUSTOMERS table.    
Evaluate the following query:   
SQL> SELECT cust\_name AS "NAME", cust\_credit\_limit/2 AS MIDPOINT,MIDPOINT+100 AS "MAX   
LOWER LIMIT"   
FROM customers;   
The above query produces an error on execution.   
What is the reason for the error?

A. An alias cannot be used in an expression.   
B. The a lias NAME should not be enclosed with in double quotation marks .   
C. The MIDPOINT+100 expression  gives an error because  CUST\_CREDIT\_LIMIT contains NULL   
values.   
D.  The  a  lias  MIDPOINT  should  be  enclosed  with  in  double  quotation  marks   for  the   
CUST\_CREDIT\_LIMIT/2 expression .

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10916597)

1. sh@TESTDB> **SELECT** cust\_name **AS** "NAME", cust\_credit\_limit/2 **AS** MIDPOINT,MIDPOINT+100 **AS** "**MAX**
2. 2  **LOWER** LIMIT"
3. 3  **FROM** customers;
4. **SELECT** cust\_name **AS** "NAME", cust\_credit\_limit/2 **AS** MIDPOINT,MIDPOINT+100 **AS** "**MAX**
5. \*
6. ERROR **at** line 1:
7. ORA-00904: "MIDPOINT": invalid identifier

报错，select中不应该用别名来用在表达式里

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10916597)

1. sh@TESTDB> **SELECT** cust\_last\_name **AS** "NAME", cust\_credit\_limit/2 **AS** MIDPOINT,cust\_credit\_limit/2+100 **AS** "MAX LOWER LIMIT"
2. 2  **FROM** customers **WHERE** ROWNUM<6;
4. **NAME**                                                 MIDPOINT **MAX** **LOWER** LIMIT
5. -------------------------------------------------- ---------- ---------------
6. Ruddy                                                     750             850
7. Ruddy                                                    3500            3600
8. Ruddy                                                    5500            5600
9. Ruddy                                                     750             850
10. Ruddy                                                    4500            4600

**此题答案选A**

16. Evaluate the following query:

SQL> SELECT promo\_name  q'{'s start date was }'  promo\_begin\_date

AS "Promotion Launches"

FROM promotions;

What would be the outcome of the above query?

A. It produces an error because flower braces have been used.

B. It produces an error because the data   types are not matching.

C. It executes successfully and introduces an   's at the end of each promo\_name in the output.

D. It executes successfully and displays the literal " {'s start date was } " for each row in the output.

Answer: C

答案解析：

参考13题：<http://blog.csdn.net/rlhua/article/details/16336797>

实验验证：

sh@TEST1107> SELECT promo\_name|| q'{'s start date was }'|| promo\_begin\_date  AS "Promotion Launches" FROM promotions;

Promotion Launches

------------------------------------------------------------------

NO PROMOTION #'s start date was 01-JAN-99

newspaper promotion #16-108's start date was 23-DEC-00

post promotion #20-232's start date was 25-SEP-98

newspaper promotion #16-349's start date was 10-JUL-98

internet promotion #14-471's start date was 26-FEB-00

TV promotion #13-448's start date was 06-AUG-00

17. View the E xhibit and examine the data in the EMPLOYEES table.   
You want to generate a report showing the total compensation paid to each employee to  date.  
You issue the following query:   
SQL>SELECT ename ' joined on ' hiredate        
', the total compensation paid is '       
TO\_CHAR(ROUND(ROUND(SYSDATE-hiredate)/365) \* sal + comm)       
"COMPENSATION UNTIL DATE"       
FROM employees;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10917281)

1. sys@TESTDB> **select** \* **from** scott.emp;
3. EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
4. ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
5. 7369 SMITH      CLERK           7902 17-**DEC**-80        800                    20
6. 7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
7. 7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
8. 7566 JONES      MANAGER         7839 02-APR-81       2975                    20
9. 7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
10. 7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30

What is the outcome?

A. It generates an error because the alias is not valid.   
B. It executes successfully and gives the correct output.   
C. It executes successfully but does not give the correct output.   
D. It generates an error because the usage of the  ROUND function in the expression is not valid.  
E. It generates an error because the concatenation operator can be used to combine only two items.

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10917281)

1. sys@TESTDB> **select** TO\_CHAR(ROUND(ROUND(SYSDATE-hiredate)/365) \* sal + comm) "COMPENSATION UNTIL DATE"
2. 2  **from** scott.emp;
4. COMPENSATION UNTIL **DATE**
5. ----------------------------------------
7. 53100
8. 41750
10. 41400



15. 48000




21. 14 **rows** selected.

因为comm有null值，故任何数值与nll相加为null，故出错。

**此题答案选C**

18. Examine the structure of the PROMOTIONS table:   
name            Null       Type   
PROMO\_ID        NOT NULL    NUMBER(6)   
PROMO\_NAME      NOT NULL    VARCHAR2(30)   
PROMO\_CATEGORY NOT NULL    VARCHAR2(30)   
PROMO\_COST      NOT NULL    NUMBER(10,2)   
The management wants to see a report of unique promotion costs in each promotion category.   
Which query would achieve the required result?   
A. SELECT DISTINCT promo\_cost, promo\_category FROM promotions;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10918633)

1. sh@TESTDB> **SELECT** **DISTINCT** promo\_cost, promo\_category **FROM** promotions **where** rownum<6;
3. PROMO\_COST PROMO\_CATEGORY
4. ---------- ------------------------------
5. 0 **NO** PROMOTION
6. 200 newspaper
7. 300 post
8. 600 internet
9. 400 newspaper

B. SELECT promo\_category, DISTINCT promo\_cost FROM promotions;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10918633)

1. sh@TESTDB> **SELECT** promo\_category, **DISTINCT** promo\_cost **FROM** promotions
2. 2  ;
3. **SELECT** promo\_category, **DISTINCT** promo\_cost **FROM** promotions
4. \*
5. ERROR **at** line 1:
6. ORA-00936: missing expression

C. SELECT DISTINCT promo\_cost, DISTINCT promo\_category FROM promotions;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10918633)

1. sh@TESTDB> **SELECT** **DISTINCT** promo\_cost, **DISTINCT** promo\_category **FROM** promotions;
2. **SELECT** **DISTINCT** promo\_cost, **DISTINCT** promo\_category **FROM** promotions
3. \*
4. ERROR **at** line 1:
5. ORA-00936: missing expression

D. SELECT DISTINCT promo\_category, promo\_cost FROM promotions ORDER BY 1;

**[sql]** [view plaincopyprint?](http://blog.csdn.net/rlhua/article/details/10918633)

1. sh@TESTDB> **SELECT** **DISTINCT** promo\_category, promo\_cost **FROM** promotions **where** rownum<6 **order** **by** 1;
3. PROMO\_CATEGORY                 PROMO\_COST
4. ------------------------------ ----------
5. **NO** PROMOTION                            0
6. internet                              600
7. newspaper                             200
8. newspaper                             400
9. post                                  300

unique promotion costs in each promotion category

promotion category前加DISTINCT

A与描述的不一致

BC语法错误，DISTINCT放在第二列要报错。

**此题答案选D**

19. Evaluate the following query:

SELECT INTERVAL '300' MONTH,

INTERVAL '54-2' YEAR TO MONTH,

INTERVAL '11:12:10.1234567' HOUR TO SECOND

FROM dual;

What is the correct output of the above query?

A. +25-00 , +54-02, +00 11:12:10.123457

B. +00-300, +54-02, +00 11:12:10.123457

C. +25-00 , +00-650, +00 11:12:10.123457

D. +00-300 , +00-650, +00 11:12:10.123457

Answer: A

答案解析：

实验验证：

sys@TEST0924> SELECT INTERVAL '300' MONTH, INTERVAL '54-2' YEAR TO MONTH,INTERVAL '11:12:10.1234567' HOUR TO SECOND FROM dual;

INTERVAL'300'MONTH

---------------------------------------------------------------------------

INTERVAL'54-2'YEARTOMONTH

---------------------------------------------------------------------------

INTERVAL'11:12:10.1234567'HOURTOSECOND

---------------------------------------------------------------------------

+25-00

+54-02

+00 11:12:10.123457

INTERVAL '300' MONTH：300个月为+25-00（25年-0个月，即25\*12+0）

INTERVAL '54-2' YEAR TO MONTH：+54-02

INTERVAL '11:12:10.1234567' HOUR TO SECOND: +00 11:12:10.123457（+00表示0天）

20. Which three statements are true regarding the data types in Oracle Database 10g/11g? (Choose

three.)

A. Only one LONG column can be used per table.

B. A TIMESTAMP data type column stores only time values with fractional seconds.

C. The BLOB data type column is used to store binary data in an operating system file.

D. The minimum column width that can be specified for a VARCHAR2 data type column is one.

E. The value for a CHAR data type column is blank-padded to the maximum defined column width.

Answer: ADE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

准则

• 在使用子查询创建表时不复制LONG列。

• 不能在GROUP BY或ORDER BY子句中包括LONG列。

• 每个表只能使用一个LONG列。

• 不能对LONG列定义约束条件。

• 可以要求使用CLOB列，而不是LONG列。

A. 每个表中只能使用一个LONG列。正确

B. 一个TIMESTAMP数据类型列只能存储带有小数秒的时间值。错误，也可以存储不带小数秒的日期或时间

C. BLOB数据类型列被用于存储二进制数据的操作系统文件。错误，不是操作系统文件。

D. 对于VARCHAR2数据类型最小的列宽可以为1（VARCHAR2为可变长度的数据类型）.

E. CHAR数据类型列的值是用空格填充到列定义的最大值（也就是说，如果定义了CHAR(20),而字符串长度不够20个，则用空格填充不够的位数，CHAR为固定长度数据类型）

21. Examine the description of the EMP\_DETAILS table given below:

name               NULL                 TYPE

EMP\_ID             NOT NULL             NUMBER

EMP\_NAME           NOT NULL             VARCHAR2 (40)

EMP\_IMAGE                              LONG

Which two statements are true regarding SQL statements that can be executed on the EMP\_DETAIL

table? (Choose two.)

A. An EMP\_IMAGE column can be included in the GROUP BY clause.

B. An EMP\_IMAGE column cannot be included in the ORDER BY clause.

C. You cannot add a  new column to the table with LONG as the  data type.

D. You can alter the table to include the  NOT NULL constraint on the EMP\_IMAGE column.

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

准则

• 在使用子查询创建表时不复制LONG列。

• 不能在GROUP BY或ORDER BY子句中包括LONG列。

• 每个表只能使用一个LONG列。

• 不能对LONG列定义约束条件。

• 可以要求使用CLOB列，而不是LONG列。

22. You need to create a table for a banking application. One of the columns in the table has the following

requirements:

1) You want a column in the table to store the duration of the credit period.

2) The data in the column should be stored in a format such that it can be easily added and subtracted

with

DATE data type without using conversion functions.

3) The maximum period of the credit provision in the application is 30 days.

4) The interest has to be calculated for the number of days an individual has taken a credit for.

Which data type would you use for such a column in the table?

A. DATE

B. NUMBER

C. TIMESTAMP

D. INTERVAL DAY TO SECOND

E. INTERVAL YEAR TO MONTH

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

题意要求

1) 你想让该列存储信贷周期的持续时间

2) 该列的日期格式如下：它能非常容易的与DATE数据类型进行加减，无需使用转换函数

3) 应用程序中的最大信贷条款周期是30天

4) 已经使用信用卡的个人按天计算利息

A和C都是存储的时间点，不是一段时间，而B是数值型数据不能满足条件，E不正确因为虽然它也是存储一段时间但是，是从年到月的，不能满足第3条和第4条件需要精确到天

23. Examine the structure proposed for the TRANSACTIONS table:

name                    Null      Type

TRANS\_ID               NOT NULL NUMBER(6)

CUST\_NAME              NOT NULL VARCHAR2(20)

CUST\_STATUS            NOT NULL CHAR

TRANS\_DATE             NOT NULL DATE

TRANS\_VALIDITY                  VARCHAR2

CUST\_CREDIT\_LIMIT               NUMBER

Which statements are true regarding the creation and storage of data in the above table structure?

(Choose all that apply.)

A. The CUST\_STATUS column would give an error.

B. The TRANS\_VALIDITY column would give an error.

C. The CUST\_STATUS column would store exactly one character.

D. The CUST\_CREDIT\_LIMIT column would not be able to store decimal values.

E. The TRANS\_VALIDITY column would have a maximum size of one character.

F. The TRANS\_DATE column would be able to store day, month, century, year, hour, minutes, seconds,

and fractions of seconds.

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

VARCHAR2长度可变的字符数据（必须指定最大size，最小size为1；最大size为4000）

TRANS\_VALIDITY  的VARCHAR2没有指定size，所以错误。B正确。E错误。

CHAR 固定长度的字符数据，长度为size字节（默认和最小size为1；最大size为2000），可以不用指定size

所以CUST\_STATUS的CHAR，没有指定size，但默认size为1，不会报错，所以A是错误的。C是正确的。

NUMBER [(p，s)]精度为p和小数位为s的数字（精度是十进制数字的总位数，而小数位数是小数点右侧的位数，精度范围在1~38之间，而该小数位数范围在-84~127之间），

所以CUST\_CREDIT\_LIMIT 的NUMBER数据类型是可以存储小数位的，D错误。

 DATE：公元前4712年1月1日到公元9999年12月31日之间的日期和时间值，精确到最接近的秒。但是没有小数秒，所以F错误。

24. Examine the structure proposed for the TRANSACTIONS table:

name                    Null      Type

TRANS\_ID               NOT NULL NUMBER(6)

CUST\_NAME              NOT NULL VARCHAR2(20)

CUST\_STATUS            NOT NULL VARCHAR2

TRANS\_DATE             NOT NULL DATE

TRANS\_VALIDITY                  INTERVAL DAY TO SECOND

CUST\_CREDIT\_VALUE               NUMBER(10)

Which two statements are true regarding the storage of data in the above table structure? (Choose two.)

A. The TRANS\_DATE column would allow storage of dates only in the dd-mon-yyyy format.

B. The CUST\_CREDIT\_VALUE column would allow storage of positive and negative integers.

C. The TRANS\_VALIDITY column would allow storage of a time interval in days, hours, minutes, and

seconds.

D. The CUST\_STATUS column would allow storage of data up to the maximum VARCHAR2 size of 4,000

characters.

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

VARCHAR2长度可变的字符数据（必须指定最大size，最小size为1；最大size为4000）

CUST\_STATUS  的VARCHAR2没有指定size，所以错误。D错误。

INTERVAL DAY TO SECOND：可以将时间存储为以天数、时数、分钟数和秒数表示的间隔。用于表示两个日期时间值之间的精确差。

所以C正确。

NUMBER [(p，s)]精度为p和小数位为s的数字（精度是十进制数字的总位数，而小数位数是小数点右侧的位数，精度范围在1~38之间，而该小数位数范围在-84~127之间），即可以存储正负数。

所以CUST\_CREDIT\_VALUE 的NUMBER数据类型是可以存储正负数的，B正确。

 DATE：公元前4712年1月1日到公元9999年12月31日之间的日期和时间值，精确到最接近的秒。可以使用to\_date转换成不同的时间格式，所以A错误。

25. You need to create a table with the following column specifications:

1. Employee ID (numeric data type) for each employee

2. Employee Name (character data type) that stores the employee name

3. Hire date, which stores the date of joining the organization for each employee

4. Status (character data type), that contains the value 'ACTIVE' if no data is entered

5. Resume (character large object [CLOB] data type), which contains the resume submitted by the

employee

Which is the correct syntax to create this table?

A. CREATE TABLE EMP\_1

(emp\_id NUMBER(4),

emp\_name VARCHAR2(25),

start\_date DATE,

e\_status VARCHAR2(10) DEFAULT 'ACTIVE',

resume CLOB(200));

B. CREATE TABLE 1\_EMP

(emp\_id NUMBER(4),

emp\_name VARCHAR2(25),

start\_date DATE,

emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',

resume CLOB);

C. CREATE TABLE EMP\_1

(emp\_id NUMBER(4),

emp\_name VARCHAR2(25),

start\_date DATE,

emp\_status VARCHAR2(10) DEFAULT "ACTIVE",

resume CLOB);

D. CREATE TABLE EMP\_1

(emp\_id  NUMBER,

emp\_name VARCHAR2(25),

start\_date DATE,

emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',

resume CLOB);

Answer: D

答案解析：

A答案：CLOB不需要跟size。

sh@TEST0924> CREATE TABLE EMP\_1

  2  (resume CLOB(200));

(resume CLOB(200))

            \*

ERROR at line 2:

ORA-00907: missing right parenthesis

sh@TEST0924> CREATE TABLE EMP\_1

  2  (resume CLOB);

Table created.

B答案：表名不以数字开头，或者加上双引号，或者只能以字母开头

sh@TEST0924> CREATE TABLE 1\_EMP

  2  (emp\_id NUMBER(4));

CREATE TABLE 1\_EMP

             \*

ERROR at line 1:

ORA-00903: invalid table name

sh@TEST0924> CREATE TABLE "1\_EMP"

  2  (emp\_id NUMBER(4));

Table created.

C答案：字符默认值不能用双引号，需要使用单引号。

sh@TEST0924> CREATE TABLE EMP\_2

  2  (emp\_status VARCHAR2(10) DEFAULT "ACTIVE");

(emp\_status VARCHAR2(10) DEFAULT "ACTIVE")

                                 \*

ERROR at line 2:

ORA-00984: column not allowed here

sh@TEST0924> CREATE TABLE EMP\_2

  2  (emp\_status VARCHAR2(10) DEFAULT 'ACTIVE');

Table created.

D答案：D正确，可成功创建表。

sh@TEST0924> CREATE TABLE EMP\_3

  2  (emp\_id  NUMBER,

  3  emp\_name VARCHAR2(25),

  4  start\_date DATE,

  5  emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',

  6  resume CLOB);

Table created.

26. Which is the valid CREATE TABLE statement?

A. CREATE TABLE  emp9$#  (emp\_no NUMBER (4));

B. CREATE TABLE  9emp$#  (emp\_no NUMBER(4));

C. CREATE TABLE  emp\*123  (emp\_no NUMBER(4));

D. CREATE TABLE  emp9$#  (emp\_no NUMBER(4), date DATE);

Answer: A

答案解析：

A答案

sys@TESTDB> CREATE TABLE  emp9$#  (emp\_no NUMBER (4));

Table created.

 B答案，表名不能以数字开头，只能以字母开头

sys@TESTDB> CREATE TABLE  9emp$#  (emp\_no NUMBER(4));

CREATE TABLE  9emp$#  (emp\_no NUMBER(4))

              \*

ERROR at line 1:

ORA-00903: invalid table name

C答案，表名只能有字母，数字、下划线\_、美元符$、井字号#等，没有星号\*号

sys@TESTDB> CREATE TABLE  emp\*123  (emp\_no NUMBER(4));

CREATE TABLE  emp\*123  (emp\_no NUMBER(4))

                 \*

ERROR at line 1:

ORA-00922: missing or invalid option

D列名，不能用预留字作为列名。

sys@TESTDB> CREATE TABLE  emp9$#  (emp\_no NUMBER(4), date DATE);

CREATE TABLE  emp9$#  (emp\_no NUMBER(4), date DATE)

                                         \*

ERROR at line 1:

ORA-00904: : invalid identifier

27. Which two statements are true regarding tables? (Choose two.)

A. A table name can be of any length.

B. A table can have any number of columns.

C. A column that has a DEFAULT value cannot store null values.

D. A table and a view can have the same name in the same schema.

E. A table and a synonym can have the same name in the same schema.

F. The same table name can be used in different schemas in the same database.

Answer: EF

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

A不正确因为表名的长度为1-30个字符

B不正确因为表的列最多为1000个

C不正确因为有DEFAULT值的列可以存储空值

D不正确因为在相同的schema里，表和view的名字不能相同

E正确，表和同义词在相同的schema里名称可以相同，

注意：http://blog.csdn.net/rlhua/article/details/12781353，”• 私用同义词名一定不能与同一用户拥有的其它所有对象的名称相同。公用同义词是允许所有的用户来使用的。

F正确，在同一个数据库里，在不同的schema里，可以有相同的名称。

题中要求选择两个，故只能选择EF

28. Which two statements are true regarding constraints? (Choose two.)

A. A foreign key cannot contain NULL values.

B. A column with the UNIQUE constraint can contain NULL values.

C. A constraint is enforced only for the INSERT operation on a table.

D. A constraint can be disabled even if the constraint column contains data.

E. All constraints can be defined at the column level as well as the table level.

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

A不对因为外键可以包含空值

C不对因为约束可以作用在INSERT,UPDATE,DELETE操作上

E不对因为NOT NULL约束只能在列级定义，不能在表级定义

B正确，唯一约束可以包含空值

D正确，即使约束列上有数据，也可以让约束不可用。

29. Which two statements are true regarding constraints? (Choose two.)

A. A foreign key cannot contain NULL values.

B. The column with a  UNIQUE  constraint can store NULLS .

C. A constraint is enforced only for an INSERT operation on a table.

D. You can have more than one column in a table as part of a primary key.

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

A不对因为外键可以包含空值

C不对因为约束可以作用在INSERT,UPDATE,DELETE操作上

B正确，唯一约束可以包含空值。

D正确，可以多个组合列组成一个表的主键

30. Evaluate the following CREATE TABLE commands:

CREATE TABLE orders

(ord\_no NUMBER(2) CONSTRAINT ord\_pk PRIMARY KEY,

ord\_date DATE,

cust\_id NUMBER(4));

CREATE TABLE ord\_items

(ord\_no NUMBER(2),

item\_no NUMBER(3),

qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200),

expiry\_date date CHECK (expiry\_date > SYSDATE), 系统时间是变化的，不能作为约束

CONSTRAINT it\_pk PRIMARY KEY (ord\_no,item\_no),

CONSTRAINT ord\_fk FOREIGN KEY(ord\_no) REFERENCES orders(ord\_no));

The above command fails when executed. What could be the reason?

A. SYSDATE cannot be used with the CHECK constraint.

B. The BETWEEN clause cannot be used for the CHECK constraint.

C. The CHECK constraint cannot be placed on columns having the DATE data type.

D. ORD\_NO and ITEM\_NO cannot be used as a composite primary key because ORD\_NO is also the

FOREIGN KEY.

Answer: A

答案解析：

sys@TESTDB>  CREATE TABLE orders

  2  (ord\_no NUMBER(2) CONSTRAINT ord\_pk1  PRIMARY KEY,

  3  ord\_date DATE,

  4  cust\_id NUMBER(4));

Table created.

sys@TESTDB> CREATE TABLE ord\_items

  2  (ord\_no NUMBER(2),

  3  item\_no NUMBER(3),

  4  qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200),

  5  expiry\_date date CHECK (expiry\_date > SYSDATE),

  6  CONSTRAINT it\_pk PRIMARY KEY (ord\_no,item\_no),

  7  CONSTRAINT ord\_fk FOREIGN KEY(ord\_no) REFERENCES orders(ord\_no))

  8  ;

expiry\_date date CHECK (expiry\_date > SYSDATE),

                                      \*

ERROR at line 5:

ORA-02436: date or system variable wrongly specified in CHECK constraint

A答案：sysdate不能用在check约束中，A正确

sys@TESTDB> CREATE TABLE ord\_items2

  2  (expiry\_date date CHECK (expiry\_date > SYSDATE));

(expiry\_date date CHECK (expiry\_date > SYSDATE))

                                       \*

ERROR at line 2:

ORA-02436: date or system variable wrongly specified in CHECK constraint

 B答案：BETWEEN 可以用在check约束中，B错误

sys@TESTDB> CREATE TABLE ord\_items1

  2  (qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200));

Table created.

C答案：check约束可以作用在data类型的列上，C错误。

sys@TESTDB>  CREATE TABLE ord\_items3

  2  (expiry\_date date CHECK (expiry\_date >to\_date('2013-09-03','yyyy-mm-dd')));

Table created.

D答案：外键约束和字段可以和本表组合成主键

sh@TESTDB> create table  orders

  2  (ord\_no number primary key);

Table created.

sh@TESTDB> create table ord\_items

  2  (ord\_no number not null,

  3  item\_no number,

  4  constraint ord\_fk foreign key(ord\_no) references orders(ord\_no),

  5  constraint it\_pk primary key(ord\_no,item\_no));

Table created.

31. Evaluate the following SQL commands:

SQL>CREATE SEQUENCE ord\_seq

INCREMENT BY 10

START WITH 120

MAXVALUE 9999

NOCYCLE;

SQL>CREATE TABLE ord\_items

(ord\_no NUMBER(4)DEFAULT ord\_seq.NEXTVAL NOT NULL,

item\_no NUMBER(3),

qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200),

expiry\_date date CHECK (expiry\_date > SYSDATE),

CONSTRAINT it\_pk PRIMARY KEY (ord\_no,item\_no),

CONSTRAINT ord\_fk FOREIGN KEY(ord\_no) REFERENCES orders(ord\_no));

The command to create a table fails. Identify the reason for the SQL statement failure? (Choose all that

apply.)

A. You cannot use SYSDATE in the condition of a CHECK constraint.

B. You cannot use the BETWEEN clause in the condition of a CHECK constraint.

C. You cannot use the NEXTVAL sequence value as a DEFAULT value for a column.

D. You cannot use ORD\_NO and ITEM\_NO columns as a composite primary key because ORD\_NO is

also the FOREIGN KEY.

Answer: AC

答案解析：

scott@TESTDB> create sequence ord\_seq

  2  increment by 10

  3  start with 120

  4  maxvalue 9999

  5  nocycle;

Sequence created.

scott@TESTDB>  CREATE TABLE ord\_items

  2  (expiry\_date date CHECK (expiry\_date > SYSDATE));

(expiry\_date date CHECK (expiry\_date > SYSDATE))

                                       \*

ERROR at line 2:

ORA-02436: date or system variable wrongly specified in CHECK constraint

A 、SYSDATE不能用在CHECK约束中，对

scott@TESTDB>  CREATE TABLE ord\_items

  2  (qty NUMBER(3) CHECK (qty BETWEEN 100 AND 200));

Table created.

B、BETWEEN可以用在约束中，B错

scott@TESTDB>  CREATE TABLE ord\_items

  2  (ord\_no NUMBER(4)DEFAULT ord\_seq.NEXTVAL NOT NULL);

(ord\_no NUMBER(4)DEFAULT ord\_seq.NEXTVAL NOT NULL)

                         \*

ERROR at line 2:

ORA-00984: column not allowed here

C、NEXTVAL不能用作默认值，C对

sh@TESTDB> create table  orders

  2  (ord\_no number primary key);

Table created.

sh@TESTDB> create table ord\_items

  2  (ord\_no number not null,

  3  item\_no number,

  4  constraint ord\_fk foreign key(ord\_no) references orders(ord\_no),

  5  constraint it\_pk primary key(ord\_no,item\_no));

Table created.

D、外键约束和字段可以和本表组合成主键，D错。

32. Which CREATE TABLE statement is valid?

A. CREATE TABLE ord\_details

(ord\_no NUMBER(2) PRIMARY KEY,

item\_no NUMBER(3) PRIMARY KEY,

ord\_date DATE NOT NULL);

B. CREATE TABLE ord\_details

(ord\_no NUMBER(2) UNIQUE, NOT NULL,

item\_no NUMBER(3),

ord\_date DATE DEFAULT SYSDATE NOT NULL);

C. CREATE TABLE ord\_details

(ord\_no NUMBER(2) ,

item\_no NUMBER(3),

ord\_date DATE DEFAULT NOT NULL,

CONSTRAINT ord\_uq UNIQUE (ord\_no),

CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));

D. CREATE TABLE ord\_details

(ord\_no NUMBER(2),

item\_no NUMBER(3),

ord\_date DATE DEFAULT SYSDATE NOT NULL,

CONSTRAINT ord\_pk PRIMARY KEY (ord\_no, item\_no));

Answer: D

答案解析

A，一个表只有一个主键。A错。

sys@TEST0910> CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2) PRIMARY KEY,

  3  item\_no NUMBER(3) PRIMARY KEY,

  4  ord\_date DATE NOT NULL);

item\_no NUMBER(3) PRIMARY KEY,

                  \*

ERROR at line 3:

ORA-02260: table can have only one primary key

B答案：UNIQUE, NOT NULL之间不要逗号，列级约束不要逗号，B错

sys@TEST0910> CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2) UNIQUE, NOT NULL,

  3  item\_no NUMBER(3),

  4  ord\_date DATE DEFAULT SYSDATE NOT NULL);

(ord\_no NUMBER(2) UNIQUE**,**NOT NULL,

                          \*

ERROR at line 2:

ORA-00904: : invalid identifier

逗号去掉可以创建，成功

sys@TEST0910> CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2) UNIQUE NOT NULL,

  3  item\_no NUMBER(3),

  4  ord\_date DATE DEFAULT SYSDATE NOT NULL);

Table created.

C答案，两处错误，一个DEFAULT 后面没有指定默认值，一个是ord\_no列的UNIQUE 和PRIMARY KEY

sys@TEST0910> CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2) ,

  3  item\_no NUMBER(3),

  4  ord\_date DATE DEFAULT NOT NULL,

  5  CONSTRAINT ord\_uq UNIQUE (ord\_no),

  6  CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));

ord\_date DATE DEFAULT NOT NULL,

                      \*

ERROR at line 4:

ORA-00936: missing expression

sys@TEST0910> CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2) ,

  3  tem\_no NUMBER(3),

  4  ord\_date DATE DEFAULT sysdate NOT NULL,

  5  CONSTRAINT ord\_uq UNIQUE (ord\_no),

  6   CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));

CONSTRAINT ord\_uq UNIQUE (ord\_no),

                  \*

ERROR at line 5:

ORA-02261: such unique or primary key already exists in the table

sys@TEST0910> CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2) ,

  3  tem\_no NUMBER(3),

  4   ord\_date DATE DEFAULT sysdate NOT NULL,

  5  CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));

Table created.

D答案，正确

sys@TEST0910>  CREATE TABLE ord\_details

  2  (ord\_no NUMBER(2),

  3  item\_no NUMBER(3),

  4  ord\_date DATE DEFAULT SYSDATE NOT NULL,

  5  CONSTRAINT ord\_pk PRIMARY KEY (ord\_no, item\_no));

Table created.

33. You want to create an ORD\_DETAIL table to store details for an order placed having the following

business requirement:

1) The order ID will be unique and cannot have null values.（唯一+非空约束=主键约束）

2) The order date cannot have null values and the default should be the current date. （order date非空default sysdate）

3) The order amount should not be less than 50.（不少于50，即大于等于50）

4) The order status will have values either shipped or not shipped.（order status IN ('Shipped', 'Not Shipped'） ）

5) The order payment mode should be cheque, credit card, or cash on delivery (COD).（ order payment  mode ('Cheque', 'Credit Card', 'Cash On Delivery')）

Which is the valid DDL statement for creating the ORD\_DETAIL table?

A. CREATE TABLE ord\_details

(ord\_id NUMBER(2) CONSTRAINT ord\_id\_nn NOT NULL,   **此处没有唯一约束**

ord\_date DATE DEFAULT SYSDATE NOT NULL,

ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min

CHECK (ord\_amount > 50),    **此处应该大于等于50**

ord\_status VARCHAR2(15) CONSTRAINT ord\_status\_chk

CHECK (ord\_status IN ('Shipped', 'Not Shipped')),

ord\_pay\_mode VARCHAR2(15) CONSTRAINT ord\_pay\_chk

CHECK (ord\_pay\_mode IN ('Cheque', 'Credit Card',

'Cash On Delivery')));

B. CREATE TABLE ord\_details

(ord\_id NUMBER(2) CONSTRAINT ord\_id\_uk UNIQUE NOT NULL,

ord\_date DATE DEFAULT SYSDATE NOT NULL,

ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min

CHECK (ord\_amount > 50),     **此处应该大于等于50**

ord\_status VARCHAR2(15) CONSTRAINT ord\_status\_chk

CHECK (ord\_status IN ('Shipped', 'Not Shipped')),

ord\_pay\_mode VARCHAR2(15) CONSTRAINT ord\_pay\_chk

CHECK (ord\_pay\_mode IN ('Cheque', 'Credit Card',

'Cash On Delivery')));

C. CREATE TABLE ord\_details

(ord\_id NUMBER(2) CONSTRAINT ord\_id\_pk PRIMARY KEY,

ord\_date DATE DEFAULT SYSDATE NOT NULL,

ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min

CHECK (ord\_amount >= 50),

ord\_status VARCHAR2(15) CONSTRAINT ord\_status\_chk

CHECK (ord\_status IN ('Shipped', 'Not Shipped')),

ord\_pay\_mode VARCHAR2(15) CONSTRAINT ord\_pay\_chk

CHECK (ord\_pay\_mode IN ('Cheque', 'Credit Card',

'Cash On Delivery')));

D. CREATE TABLE ord\_details

(ord\_id NUMBER(2),   **此处没有唯一非空约束**

ord\_date DATE NOT NULL DEFAULT SYSDATE,    **此处报语法错误**

ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min

CHECK (ord\_amount >= 50),

ord\_status VARCHAR2(15) CONSTRAINT ord\_status\_chk

CHECK (ord\_status IN ('Shipped', 'Not Shipped')),

ord\_pay\_mode VARCHAR2(15) CONSTRAINT ord\_pay\_chk

CHECK (ord\_pay\_mode IN ('Cheque', 'Credit Card',

'Cash On Delivery')));

Answer: C

D答案应该先default sysdate，后not null。

sys@TEST0910> create table t7(ord\_date date not null default sysdate);

create table t7(ord\_date date not null default sysdate)

                                       \*

ERROR at line 1:

ORA-00907: missing right parenthesis

sys@TEST0910> create table t7(ord\_date date default sysdate not null);

Table created.

34. You created an ORDERS table with the following description:

name                Null            Type

ORD\_ID              NOT NULL        NUMBER(2)

CUST\_ID             NOT NULL        NUMBER(3)

ORD\_DATE            NOT NULL        DATE

ORD\_AMOUNT          NOT NULL        NUMBER (10,2)

You inserted some rows in the table. After some time, you want to alter the table by creating the

PRIMARY KEY constraint on the ORD\_ID column. Which statement is true in this scenario?

A. You cannot have two constraints on one column.

B. You cannot add a  primary key constraint if data exists in the column.

C. The primary key constraint can be created only at the time of table creation .

D. You can add the  primary key constraint even if data exists,  provided that   there are  no duplicate

values.

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

A. 一个列上不能有两个约束,错误，例如非空+唯一可以作用在一个列上。

B. 如果数据已经存在不能添加主键约束，错误，有数据存在，也可以插入主键约束，前提是数据没有重复

C. 主键约束只有在表创建时才能被创建，错误。有了数据之后一样可以增加，不一定只有在创建的时候加

D. 即使数据存在，如果没有重复值，你能添加主键约束，正确。

实验验证，有数据是否可以加主键约束。

sh@TESTDB> create table orders

  2  (ord\_id number(2) not null,

  3  cust\_id number(3) not null,

  4  ord\_date date not null,

  5  ord\_amount number(10,2) not null)

  6  ;

Table created.

h@TESTDB> select \* from orders;

    ORD\_ID    CUST\_ID ORD\_DATE  ORD\_AMOUNT

---------- ---------- --------- ----------

         1         11      03-SEP-13        111

         1         10      03-SEP-13        111

         2         22       03-SEP-13        222

         3         33       03-SEP-13        333

sh@TESTDB> alter table orders add constraint ord\_pk primary key(ord\_id);

alter table orders add constraint ord\_pk primary key(ord\_id)

                                  \*

ERROR at line 1:

ORA-02437: cannot validate (SH.ORD\_PK) - primary key violated

去掉ord\_id重复值

sh@TESTDB> select \* from orders;

    ORD\_ID    CUST\_ID ORD\_DATE  ORD\_AMOUNT

---------- ---------- --------- ----------

         1         11 03-SEP-13        111

         2         22 03-SEP-13        222

         3         33 03-SEP-13        333

sh@TESTDB> alter table orders add constraint ord\_pk primary key(ord\_id);

Table altered.

35. Which two statements are true regarding constraints? (Choose two.)

A. A table can have only one primary key and one foreign key.

B. A table can have only one primary key but multiple foreign keys.

C. Only the primary key can be defined at the column and table levels.

D. The foreign key and parent table primary key must have the same name.

E. Both primary key and foreign key constraints can be defined at both column and table levels.

Answer: BE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

A. 一个表只能有一个主键和一个外键,不正确因为一个表只能有一个主键，但可以有多个外键

B. 一个表只能有一个主键，但可以有多个外键,正确。

C. 只有主键能被定义成列级或表级，不正确因为除了NOT NULL只有被定义成列级，其它的约束都可以被定义成列级或表级

D. 外键和父表主键必须名字相同，不正确因为名字可以不同，但是数据类型需一致

E. 主键和外键都可以被定义成列级或表级

. 36. Examine the following SQL commands:

SQL>CREATE TABLE products (

prod\_id NUMBER(3) CONSTRAINT p\_ck CHECK (prod\_id > 0),

prod\_name CHAR(30),

prod\_qty NUMBER(6),

CONSTRAINT p\_name NOT NULL,

CONSTRAINT prod\_pk PRIMARY KEY (prod\_id));

SQL>CREATE TABLE warehouse (

warehouse\_id NUMBER(4),

roomno NUMBER(10) CONSTRAINT r\_id CHECK(roomno BETWEEN 101 AND 200),

location VARCHAR2(25),

prod\_id NUMBER(3),

CONSTRAINT wr\_pr\_pk PRIMARY KEY (warehouse\_id,prod\_id),

CONSTRAINT prod\_fk FOREIGN KEY (prod\_id) REFERENCES products(prod\_id));

Which statement is true regarding the execution of the above SQL commands?

A. Both commands execute successfully.

B. The first CREATE TABLE command generates an error because the NULL constraint is not valid.

C. The second CREATE TABLE command generates an error because the CHECK constraint is not valid.

D. The first CREATE TABLE command generates an error because CHECK and PRIMARY KEY

constraints cannot be used for the same column.

E. The first CREATE TABLE command generates an error because the column PROD\_ID cannot be used

in the PRIMARY KEY and FOREIGN KEY constraints.

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12905109>

第一个命令报错，因为CONSTRAINT p\_name NOT NULL不能定义为表级，第一个命令里的CHECK和主键约束能用在相同的列上。

sh@TESTDB> create table pro

  2  (prod\_id number,

  3  prod\_name char(30),

  4  constraint p\_name not null);

constraint p\_name not null)

                  \*

ERROR at line 4:

ORA-00904: : invalid identifier

A错误，因为第一个报错。

B正确，NOT NULL不能定义为表级，只能定义为列级

C错误，between可以用在check约束里

D错误，check和主键约束可以用在同一列

E错误，可以把主键作为其他表的外键

37. You issued the following command to drop the PRODUCTS table:

SQL> DROP TABLE products;

What is the implication of this command? (Choose all that apply.)

A. All data along with the table structure is  deleted.

B. The  pending transaction in the session   is  committed.

C. All indexes on the table will remain but they are   invalidated.

D. All views and synonyms will remain but they are   invalidated.

E. All data in the table are  deleted but the   table structure will  remain.

Answer: ABD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12796201>

A正确，表结构以及表中所有数据都删除。

B正确，提交所有待定的事务处理。

C错误，删除表后，在数据库中会失去表的所有数据以及与表关联的所有索引。

D正确，保留视图和同义词，但不再有效。

E错误，表结构以及表中所有数据都删除。

38. Which two statements are true regarding views? (Choose two.)

A. A simple view in which column aliases have been used cannot be updated.

B. Rows cannot be deleted through a view if the view definition contains the DISTINCT keyword.

C. Rows added through a view are deleted from the table automatically when the view is dropped.

D. The OR REPLACE option is used to change the definition of an existing view without dropping and re-creating it.

E. The WITH CHECK OPTION constraint can be used in a view definition to restrict the columns displayed through the view.

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12790467>

A错误，简单视图可以执行DML操作。

B正确，包含DISTINCT关键字不能被修改。

C错误，删除视图不会影响视图的基表。

D正确，使用OR REPLACE选项，可以创建一个视图，甚至可以创建一个与已存在的视图同名的视图，以便替换旧版本的视图。这意味着可以更改视图，而不必经过删除、重新创建对象和重新授予对象权限的过程。

E错误，WITH CHECK OPTION子句指定通过视图执行的INSERT和UPDATE不能创建该视图无法选择的行。因此，能够对要插入或更新的数据强制执行完整性约束条件和数据验证检查。而不是限制显示的值。

39. Evaluate the following command:

CREATE TABLE employees

(employee\_id      NUMBER(2) PRIMARY KEY,

last\_name        VARCHAR2(25) NOT NULL,

department\_id    NUMBER(2)NOT NULL,

job\_id           VARCHAR2(8),

salary        NUMBER(10,2));

You issue the following command to create a view that displays the IDs and last names of the sales staff

in the organization:

CREATE OR REPLACE VIEW sales\_staff\_vu AS

SELECT employee\_id, last\_name,job\_id

FROM employees

WHERE job\_id LIKE 'SA\_%'

WITH CHECK OPTION

Which two statements are true regarding the above view? (Choose two.)

A. It allows you to insert rows into the  EMPLOYEES table .

B. It allows you to delete details of the existing sales staff from the EMPLOYEES table.

C. It allows you to update job IDs of the existing sales staff to any other job ID in the EMPLOYEES table.

D. It allows you to insert IDs, last names, and job IDs of the sales staff from the view if it is used in multitable INSERT statements.

Answer: BD

答案解析：

如果硬要选两个，我觉得应该是BC

参考：<http://blog.csdn.net/rlhua/article/details/12790467>

A错误，因为employees表的department\_id有非空约束，如果通过视图插入，则department\_id列则为空，会报错。

B正确，可以通过视图来删除employees任意行。

C错误，因为有WITH CHECK OPTION，是根据job\_id LIKE 'SA\_%' 来创建的视图，所以不能更改为不是SA\_开头的，但是可以更新为SA\_开头的。

D错误，也是因为department\_id有非空约束，如果通过视图插入，则department\_id列则为空，会报错。

但是可以通过multitable INSERT在视图上插入，所有官方文档有错误？

参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_9014.htm#SQLRF55094>

官方文档说明，multitable INSERT 不能用在视图上，故D是错误的。

**Restrictions on Multitable Inserts**Multitable inserts are subject to the following restrictions:

* You can perform multitable inserts only on tables, not on views or materialized views.
* You cannot perform a multitable insert into a remote table.
* You cannot specify a TABLE collection expression when performing a multitable insert.
* In a multitable insert, all of the *insert\_into\_clause*s cannot combine to specify more than 999 target columns.
* Multitable inserts are not parallelized if any target table is index organized or if any target table has a bitmap index defined on it.
* Plan stability is not supported for multitable insert statements.
* You cannot specify a sequence in any part of a multitable insert statement. A multitable insert is considered a single SQL statement. Therefore, the first reference to NEXTVAL generates the next number, and all subsequent references in the statement return the same number.

实验验证：

hr@TEST1107> create table emp

  2  (employee\_id number primary key,

  3  last\_name varchar2(25) not null,

  4  department\_id number(2) not null,

  5  job\_id varchar2(8),

  6  salary number(10,2)

  7  );

Table created.

hr@TEST1107> create or replace view sales\_staff\_vu as

  2  select employee\_id,last\_name,job\_id

  3  from emp

  4  where job\_id like 'SA\_%'

  5  with check option

  6  /

View created.

hr@TEST1107> insert into emp select EMPLOYEE\_ID,LAST\_NAME,DEPARTMENT\_ID,JOB\_ID,SALARY from employees

  2  where job\_id like 'SA\_%' and rownum<6

  3  /

5 rows created.

hr@TEST1107> select \* from emp;

EMPLOYEE\_ID LAST\_NAME                 DEPARTMENT\_ID JOB\_ID       SALARY

----------- ------------------------- ------------- -------- ----------

        145 Russell                              80 SA\_MAN        14000

        146 Partners                             80 SA\_MAN        13500

        147 Errazuriz                            80 SA\_MAN        12000

        148 Cambrault                            80 SA\_MAN        11000

        149 Zlotkey                              80 SA\_MAN        10500

hr@TEST1107> select \* from sales\_staff\_vu;

EMPLOYEE\_ID LAST\_NAME                 JOB\_ID

----------- ------------------------- --------

        145 Russell                   SA\_MAN

        146 Partners                  SA\_MAN

        147 Errazuriz                 SA\_MAN

        148 Cambrault                 SA\_MAN

        149 Zlotkey                   SA\_MAN

 A答案：A答案错误，DEPARTMENT\_ID为非空，不能插入null值。

hr@TEST1107> insert into sales\_staff\_vu values (501,'lihua','SA\_MAN');

insert into sales\_staff\_vu values (501,'lihua','SA\_MAN')

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("HR"."EMP"."DEPARTMENT\_ID")

B答案：正确，可以删除

hr@TEST1107> delete from sales\_staff\_vu where EMPLOYEE\_ID=145;

1 row deleted.

hr@TEST1107> select \* from sales\_staff\_vu;

EMPLOYEE\_ID LAST\_NAME                 JOB\_ID

----------- ------------------------- --------

        146 Partners                  SA\_MAN

        147 Errazuriz                 SA\_MAN

        148 Cambrault                 SA\_MAN

        149 Zlotkey                   SA\_MAN

hr@TEST1107> select \* from emp;

EMPLOYEE\_ID LAST\_NAME                 DEPARTMENT\_ID JOB\_ID       SALARY

----------- ------------------------- ------------- -------- ----------

        146 Partners                             80 SA\_MAN        13500

        147 Errazuriz                            80 SA\_MAN        12000

        148 Cambrault                            80 SA\_MAN        11000

        149 Zlotkey                              80 SA\_MAN        10500

C答案：可以将job\_id更新为SA\_开头的，不能更新为其他不是SA\_开头的。

hr@TEST1107> update sales\_staff\_vu set JOB\_ID='SA\_REP' where EMPLOYEE\_ID=146;

1 row updated.

hr@TEST1107> update sales\_staff\_vu set JOB\_ID='MK\_MAN' where EMPLOYEE\_ID=147;

update sales\_staff\_vu set JOB\_ID='MK\_MAN' where EMPLOYEE\_ID=147

       \*

ERROR at line 1:

ORA-01402: view WITH CHECK OPTION where-clause violation

D答案：multitable INSERT不能插入非空

hr@TEST1107> insert into sales\_staff\_vu

  2  select EMPLOYEE\_ID,LAST\_NAME,JOB\_ID from employees

  3  where job\_id like 'SA\_%'

  4  /

insert into sales\_staff\_vu

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("HR"."EMP"."DEPARTMENT\_ID")

实验证明，multitable INSERT可以插入视图？

hr@TEST1107> create or replace view sales\_staff\_vu\_new as

  2  select employee\_id,last\_name,job\_id,department\_id from emp

  3  where job\_id like 'SA\_%'

  4  with check option

  5  /

View created.

hr@TEST1107> insert into sales\_staff\_vu\_new

  2  select EMPLOYEE\_ID,LAST\_NAME,JOB\_ID,DEPARTMENT\_ID from employees where job\_id like 'SA\_%'

  3  and EMPLOYEE\_ID in (150,151);

2 rows created.

hr@TEST1107> select \* from sales\_staff\_vu\_new;

EMPLOYEE\_ID LAST\_NAME                 JOB\_ID   DEPARTMENT\_ID

----------- ------------------------- -------- -------------

        146 Partners                  SA\_REP              80

        147 Errazuriz                 SA\_MAN              80

        148 Cambrault                 SA\_MAN              80

        149 Zlotkey                   SA\_MAN              80

        150 Tucker                    SA\_REP              80

        151 Bernstein                 SA\_REP              80

6 rows selected.

40. View the Exhibit to examine the description for the SALES and PRODUCTS tables.

You want to create a SALE\_PROD view  by executing the following SQL statement:

CREATE VIEW sale\_prod

AS SELECT p.prod\_id, cust\_id, SUM(quantity\_sold) "Quantity" , SUM(prod\_list\_price) "Price"

FROM products p, sales s

WHERE p.prod\_id=s.prod\_id

GROUP BY p.prod\_id, cust\_id;

Which statement is true regarding the execution of the  above statement?

A. The view will be created and you can perform DML operations on the view.

B. The view will be created but no DML operations will be allowed on the view.

C. The view will not be created because the join statements are not allowed for creating a  view.

D. The view will not be created because the  GROUP BY clause is not allowed for creating a  view.

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12790467>

A错误，此视图为复杂视图，带有group by子句，不能对视图进行DML操作，

B正确，可以创建成功，但不能进行DML操作。

*sh@TESTDB> CREATE VIEW sale\_prod*

*2  AS SELECT p.prod\_id, cust\_id, SUM(quantity\_sold) "Quantity" , SUM(prod\_list\_price) "Price"*

*3  FROM products p, sales s*

*4  WHERE p.prod\_id=s.prod\_id*

*5  GROUP BY p.prod\_id, cust\_id;*

*View created.*

C错误，可以通过join来连接两张表创建视图。

D错误，可以通过group by 子句来创建视图。

41. Which two statements are true regarding views? (Choose two.)

A. A  subquery that defines a view cannot include the  GROUP BY clause.

B. A  view that is  created with the subquery having the DISTINCT keyword can be updated.

C. A  view that is  created with the subquery having   the pseudo  column ROWNUM keyword cannot

be updated.

D. A  data manipulation language ( DML)  operation can be performed on a  view that is  created with

the subquery having all the NOT NULL columns of a table.

Answer: CD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12790467>

A错误，创建视图的子查询可以使用group by子句。

B错误，子查询中带有DISTINCT 不能对视图进行DML操作。

C正确，子查询中带有ROWNUM 关键字不能对视图进行DML操作。

D正确，所有非空列都可以进行DML操作。

42. Which three statements are true regarding views? (Choose three.)

A. Views can be created only from tables.

B. Views can be created from tables or other views.

C. Only simple views can use indexes existing on the underlying tables.

D. Both simple and complex views can use indexes existing on the underlying tables.

E. Complex views can be created only on multiple tables that exist in the same schema.

F. Complex views can be created on multiple tables that exist in the same or different schemas.

Answer: BDF

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12790467>

A错误，视图是一种基于表或其它视图的逻辑表。

B正确，视图是一种基于表或其它视图的逻辑表。

C错误，简单视图和复杂视图都能使用相关表的索引。

D正确，同C.

E错误，可以是不同schema间。

F正确，复杂视图的子查询可以是相同或者不同的schema间的多张表。

43. Evaluate the following CREATE SEQUENCE statement:

CREATE SEQUENCE seq1

START WITH 100

INCREMENT BY 10

MAXVALUE 200

CYCLE

NOCACHE;

The SEQ1 sequence has generated numbers up to the maximum limit of 200. You issue the following

SQL statement:

SELECT seq1.nextval FROM dual;

What is displayed by the SELECT statement?

A. 1

B. 10

C. 100

D. an error

Answer: A

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12776779>

实验验证：

1、首先创建这个序列

sh@TESTDB> CREATE SEQUENCE seq1

  2  START WITH 100

  3  INCREMENT BY 10

  4  MAXVALUE 200

  5  CYCLE

  6  NOCACHE;

Sequence created.

 2、查看值。

sh@TESTDB> select seq1.nextval from dual;

   NEXTVAL

----------

       100

sh@TESTDB> select seq1.currval from dual;

   CURRVAL

----------

       100

 3、因为该序列是循环的，到达最大值后从最小值MINVALUE开始循环，该序列因为省略MINVALUE省略，所以默认为NOMINVALUE最小值为 1，所有又从1开始。

sh@TESTDB> select seq1.nextval from dual;

   NEXTVAL

----------

       200

sh@TESTDB>

sh@TESTDB> select seq1.nextval from dual;

   NEXTVAL

----------

         1

 故答案选A

44. View the Exhibit and examine the structure of the ORD table.

Evaluate the following SQL statements that are executed in a user session in the specified order:

CREATE SEQUENCE ord\_seq;

SELECT ord\_seq.nextval

FROM dual;

INSERT INTO ord

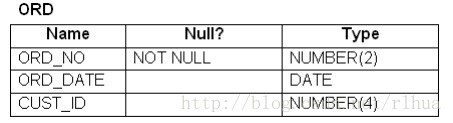
VALUES (ord\_seq.CURRVAL, '25-jan-2007',101);

UPDATE ord

SET  ord\_no= ord\_seq.NEXTVAL

WHERE cust\_id =101;

What would be the outcome of the above statements?



A. All the statements would execute successfully and the ORD\_NO column would contain the value 2 for

the CUST\_ID 101.

B. The CREATE SEQUENCE command would not execute because the minimum value and maximum

value for the sequence have not been specified.

C. The CREATE SEQUENCE command would not execute because the starting value of the sequence

and the increment value have not been specified.

D. All the statements would execute successfully and the ORD\_NO column would have the value 20 for

the CUST\_ID 101 because the default CACHE value is 20.

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12776779>

使用NEXTVAL和CURRVAL的规则

可以在下列上下文中使用NEXTVAL和CURRVAL：

• 不是子查询一部分的SELECT 语句的SELECT 列表

• INSERT 语句中子查询的SELECT 列表

• INSERT 语句的VALUES 子句

• UPDATE 语句的SET 子句

CREATE SEQUENCE时可以省略所有参数，默认起始值为1，步长为1，无上限

此处提一下CACHE这个关键字：如果建立序列时不指定CACHE和NOCACHE，默认值为20，

即一次性从序列里取20个数放入内存，如果内存崩溃，则这20个数就会丢失，再取值时从第21个数开始取值，CACHE设置的最小值为2。

实验验证：

1、创建一个序列，什么选项都不用跟。

*sh@TESTDB> CREATE SEQUENCE ord\_seq;*

*Sequence created.*

 2、使用序列。应用select列表， INSERT 语句的VALUES 子句，UPDATE 语句的SET 子句都可以成功执行。排除BC，而最后经过UPDATE后ORD\_NO列值为2，而不是20.

*sh@TESTDB> SELECT ord\_seq.nextval  FROM dual;*

*NEXTVAL*

*----------*

*1*

*sh@TESTDB> INSERT INTO ord  VALUES (ord\_seq.CURRVAL, '25-jan-2007',101);*

*1 row created.*

*sh@TESTDB> UPDATE ord SET  ord\_no= ord\_seq.NEXTVAL WHERE cust\_id =101;*

*1 row updated.*

45. Which two statements are true about sequences created in a single instance database? (Choose

two.)

A. The numbers generated by a sequence can be used only for one table.

B. DELETE <sequencename> would remove a sequence from the database.

C. CURRVAL is used to refer to the last sequence number that has been generated.

D. When the MAXVALUE limit for a sequence is reached, you can increase the MAXVALUE limit by using

the ALTER SEQUENCE statement.

E. When a database instance shuts down abnormally, the sequence numbers that have been cached but

not used would be available once again when the database instance is restarted.

Answer: CD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12776779>

A不正确，因为一个序列生成的值可以用于多个表

B不正确，因为删除序列的语句为DROP SEQUENCE sequencename

C正确，对，CURRVAL是最后的那个值

D正确，对当最大值达到极限时，可以通过ALTER SEQUENCE修改序列来增加最大值

E不正确，因为如果内存非正常关闭，缓存的数会丢失。

如果建立序列时不指定CACHE和NOCACHE，默认值为20，

即一次性从序列里取20个数放入内存，如果内存崩溃，则这20个数就会丢失，再取值时从第21个数开始取值，CACHE设置的最小值为2。

46. Which statements are correct regarding indexes? (Choose all that apply.)

A. When a table is dropped, the corresponding indexes are automatically dropped.

B. A FOREIGN KEY constraint on a column in a table automatically creates a nonunique index.

C. A nondeferrable PRIMARY KEY or UNIQUE KEY constraint in a table automatically creates a unique

index.

D. For each data manipulation language (DML) operation performed, the corresponding indexes are

automatically updated.

Answer: ACD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12780775>

A正确，在删除表时，会删除相应的索引。

B错误，外键不能够自动创建索引，只有主键约束和唯一性约束才能自动创建唯一的索引。

C正确，在表定义中定义了PRIMARY KEY 或UNIQUE 约束条件，则会自动创建一个唯一的索引。

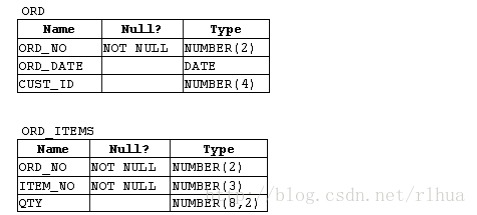
D正确，在具有索引的表上提交每项DML 操作后，都意味着必须更新相应的索引。

47. View the Exhibit and examine the structure of ORD and ORD\_ITEMS tables.

The ORD\_NO column is PRIMARY KEY in the ORD table and the ORD\_NO and ITEM\_NO columns are

composite PRIMARY KEY in the ORD\_ITEMS table.

Which two CREATE INDEX statements are valid? (Choose two.)



A. CREATE INDEX ord\_idx1

ON ord(ord\_no);

B. CREATE INDEX ord\_idx2

ON ord\_items(ord\_no);

C. CREATE INDEX ord\_idx3

ON ord\_items(item\_no);

D. CREATE INDEX ord\_idx4

ON ord,ord\_items(ord\_no, ord\_date,qty);

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12780775>

实验验证：

1、创建两张表

创建ord表

*sh@TESTDB> create table ord*

*2  (ord\_no number(2) not null,*

*3  ord\_date date,*

*4  cust\_id number(4));*

*Table created.*

创建ord\_items表

*sh@TESTDB> create table ord\_items*

*2  (ord\_no number(2) not null,*

*3  item\_no number(3) not null,*

*4  qty number(8,2));*

*Table created.*

 2、增加约束，使其满足题意

*sh@TESTDB> alter table ord add constraint pk\_ord primary key(ord\_no);*

*Table altered.*

*sh@TESTDB> alter table ord\_items add constraint pk\_ord\_item primary key(ord\_no,item\_no);*

*Table altered.*

A错误，因为ord\_no是主键，已经自动创建了一个唯一的索引，故不能再创建索引。

*sh@TESTDB> create index ord\_idx1 on ord(ord\_no);*

*create index ord\_idx1 on ord(ord\_no)*

*\**

*ERROR at line 1:*

*ORA-01408: such column list already indexed*

B,C正确，虽然在两列创建了主键约束，即自动创建的唯一索引是按照两列的组合来创建的，所以单列是可以创建索引的，

*sh@TESTDB> CREATE INDEX ord\_idx2*

*2  ON ord\_items(ord\_no);*

*Index created.*

*sh@TESTDB> CREATE INDEX ord\_idx3*

*2  ON ord\_items(item\_no);*

*Index created.*

D错误，属于语法错误，不能两张表同时建立索引

*sh@TESTDB> CREATE INDEX ord\_idx4*

*2  ON ord,ord\_items(ord\_no, ord\_date,qty);*

*ON ord,ord\_items(ord\_no, ord\_date,qty)*

*\**

*ERROR at line 2:*

*ORA-00906: missing left parenthesis*

48. Which two statements are true regarding indexes? (Choose two.)

A. They can be created on tables and clusters.

B. They can be created on tables and simple views.

C. You can create only one index by using the same columns.

D. You can create more than one index by using the same columns if you specify distinctly different

combinations of the columns.

Answer: AD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12780775>

A正确，可以在表和簇上面创建索引。

B错误，不能再简单视图上创建索引

C错误，可以和别的列组合来创建别的索引。

D正确，通过指定不同列的组合来在同一列创建一个或多个索引，这是是正确的。

49. The ORDERS table belongs to the user OE. OE has granted the SELECT privilege on the ORDERS

table to the user HR.

Which statement would create a synonym ORD so that HR can execute the following query successfully?

SELECT \* FROM ord;

A. CREATE SYNONYM ord FOR orders; This command is issued by OE.

B. CREATE PUBLIC SYNONYM  ord FOR orders; This command is issued by OE.

C. CREATE SYNONYM  ord FOR oe.orders; This command is issued by the database administrator.

D. CREATE PUBLIC SYNONYM ord FOR oe.orders; This command is issued by the database administrator.

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12781353>

题意的问题，要创建一个同义词，使HR用户可以通过SELECT \* FROM ord;来查询OE的orders表。

只能由数据库管理员创建公用的同义词，使HR用户来访问。

C创建的私有同义词，因为由DBA用户创建的，所以只能由DBA用户访问。

50. SLS is a private synonym for the SH.SALES table.

The user SH issues the following command:

DROP SYNONYM sls;

Which statement is true regarding the above SQL statement?

A. Only the synonym would be dropped.

B. The synonym   would be dropped and the corresponding table would become invalid.

C. The synonym would be dropped and the packages referring to the synonym would be dropped.

D. The synonym would be dropped and any PUBLIC synonym with the same name becomes invalid.

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12781353>

删除同义词，仅仅只是删除自己的同义词。

51. Which statement is true regarding synonyms?

A. Synonyms can be created only   for a table.  不只是表，schema object

B. Synonyms are used to reference only those tables that are owned by another user.

C. A public synonym and a private synonym can exist with the same name for the same table. 共有和私有可以共同存在

D. The DROP SYNONYM statement removes the synonym, and the table on which the synonym has

been created becomes invalid.

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12781353>

A错误，不只是表，是数据库所有的对象，如表、视图、序列、过程等。

B错误，由数据库管理员创建的公共的同义词，可以所有的用户都可以访问，而由用户自己创建的同义词，只能自己访问。

C正确，共有同义词和私有同义词的名字在一张表中可以相同。

实验验证：

*sys@TEST0924> create public synonym semp for scott.emp;*

*Synonym created.*

*sys@TEST0924> conn scott/tiger*

*Connected.*

*scott@TEST0924> create synonym semp for emp;-------------必须有创建同义词权限，否则报错*

*create synonym semp for emp*

*\**

*ERROR at line 1:*

*ORA-01031: insufficient privileges*

*scott@TEST0924> conn /as sysdba*

*Connected.*

*sys@TEST0924> grant CREATE synonym to scott;-------------赋予创建同义词权限*

*Grant succeeded.*

*sys@TEST0924> conn scott/tiger*

*Connected.*

*scott@TEST0924> create synonym semp for emp;------------创建一个和共有同义词的一样名称的同义词，执行成功。*

*Synonym created.*

D错误，删除了同义词，但是表不会被变为无效的。

52. View the Exhibit and examine the structure of the PRODUCTS table.

Using the PRODUCTS table,  you issue the following query to generate the names, current list price,

and discounted list price for all those products whose list price falls below $10 after a discount of 25% is

applied on it.

SQL>SELECT prod\_name, prod\_list\_price,

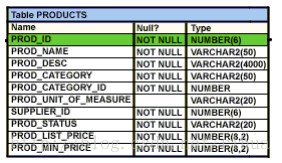
prod\_list\_price - (prod\_list\_price \* .25) "DISCOUNTED\_PRICE"

FROM products

WHERE discounted\_price < 10;

The query generates an error.

What is the reason for the error?



A. The parenthesis should be added to enclose the entire expression.

B. The double quotation marks  should be removed from the column alias.

C. The column alias should be replaced with the expression in the WHERE clause.

D. The column alias should be put in uppercase and enclosed with in double quotation marks   in the

WHERE clause.

Answer: C

 答案解析：

where子句不能跟别名,可以跟表达式

 实验验证：

*scott@TESTDB> select empno,ename,sal+10 "sal10" from emp where sal10<1500;*

*select empno,ename,sal+10 "sal10" from emp where sal10<1500*

*\**

*ERROR at line 1:*

*ORA-00904: "SAL10": invalid identifier*

*scott@TESTDB> select empno,ename,sal+10 "sal10" from emp where sal+10<1500;*

*EMPNO ENAME           sal10*

*---------- ---------- ----------*

*7369 SMITH             810*

*7521 WARD             1260*

*7654 MARTIN           1260*

*7876 ADAMS            1110*

*7900 JAMES             960*

*7934 MILLER           1310*

*6 rows selected.*

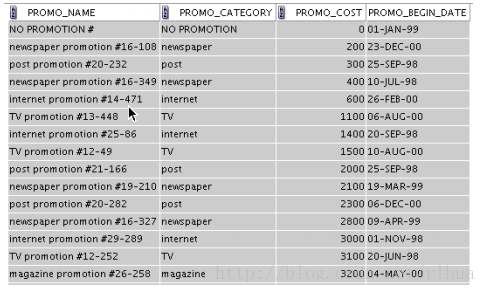
53. View the Exhibit and examine the data in the PROMOTIONS table.

PROMO\_BEGIN\_DATE is stored in the default date format, dd-mon-rr.

You need to produce a report that provides   the name, cost,  and start date of all promos in the POST

category that were   launched before January 1,  2000.

Which SQL statement would you use?



A. SELECT promo\_name, promo\_cost, promo\_begin\_date

FROM promotions

WHERE promo\_category = 'post' AND promo\_begin\_date < '01-01-00';

B. SELECT promo\_name, promo\_cost, promo\_begin\_date

FROM promotions

WHERE promo\_cost LIKE 'post%' AND promo\_begin\_date < '01-01-2000';

C. SELECT promo\_name, promo\_cost, promo\_begin\_date

FROM promotions

WHERE promo\_category LIKE 'P%' AND promo\_begin\_date < '1-JANUARY-00'; 可以，但是like条件不满足

D. SELECT promo\_name, promo\_cost, promo\_begin\_date

FROM promotions

WHERE promo\_category LIKE '%post%' AND promo\_begin\_date < '1-JAN-00';

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832019>

<http://blog.csdn.net/rlhua/article/details/12832885>

在表达式中，Oracle Server 可以自动执行以下转换：

从 VARCHAR2 或CHAR隐式转换为NUMBER

从VARCHAR2 或CHAR隐式转换为DATE，但必须符合日期的规定的格式。

Oracle Server  可以在表达式中自动执行数据类型转换。例如，表达式hire\_date > '01-JAN-90' 将导致字符串'01-JAN-90' 隐式转换为一个日期。因此，表达式中的VARCHAR2 或CHAR 值可以隐式转换为数字或日期数据类型。

A，错误，需要视同to\_date显示转换

实验验证：

首先确认默认的时间格式：

*sys@TEST0924> select sysdate from dual;*

*SYSDATE*

*------------------*

*17-OCT-13*

A，

*sh@TEST0924> SELECT promo\_name, promo\_cost, promo\_begin\_date FROM promotions WHERE promo\_category = 'post' AND promo\_begin\_date < '01-01-00';*

*SELECT promo\_name, promo\_cost, promo\_begin\_date FROM promotions WHERE promo\_category = 'post' AND promo\_begin\_date < '01-01-00'*

*\**

*ERROR at line 1:*

*ORA-01843: not a valid month*

显示转换后：

*sh@TEST0924> SELECT promo\_name, promo\_cost, promo\_begin\_date FROM promotions WHERE promo\_category = 'post' AND promo\_begin\_date <to\_date( '01-01-00','dd-mm-yy');*

*PROMO\_NAME                     PROMO\_COST PROMO\_BEGIN\_DATE*

*------------------------------ ---------- ------------------*

*post promotion #20-232                300 25-SEP-98*

*post promotion #21-166               2000 25-SEP-98*

*post promotion #20-449               4200 10-NOV-98*

B,

*sh@TEST0924> SELECT promo\_name, promo\_cost, promo\_begin\_date FROM promotions WHERE promo\_cost LIKE 'post%' AND promo\_begin\_date < '01-01-2000';*

*no rows selected*

C,

*sh@TEST0924> SELECT promo\_name, promo\_cost, promo\_begin\_date FROM promotions WHERE promo\_category LIKE 'P%' AND promo\_begin\_date < '1-JANUARY-00'*

*2  ;*

*no rows selected*

D,

*sh@TEST0924> SELECT promo\_name, promo\_cost, promo\_begin\_date FROM promotions WHERE promo\_category LIKE '%post%' AND promo\_begin\_date < '1-JAN-00';*

*PROMO\_NAME                     PROMO\_COST PROMO\_BEGIN\_DATE*

*------------------------------ ---------- ------------------*

*post promotion #20-232                300 25-SEP-98*

*post promotion #21-166               2000 25-SEP-98*

*post promotion #20-449               4200 10-NOV-98*

54. View the Exhibit and examine the structure of the CUSTOMERS table.

Evaluate the query statement:

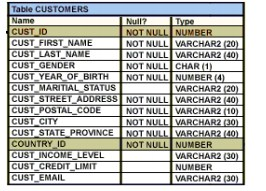
SQL> SELECT cust\_last\_name, cust\_city, cust\_credit\_limit

FROM customers

WHERE cust\_last\_name BETWEEN 'A' AND 'C' AND cust\_credit\_limit BETWEEN

1000 AND 3000;

What would be the outcome of the above statement?



A. It executes successfully.

B. It produces an error because the condition on CUST\_LAST\_NAME is invalid.

C. It executes successfully only if the CUST\_CREDIT\_LIMIT column does not contain any null values.

D. It produces an error because the  AND operator cannot be used to combine multiple BETWEEN

clauses.

Answer: A

运行上述查询语句

sh@TESTDB> SELECT cust\_last\_name, cust\_city, cust\_credit\_limit

  2  FROM customers

  3  WHERE cust\_last\_name BETWEEN 'A' AND 'C' AND cust\_credit\_limit BETWEEN

  4  1000 AND 3000 and rownum<6;

CUST\_LAST\_NAME                           CUST\_CITY                      CUST\_CREDIT\_LIMIT

---------------------------------------- ------------------------------ -----------------

Aaron                                    Pune                                        1500

Aaron                                    Diss                                        3000

Aaron                                    Hoofddorp                                   1500

Aaron                                    Stockport                                   3000

Aaron                                    Hyderabad                                   1500

55. Evaluate the following two queries:

SQL> SELECT cust\_last\_name, cust\_city

FROM customers

WHERE cust\_credit\_limit IN (1000, 2000, 3000);

SQL> SELECT cust\_last\_name, cust\_city

FROM customers

WHERE cust\_credit\_limit = 1000 OR cust\_credit\_limit = 2000 OR cust\_credit\_limit = 3000;

Which statement is true regarding the above two queries?

A. Performance would improve in query 2.

B. Performance would degrade in query 2.

C. There would be no change in performance.

D. Performance would improve in query 2 only if there are null values in the CUST\_CREDIT\_LIMIT

column.

Answer: C

答案解析：

sh@TESTDB> SELECT cust\_last\_name, cust\_city

  2  FROM customers

  3  WHERE cust\_credit\_limit IN (1000, 2000, 3000);

7975 rows selected.

Elapsed: 00:00:00.07

Execution Plan

----------------------------------------------------------

Plan hash value: 2008213504

-------------------------------------------------------------------------------

| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU)| Time     |

-------------------------------------------------------------------------------

|   0 | SELECT STATEMENT  |           | 20556 |   441K|   406   (1)| 00:00:05 |

|\*  1 |  TABLE ACCESS FULL| CUSTOMERS | 20556 |   441K|   406   (1)| 00:00:05 |

-------------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

   1 - filter("CUST\_CREDIT\_LIMIT"=1000 OR "CUST\_CREDIT\_LIMIT"=2000 OR

              "CUST\_CREDIT\_LIMIT"=3000)

sh@TESTDB> SELECT cust\_last\_name, cust\_city

  2  FROM customers

  3  WHERE cust\_credit\_limit = 1000 OR cust\_credit\_limit = 2000 OR cust\_credit\_limit = 3000;

7975 rows selected.

Elapsed: 00:00:00.08

Execution Plan

----------------------------------------------------------

Plan hash value: 2008213504

-------------------------------------------------------------------------------

| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU)| Time     |

-------------------------------------------------------------------------------

|   0 | SELECT STATEMENT  |           | 20556 |   441K|   406   (1)| 00:00:05 |

|\*  1 |  TABLE ACCESS FULL| CUSTOMERS | 20556 |   441K|   406   (1)| 00:00:05 |

-------------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

   1 - filter("CUST\_CREDIT\_LIMIT"=1000 OR "CUST\_CREDIT\_LIMIT"=2000 OR

              "CUST\_CREDIT\_LIMIT"=3000)

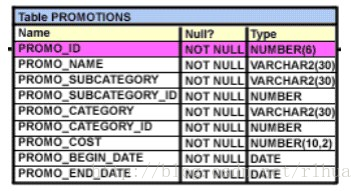
一样的执行计划，性能一样，故选C

56. View the Exhibit and examine the structure of the PROMOTIONS table.

Using the PROMOTIONS table,  you need to find out the names and cost of all the promos done on 'TV'

and 'internet' that ended in the time interval 15th March '00 to 15th October '00.

Which two queries would give the required result?   (Choose two.)



A. SELECT promo\_name,  promo\_cost

FROM promotions

WHERE  promo\_category IN ('TV', 'internet') AND

promo\_end\_date BETWEEN '15-MAR-00' AND '15-OCT-00';

B. SELECT promo\_name,  promo\_cost

FROM promotions

WHERE  promo\_category = 'TV' OR promo\_category ='internet' AND

promo\_end\_date >='15-MAR-00' OR promo\_end\_date <='15-OCT-00';

C. SELECT promo\_name,  promo\_cost

FROM promotions

WHERE   (promo\_category BETWEEN 'TV' AND 'internet') AND

(promo\_end\_date IN ('15-MAR-00','15-OCT-00'));

D. SELECT promo\_name,  promo\_cost

FROM promotions

WHERE  (promo\_category = 'TV' OR promo\_category ='internet') AND

  (promo\_end\_date >='15-MAR-00' AND promo\_end\_date <='15-OCT-00');

Answer: AD

答案解析：

题意要求done on 'TV'and 'internet' that ended in the time interval 15th March '00 to 15th October '00.

B错误，promo\_end\_date >='15-MAR-00' OR promo\_end\_date <='15-OCT-00' 应使用AND

C错误，promo\_category BETWEEN 'TV' AND 'internet'，条件不满足。

57. The CUSTOMERS table has the following structure:

name                     Null         Type

CUST\_ID    NOT NULL  NUMBER

CUST\_FIRST\_NAME   NOT NULL  VARCHAR2(20)

CUST\_LAST\_NAME   NOT NULL  VARCHAR2(30)

CUST\_INCOME\_LEVEL     VARCHAR2(30)

CUST\_CREDIT\_LIMIT     NUMBER

You need to write a query that  does the following tasks:

1. Display the first name and tax amount of the customers. Tax is 5% of their credit limit.

2. Only those customers whose income level has a value should be considered.

3. Customers whose tax amount is null should not be considered.

Which statement accomplishes all the required tasks?

A. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT

FROM customers

WHERE cust\_income\_level IS NOT NULL AND

tax\_amount IS NOT NULL;

B. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT

FROM customers

WHERE cust\_income\_level IS NOT NULL AND

cust\_credit\_limit IS NOT NULL;

C. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT

FROM customers

WHERE cust\_income\_level <> NULL AND

tax\_amount <> NULL;

D. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT

FROM customers

WHERE (cust\_income\_level,tax\_amount) IS NOT NULL;

Answer: B

答案解析：

A,WHERE子句不能跟别名

*sh@TEST0924> SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers*

*2  WHERE cust\_income\_level IS NOT NULL AND TAX\_AMOUNT IS NOT NULL*

*3  /*

*WHERE cust\_income\_level IS NOT NULL AND TAX\_AMOUNT IS NOT NULL*

*\**

*ERROR at line 2:*

*ORA-00904: "TAX\_AMOUNT": invalid identifier*

*sh@TEST0924> SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers*

*2  WHERE cust\_income\_level IS NOT NULL AND cust\_credit\_limit \* .05 IS NOT NULL；*

*CUST\_FIRST\_NAME      TAX\_AMOUNT*

*-------------------- ----------*

*Abigail                      75*

*Abigail                     350*

*...*

B,正确

*sh@TEST0924> SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers*

*2  WHERE cust\_income\_level IS NOT NULL AND cust\_credit\_limit IS NOT NULL；*

*CUST\_FIRST\_NAME      TAX\_AMOUNT*

*-------------------- ----------*

*Abigail                      75*

*Abigail                     350*

*...*

C错误，一是where子句不能用别名，二是非null不能用<>来表达

*sh@TEST0924> SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers*

*2  WHERE cust\_income\_level <> NULL AND tax\_amount <> NULL;*

*WHERE cust\_income\_level <> NULL AND tax\_amount <> NULL*

*\**

*ERROR at line 2:*

*ORA-00904: "TAX\_AMOUNT": invalid identifier*

*sh@TEST0924> SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers*

*2  WHERE cust\_income\_level <> NULL AND cust\_credit\_limit<> NULL;*

*no rows selected*

D错误，语法错误

*sh@TEST0924> SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers*

*2  WHERE (cust\_income\_level,tax\_amount) IS NOT NULL;*

*WHERE (cust\_income\_level,tax\_amount) IS NOT NULL*

*\**

*ERROR at line 2:*

*ORA-00920: invalid relational operator*

58. The PART\_CODE column in the SPARES table contains the following list of values:

PART\_CODE

A%\_WQ123

A%BWQ123

AB\_WQ123

Evaluate the following query:

SQL> SELECT part\_code

FROM spares

WHERE part\_code LIKE '%\%\_WQ12%' ESCAPE '\'; 都有% \_代替一个字符

Which statement is true regarding the outcome of the above query?

A. It produces an error.

B. It displays all values.

C. It displays only the values A%\_WQ123  and AB\_WQ123 .

D. It displays only the values A%\_WQ123  and A%BWQ123 .

E. It displays only the values A%BWQ123 and AB\_WQ123.

Answer: D

答案解析：

WHERE part\_code LIKE '%\%\_WQ12%' ESCAPE '\';

ESCAPE 将\为转义字符，故\后面的%原型输出，而%后面的\_,在SQL语句里代表一个替换字符。

故结果A%\_WQ123 和A%BWQ123

59. View the Exhibit and examine the data in the PRODUCTS table.

You need to display product names from the PRODUCTS table that belong to the 'Software/Other '

category  with minimum prices as either $2000  or $4000  and no unit of measure.

You issue the following query:

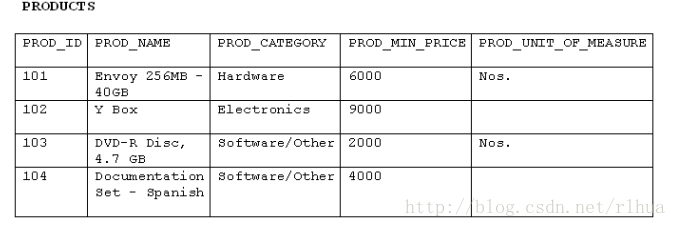
SQL>SELECT prod\_name, prod\_category, prod\_min\_price

FROM products

WHERE prod\_category LIKE '%Other%' AND (prod\_min\_price = 2000 OR

prod\_min\_price = 4000) AND prod\_unit\_of\_measure**<> '';**

Which statement is true regarding the above query?



A. It executes successfully but returns no result.

B. It executes successfully and returns the required result.

C. It generates an error because the condition specified for PROD\_UNIT\_OF\_MEASURE is not valid.

D. It generates an error because the condition specified for the  PROD\_CATEGORY column is not valid.

Answer: A

答案解析：

实验验证：

SELECT prod\_name, prod\_category, prod\_min\_price

  2  FROM products

  3  WHERE prod\_category LIKE '%Other%' AND (prod\_min\_price = 2000 OR

  4  prod\_min\_price = 4000) AND prod\_unit\_of\_measure <> '';

no rows selected

 prod\_unit\_of\_measure**<> ''; 应该改为is not null**

 因为prod\_unit\_of\_measure <> ''该条件为假，’’相当于NULL，所以会导致该WHERE子句里所有AND连接的值为假，

虽然执行成功但不会返回结果，正确的条件应该是prod\_unit\_of\_measure is not null。

60. View the Exhibit and examine the structure of CUSTOMERS table.

Evaluate the following query:

SQL>SELECT cust\_id, cust\_city

FROM customers

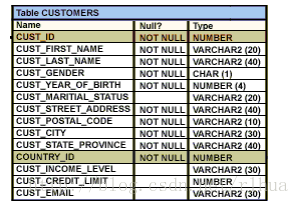
WHERE cust\_first\_name NOT LIKE 'A\_%g\_%' AND

cust\_credit\_limit BETWEEN 5000 AND 15000 AND

cust\_credit\_limit NOT IN (7000, 11000) AND

cust\_city NOT BETWEEN 'A' AND 'B';

Which statement is true regarding the above query?



A. It executes successfully.

B. It produces an error because the condition on the CUST\_CITY column is not valid.

C. It produces an error because the condition on the CUST\_FIRST\_NAME column is not valid.

D. It produces an error because conditions on the CUST\_CREDIT\_LIMIT column are not valid.

Answer: A

答案解析：

实验环境得出结果，可以正常执行。

此表在数据库的sh用户里。连接到sh用户，然后查询。因为满足条件的数量很多，故用rownum列出前5列。

sh@TESTDB> SELECT cust\_id, cust\_city

  2  FROM customers

  3  WHERE cust\_first\_name NOT LIKE 'A\_%g\_%' AND

4 cust\_credit\_limit BETWEEN 5000 AND 15000 AND

  5  cust\_credit\_limit NOT IN (7000, 11000) AND

  6  cust\_city NOT BETWEEN 'A' AND 'B'

  7  AND ROWNUM<6;

   CUST\_ID CUST\_CITY

---------- ------------------------------

     36117 Wolverhampton

     25470 Stuttgart

      4117 Clermont-l'Herault

     14784 Belfast City

     25451 Heilbronn

61. View the Exhibit and examine the structure of the PROMOTIONS table.

You need to generate a report of all promos from the PROMOTIONS table based on the following

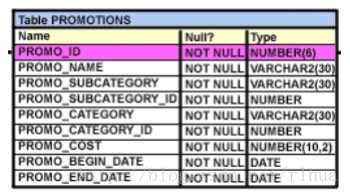
conditions:

1. The promo name should not begin with 'T' or 'N'.

2. The promo should cost more than $20000.  >20000

3. The promo should have ended after 1st January 2001.

Which WHERE clause would give the required result?



A. WHERE promo\_name NOT LIKE 'T%' OR promo\_name NOT LIKE 'N%' AND promo\_cost > 20000

AND promo\_end\_date > '1-JAN-01'

B. WHERE (promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%')OR promo\_cost > 20000

OR promo\_end\_date > '1-JAN-01'

C. WHERE promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%' AND promo\_cost > 20000

AND promo\_end\_date > '1-JAN-01'

D. WHERE (promo\_name NOT LIKE '%T%' OR promo\_name NOT LIKE '%N%') AND(promo\_cost >

20000 AND promo\_end\_date > '1-JAN-01')

Answer: C

答案解析：

因为三个条件是并行的，即要求都满足条件，所以都应该使用AND连接符。

A,promo\_name NOT LIKE 'T%' OR promo\_name NOT LIKE 'N%',OR 应改为AND

B,两个OR应该改为AND

C正确

D第一个OR应该改为AND

62. View the E xhibit and examine the structure of the CUSTOMERS table.

You want to generate a report showing the last names and credit limits of all customers whose last names

start with A, B,  or C, and credit limit is below 10, 000.

Evaluate the following two queries:

SQL> SELECT cust\_last\_name, cust\_credit\_limit FROM customers

WHERE (UPPER(cust\_last\_name) LIKE 'A%' OR

UPPER(cust\_last\_name) LIKE 'B%' OR UPPER(cust\_last\_name) LIKE 'C%')

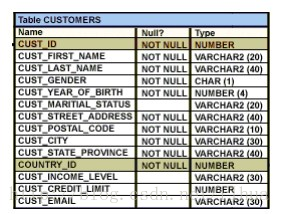
AND cust\_credit\_limit < 10000;

SQL>SELECT cust\_last\_name, cust\_credit\_limit FROM customers

WHERE UPPER(cust\_last\_name) BETWEEN 'A' AND 'C'

AND cust\_credit\_limit < 10000;

Which statement is true regarding the execution of the above queries?



A. Only the first query gives the correct result.

B. Only the second query gives the correct result.

C. Both execute successfully and give the same result.

D. Both execute successfully but do not give the required result.

Answer: A

 答案解析：题意要求start with A, B,  or C，而BETWEEN 'A' AND 'C' 包含A开头到C，不包含C开头以后的。

63. View the E xhibit and examine the structure of the PRODUCTS table.

You want to display only those product names with their list prices where the list price is at least double

the minimum price. The report should start with the product name having the maximum list price satisfying

this condition.

Evaluate the following SQL statement:

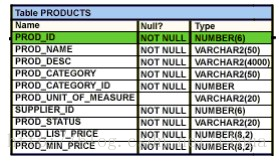
SQL>SELECT prod\_name,prod\_list\_price

FROM products

WHERE prod\_list\_price >= 2 \* prod\_min\_price

Which ORDER BY clauses  can be added to the above SQL statement to get the correct output?

(Choose all that apply.)



A. ORDER BY prod\_list\_price DESC, prod\_name;

B. ORDER BY (2\*prod\_min\_price)DESC, prod\_name;

C. ORDER BY prod\_name, (2\*prod\_min\_price)DESC;

D. ORDER BY prod\_name DESC, prod\_list\_price DESC;

E. ORDER BY prod\_list\_price DESC, prod\_name DESC;

Answer: AE

 答案解析：

The report should start with the product name having the maximum list price satisfying this condition.

题意要求按照list price由大到小附加名字来排序，即先保证 prod\_list\_price是降序的后面按照name来排序，prod\_name排序可升序可降序。

64. View the E xhibit and examine the data in the PROMO\_CATEGORY and PROMO\_COST columns of

the PROMOTIONS table.

Evaluate the following two queries:

SQL>SELECT DISTINCT promo\_category   to\_char(promo\_cost)"code"

FROM promotions

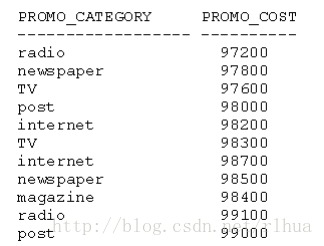
ORDER BY code;

SQL>SELECT DISTINCT promo\_category  promo\_cost "code"

FROM promotions

ORDER BY 1;

Which statement is true regarding the execution of the above queries?



A. Only the first query executes successfully.

B. Only the second query executes successfully.

C. Both queries execute successfully but give different results.

D. Both queries execute successfully and give the same result.

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12837221>

order by 可以指定表达式、别名或列位置作为排序条件，别名要完全匹配

列别名并没有完全匹配，所以报错。

*sh@TEST0924>  SELECT DISTINCT promo\_category, to\_char(promo\_cost)"code"*

*2  FROM promotions*

*3  ORDER BY code;*

*ORDER BY code*

*\**

*ERROR at line 3:*

*ORA-00904: "CODE": invalid identifier*

 应该改为：

*sh@TESTDB> SELECT DISTINCT promo\_category, to\_char(promo\_cost)"code"*

*2  FROM promotions*

*3  where rownum<6*

*4   ORDER BY "code";*

*PROMO\_CATEGORY                 code*

*------------------------------ ----------------------------------------*

*NO PROMOTION                   0*

*newspaper                      200*

*post                           300*

*newspaper                      400*

*internet                       600*

 第二个SQL如下，可执行成功，原题差一个逗号

*sh@TESTDB> SELECT DISTINCT promo\_category,promo\_cost "code"*

*2  FROM promotions*

*3  where rownum<6*

*4   ORDER BY 1;*

*PROMO\_CATEGORY                       code*

*------------------------------ ----------*

*NO PROMOTION                            0*

*internet                              600*

*newspaper                             200*

*newspaper                             400*

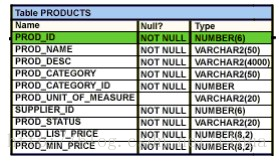
*post                                  300*

65. View the Exhibit and examine the structure of the CUSTOMERS table.

You have been asked to produce a report on the CUSTOMERS table showing the customers details

sorted in descending order of the city and in the descending order of their income level in each city.

Which query would accomplish this task?



A. SELECT cust\_city, cust\_income\_level, cust\_last\_name

FROM customers

ORDER BY cust\_city desc, cust\_income\_level DESC ;

B. SELECT cust\_city, cust\_income\_level, cust\_last\_name

FROM customers

ORDER BY cust\_income\_level desc, cust\_city DESC;

C. SELECT cust\_city, cust\_income\_level, cust\_last\_name

FROM customers

ORDER BY (cust\_city, cust\_income\_level) DESC;

D. SELECT cust\_city, cust\_income\_level, cust\_last\_name

FROM customers

ORDER BY cust\_city, cust\_income\_level DESC;

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12837221>

根据题意：sorted in descending order of the city and in the descending order of their income level in each city.

是按照城市的降序来排列，并且在每个城市中按照income level的降序来排列。

即先 cust\_city desc再cust\_income\_level DESC

66. View the Exhibit and examine the data in the COSTS table.

You need to generate a report that displays the IDs of all products in the COSTS table whose unit price is

at least 25% more than the unit cost. The details should be displayed in the descending order of 25% of

the unit cost.

You issue the following query:

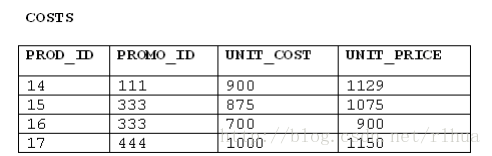
SQL>SELECT prod\_id

FROM costs

WHERE unit\_price >= unit\_cost \* 1.25

ORDER BY unit\_cost \* 0.25 DESC;

Which statement is true regarding the above query?



A. It executes and produces the required result.

B. It produces an error because an expression cannot be used in the ORDER BY clause.

C. It produces an error because the DESC option cannot be used with an expression in the ORDER BY

clause.

D. It produces an error because the expression in the ORDER BY clause should also be specified in the

SELECT clause.

Answer: A

答案解析：

Order by可以只用表达式来排序

order by 可以表达式sal\*10

               可以列别名 sal10

               可以双引号 "sal10"

sh@TESTDB> SELECT prod\_id

  2  FROM costs

  3  WHERE unit\_price >= unit\_cost \* 1.25

  4  and rownum<6

  5  ORDER BY unit\_cost \* 0.25 DESC;

   PROD\_ID

----------

        14

        14

        14

        14

        13

67. Which two statements are true regarding the ORDER BY clause? (Choose two.)

A. It is executed first in the query execution.

B. It must be the last clause in the SELECT statement.

C. It cannot be used in a SELECT statement containin g a HAVING clause.

D. You cannot specify a column name followed by an expression in this clause. order by可以用表达式

E. You can specify a combination of numeric positions and column names in this clause. 您可以指定数字的位置和在本节中列名的组合

Answer: BE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12837221>

A错，使用ORDER BY 子句，该子句必须是SQL 语句的最后一个子句。

B对，

C错，放在最后就可以了。

D错，可以指定表达式、别名或列位置或组合作为排序条件

E对，可以指定表达式、别名或列位置或组合作为排序条件

68. Which statement is true regarding the default behavior of the ORDER BY clause?

A. In a character sort,   the values are case- sensitive.

B. NULL values are not considered at all by the sort operation.

C. Only those columns that are specified in the SELECT list can be used in the ORDER BY clause.

D. Numeric values are displayed from the maximum to the minimum value if they have decimal positions.

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12837221>

A对，在字符排序中，是区分大小写的。

B错，对于升序排序，空值显示在最后，对于降序排序，则显示在最前面。即NULL是被考虑的

C错，可以指定不在select列中的字段来排序。

D错，不管是否小数，都是按照数值的大小来升序或降序排列。

69. You need to generate a list of all customer last names with their credit limits from the CUSTOMERS

table. Those customers who do not have a credit limit should appear last in the list.

Which two queries would achieve the required result? (Choose two.)

A. SELECT cust\_last\_name, cust\_credit\_limit

FROM customers

ORDER BY cust\_credit\_limit DESC ;

B. SELECT cust\_last\_name, cust\_credit\_limit

FROM customers

ORDER BY cust\_credit\_limit;

C. SELECT cust\_last\_name, cust\_credit\_limit

FROM customers

ORDER BY cust\_credit\_limit NULLS LAST;

D. SELECT cust\_last\_name, cust\_credit\_limit

FROM customers

ORDER BY cust\_last\_name, cust\_credit\_limit NULLS LAST;

Answer: BC

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12837221>

对于升序排序，空值显示在最后，对于降序排序，则显示在最前面

who do not have a credit limit should appear last in the list

题目要求空值在最后面

要么升序排列，要么指定nulls last

D先按照cust\_last\_name排序，在按照 cust\_credit\_limit空值最后排序，不符合题意。

实验验证：

B,

*scott@TESTDB> select \* from emp order by comm;*

*EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO*

*---------- ---------- --------- ---------- --------- ---------- ---------- ----------*

*7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30*

*7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30*

*7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30*

*7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30*

*7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20*

*7839 KING       PRESIDENT            17-NOV-81       5000                    10*

*7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20*

 C,

*scott@TESTDB> select \* from emp order by comm nulls last;*

*EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO*

*---------- ---------- --------- ---------- --------- ---------- ---------- ----------*

*7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30*

*7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30*

*7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30*

*7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30*

*7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20*

*7839 KING       PRESIDENT            17-NOV-81       5000                    10*

*7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20*

*14 rows selected.*

 D,

*scott@TESTDB> select \* from emp order by ename,comm nulls last;*

*EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO*

*---------- ---------- --------- ---------- --------- ---------- ---------- ----------*

*7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20*

*7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30*

*7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30*

*7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30*

*7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30*

*7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30*

*14 rows selected.*

70. View the E xhibit and examine the structure of the PRODUCTS table.

You want to display only those product names with their list prices where the list price is at least double

the minimum price. The report should start with the product name having the maximum list price satisfying

this condition.

Evaluate the following SQL statement:

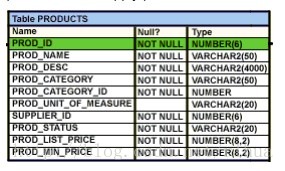
SQL>SELECT prod\_name,prod\_list\_price

FROM products

WHERE prod\_list\_price >= 2 \* prod\_min\_price

Which ORDER BY clauses  can be added to the above SQL statement to get the correct output?

(Choose all that apply.)



A. ORDER BY prod\_list\_price DESC, prod\_name;

B. ORDER BY (2\*prod\_min\_price)DESC, prod\_name;

C. ORDER BY prod\_name, (2\*prod\_min\_price)DESC;

D. ORDER BY prod\_name DESC, prod\_list\_price DESC;

E. ORDER BY prod\_list\_price DESC, prod\_name DESC;

Answer: AE

 答案解析：

The report should start with the product name having the maximum list price satisfying this condition.

题意要求按照list price由大到小附加名字来排序，即先保证 prod\_list\_price是降序的后面按照name来排序，prod\_name排序可升序可降序。

同63题：<http://blog.csdn.net/rlhua/article/details/12836675>

71. Which arithmetic operations can be performed on a column by using a  SQL function that is built into

Oracle database ? (Choose three .)

A. addition SUM

B. subtraction

C. raising   to a power

D. finding   the quotient

E. finding   the lowest value

Answer: ACE

答案解析：

题意问：. 哪一种算术运算能被Oracle数据库内置SQL函数在一个列上执行操作？（意思应该是哪一种算术运算可以被内置函数替代执行）（选择三个）

A. 加法 ,如SUM()

B. 减法 ,MINUS可以实现，但是它是集合运算符，不是SQL函数。

C. 乘方,如POWER()

D. 找商，在运算时可以使用/，但这不是SQL函数。

E. 找最小值,MIN()

72. Which tasks can be performed using SQL functions built into Oracle Database ? (Choose three.)

A. displaying a date in a nondefault format

B. finding the number of characters in an  expression

C. substituting a character string in a text expression with a specified string

D. combining more than two  columns or expressions into a single column in the output

Answer: ABC

 答案解析：

题意问：内置到Oracle数据库使用SQL函数可以执行哪些任务？ （选择三项）。

A.显示一个非默认格式的日期，例如：to\_char()转换日期输出

B.在一个表达式中查找字符的数量，使用REGEXP\_COUNT 函数。REGEXP\_COUNT 返回在源串中出现的模式的次数。

C.使用指定的字符串来替换文本表达式中的字符串，replace函数。

D.联合超过两个列或表达式输出为一个列，contact只能连接两列。||可以连接多个，但它不是SQL函数。

73. Which tasks can be performed using SQL functions that are built into Oracle database ? (Choose

three .)

A. finding   the remainder of a division

B. adding  a number to a date for a resultant date value

C. comparing   two expressions to check whether   they are equal

D. checking  whether  a specified character exists in a given string

E. removing   trailing, leading,  and  embedded characters from a character string

Answer: ACD

答案解析：

A. 取余，如MOD()

B. 给日期添加一个数字合成一个日期值,个人觉得可以使用ADD\_MONTHS()，含义上也是一个日期加上一个数字等于另外一个日期。

C. 比较两个表达式查看是否相等,例如NULLIF

D. 检查指定的字符串是否存在另一个字符串中,例如，INSTR()

E. 从一个字符串中移除尾部，前部，内含的字符,trim()只能去除前后的字符

74. Which statements are true regarding single row functions? (Choose all that apply.)

A. MOD : returns the quotient of a division

B. TRUNC : can be used with NUMBER and DATE values

C. CONCAT : can be used to combine any number of values

D. SYSDATE : returns the database server current date and time

E. INSTR : can be used to find only the first occurrence of a character in a string

F. TRIM : can be used to remove all the occurrences of a character from a string

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

A.MOD:返回除法运算的余数

B.TRUNC：返回除法运算的余数，可用于数字和日期。

C.CONCAT：将值联接在一起（CONCAT 函数中只能使用两个参数）

D.SYSDATE：返回当前数据库服务器的日期和时间

E.INSTR：查找指定字符串的数字位置

F.TRIM：截去字符串首字符或尾字符（或者两者都截去）（如果trim\_character或trim\_source 是一个字符文字，则必须将其放在单引号内）

由上可看，A，返回的应为除法运算的余数而不是商，所以错误；B TRUNC可用于数字和日期，正确；C CONCAT 只能连接两个值，所以错误；D，SYSDATE 返回当前数据库服务器的日期和时间，正确；E，INSTR 查找指定字符串的数字位置，而不是题中所描述的，F, TRIM 截去字符串首字符或尾字符（或者两者都截去），而题中所描述的为replace函数。

75. The following data exists in the PRODUCTS table:

PROD\_ID     PROD\_LIST\_PRICE

123456        152525.99

You issue the following query:

SQL> SELECT RPAD(( ROUND(prod\_list\_price)), 10,'\*')

FROM products

WHERE prod\_id = 123456;

What would be the outcome?

A. 152526 \*\*\*\*

B. \*\*152525.99

C. 152525\*\* \*\*

D. an error message

Answer: A

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

Round( ) 函数

传回一个数值，该数值是按照指定的小数位元数进行四舍五入运算的结果

语法

ROUND( number, decimal\_places )

number : 需四舍五入处理的数值

decimal\_places : 四舍五入 , 小数取几位 ( 预设为 0 )

 RPAD：返回一个表达式，右边使用一个字符表达式填充到n 个字符的长度。

实验验证：

scott@TESTDB> create table products(prod\_id number,prod\_list\_price number);

Table created.

scott@TESTDB> insert into products values(123456,152525.99);

1 row created.

scott@TESTDB> select \* from  products;

   PROD\_ID PROD\_LIST\_PRICE

---------- ---------------

    123456       152525.99

scott@TESTDB> select round(PROD\_LIST\_PRICE) from products;

ROUND(PROD\_LIST\_PRICE)

----------------------

                152526

scott@TESTDB> select rpad((round(PROD\_LIST\_PRICE)),10,'\*') from products;

RPAD((ROUND(PROD\_LIS

--------------------

152526\*\*\*\*

76. You need to display the first names of all customers from the CUSTOMERS table that contain the

character 'e' and have the character 'a' in the second last position.

Which query would give the required output?

A. SELECT cust\_first\_name

FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>0 AND         e存在

SUBSTR(cust\_first\_name, -2, 1)='a'; 倒数第二个是a

B. SELECT cust\_first\_name

FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>'' AND

SUBSTR(cust\_first\_name, -2, 1)='a';

C. SELECT cust\_first\_name

FROM customers

WHERE INSTR(cust\_first\_name, 'e')IS NOT NULL  AND

SUBSTR(cust\_first\_name, 1,-2)='a';

D. SELECT cust\_first\_name

FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>0 AND

SUBSTR(cust\_first\_name, LENGTH(cust\_first\_name),-2)='a';

Answer: A

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

SUBSTR：提取确定长度的字符串

SUBSTR(cust\_first\_name, -2, 1)='a'意思是cust\_first\_name从右边开始数，第二个字符是a

INSTR：查找指定字符串的数字位置，如果没有找到，instr函数返回0

 INSTR(cust\_first\_name, 'e')确定是否有e这个字符，有则输出器位置，没有则输出0,。

则A正确。

INSTR(源字符串, 目标字符串, 起始位置, 匹配序号)

　　在oracle/PLSQL中，instr函数返回要截取的字符串在源字符串中的位置。只检索一次，就是说从字符的开始到字符的结尾就结束。

　　语法如下：instr( string1, string2 [, start\_position [, nth\_appearance ] ] )

　　参数分析：string1源字符串，要在此字符串中查找。

　　string2要在string1中查找的字符串.

　　start\_position代表string1 的哪个位置开始查找。此参数可选，如果省略默认为1. 字符串索引从1开始。如果此参数为正，从左到右开始检索，如果此参数为负，从右到左检索，返回要查找的字符串在源字符串中的开始索引。

　　nth\_appearance代表要查找第几次出现的string2. 此参数可选，如果省略，默认为 1.如果为负数系统会报错。

　　注意：如果String2在String1中没有找到，instr函数返回0.

77.  In  the CUSTOMERS  table,  the  CUST\_CITY  column  contains  the  value  'Paris'  for  the

CUST\_FIRST\_NAME 'ABIGAIL'.

Evaluate the following query:

SQL> SELECT INITCAP(cust\_first\_name   ' '

UPPER(SUBSTR(cust\_city,-LENGTH(cust\_city),2)))

FROM customers

WHERE cust\_first\_name = 'ABIGAIL';

What would be the outcome?

A. Abigail PA

B. Abigail Pa

C. Abigail IS

D. an error message

Answer: B

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

此为嵌套函数，由里到外可知：

LENGTH(cust\_city)=LENGTH('Paris')=5

SUBSTR(cust\_city,-5,2)=从右往左第5个，在顺着输出2个字符，即Pa

UPPER(Pa)=PA

INITCAP(cust\_first\_name   ' ' UPPER(SUBSTR(cust\_city,-LENGTH(cust\_city),2)))=INITCAP('ABIGAIL' '' 'PA')=Abigail Pa

即选B。

78. Evaluate the following query:

SQL> SELECT TRUNC(ROUND(156.00,-1),-1)

FROM DUAL;

What would be the outcome?

A. 16

B. 100

C. 160

D. 200

E. 150

Answer: C

  答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

ROUND(156.00,-1)=160，四舍五入，精确到十位

TRUNC(160，-1)=160，截断精确到十位。

实验验证：

scott@TESTDB> select round(156.00,-1) from dual;

ROUND(156.00,-1)

----------------

             160

scott@TESTDB> select trunc(160,-1) from dual;

TRUNC(160,-1)

-------------

          160

79. View the Exhibit and examine the structure of the CUSTOMERS table.

In the CUSTOMERS table, the CUST\_LAST\_NAME column contains the values 'Anderson' and 'Ausson'.

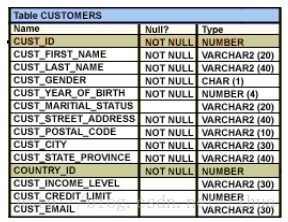
You issue the following query:

SQL> SELECT LOWER(REPLACE(TRIM('son' FROM cust\_last\_name),'An','O'))

FROM CUSTOMERS

WHERE LOWER(cust\_last\_name) LIKE 'a%n';

What would be the outcome?



A. 'Oder' and 'Aus'

B. a n error because the  TRIM function specified is not valid

C. a n error because the  LOWER function specified is not valid

D. a n error because the  REPLACE function specified is not valid

Answer: B

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

TRIM：截去字符串首字符或尾字符（或者两者都截去）（如果trim\_character或trim\_source 是一个字符文字，则必须将其放在单引号内）

实验验证：

sh@TESTDB> select trim('son' from cust\_last\_name) from customers ;

select trim('son' from cust\_last\_name) from customers

       \*

ERROR at line 1:

ORA-30001: trim set should have only one character

sh@TESTDB> select trim('s' from cust\_last\_name) from customers where rownum<3;

TRIM('S'FROMCUST\_LAST\_NAME)

----------------------------------------

Ruddy

Ruddy

sh@TESTDB> select trim('sa' from 'saabd') from dual;

select trim('sa' from 'saabd') from dual

       \*

ERROR at line 1:

ORA-30001: trim set should have only one character

sh@TESTDB> select trim('s' from 'saabd') from dual;

TRIM

----

aabd

80. Which two statements are true regarding working with dates? (Choose two.)

A. The default internal storage of dates is in the  numeric format.

B. The default internal storage of dates is in the character format.

C. The RR date format automatically calculates the century from the SYSDATE function and  does not

allow the user to enter the century.

D. The RR date format automatically calculates the century from the SYSDATE function but allows the

user to enter the century if required.

Answer: AD

答案解析：

AB,默认内部日期的存储形式是数字形式,所有A对，B错。

A参照Database Concepts的DATE Data Type部分：

The database stores dates internally as numbers. Dates are stored in fixed-length fields of 7 bytes each, corresponding to century, year, month, day, hour, minute, and second

CD,RR日期格式自动从SYSDATE函数中计算出世纪，但是如果需要还允许用户输入世纪,所有C错，D对。

81. You are currently located in Singapore and have connected to a remote database in Chicago.

You issue the following command:

SQL> SELECT ROUND(SYSDATE-promo\_begin\_date,0)

FROM promotions

WHERE (SYSDATE-promo\_begin\_date)/365 > 2;

PROMOTIONS is the public synonym for the public database link for the PROMOTIONS table.

What is the outcome?

A. a n error because the ROUND function specified is invalid

B. a n error because the WHERE condition specified is invalid

C. number of days since the promo started based on the current Chicago date and time

D. number of days since the promo started based on the current  Singapore date and time

Answer: C

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

日期-日期，返回的是天数，是一个数字，

由本地连接到远程，SYSDATE是一个 SQL 函数，它用于返回当前所连接的远程服务器上数据库的日期和时间。

故选C.

82. Examine the data in the CUST\_NAME column of the CUSTOMERS table.

CUST\_NAME

Renske Ladwig

Jason Mallin

Samuel McCain

Allan MCEwen

Irene Mikkilineni

Julia Nayer

You need to display customers' second names where the second name starts with "Mc" or "MC."

Which query gives the required output?

A. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)

FROM customers

WHERE INITCAP(SUBSTR(cust\_name,INSTR(cust\_name,' ')+1))='Mc';

B. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)

FROM customers

WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)) LIKE 'Mc%';

C. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)

FROM customers

WHERE SUBSTR(cust\_name, INSTR(cust\_name,' ')+1) LIKE INITCAP('MC%');

D. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)

FROM customers

WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)) = INITCAP('MC%');

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

题意要求：You need to display customers' second names where the second name starts with "Mc" or "MC."

cust\_name由first name和second name组成，中间有空格，要求找出second name以"Mc" or "MC."开头

INSTR(cust\_name,' ')找出空格的位置，并返回，这个位置之后的second name的开始。

SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)找出空格+1的之后的字符，即是找出second name的名字。

INITCAP(SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)) LIKE 'Mc%'  匹配出 LIKE 'Mc%'or "MC“的字符，使用INITCAP，让其匹配。

故答案选B

83. Examine the data in the CUST\_NAME column of the CUSTOMERS table.

CUST\_NAME

Lex De Haan

Renske Ladwig

Jose Manuel Urman

Jason Mallin

You want to extract only those customer names that have three names and display the \* symbol in place

of the first name as follows:

CUST NAME

\*\*\* De Haan

\*\*\*\* Manuel Urman

Which two queries give the required output? (Choose two.)

A. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*') "CUST NAME"

FROM customers

WHERE INSTR(cust\_name, ' ',1,2)<>0;

B. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*') "CUST NAME"

FROM customers

WHERE INSTR(cust\_name, ' ',-1,2)<>0;

C.  SELECT  LPAD(SUBSTR(cust\_name,INSTR(cust\_name,'  ')),LENGTH(cust\_name)-

INSTR(cust\_name,' '),'\*') "CUST NAME"

FROM customers

WHERE INSTR(cust\_name, ' ',-1,-2)<>0;

D.  SELECT  LPAD(SUBSTR(cust\_name,INSTR(cust\_name,'  ')),LENGTH(cust\_name)-

INSTR(cust\_name,' '),'\*') "CUST NAME"

FROM customers

WHERE INSTR(cust\_name, ' ',1,2)<>0 ;

Answer: AB

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

题意要求找出三个名字的，并且将第一个名字使用\*替代。

**instr( string1, string2, start\_position,nth\_appearance )**

|  |  |
| --- | --- |
| string1 | 源字符串，要在此字符串中查找。 |
| string2 | 要在string1中查找的字符串 。 |
| start\_position | 代表string1 的哪个位置开始查找。此参数可选，如果省略默认为1. 字符串索引从1开始。如果此参数为正，从左到右开始检索，如果此参数为负，从右到左检索，返回要查找的字符串在源字符串中的开始索引。 |
| nth\_appearance | 代表要查找第几次出现的string2. 此参数可选，如果省略，默认为 1.如果为负数系统会报错。 |

INSTR(cust\_name, ' ',1,2)<>0和INSTR(cust\_name, ' ',-1,2)<>0确定空格第二次出现的位置能找到，即不等于0.

也即是说有三个名字。

C的where条件不对。

LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*')

即是INSTR(cust\_name,' ')找出第一个空格的位置。

 SUBSTR ：提取确定长度的字符串

SUBSTR(cust\_name,INSTR(cust\_name,' '))输出第一个空格之后的字符，即是输出第二个名字和第三个名字

LPAD：返回一个表达式，左边使用一个字符表达式填充到n 个字符的长度

 LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*')根据cust\_name，来让\*替代第一个名字。

84. View the Exhibit and examine the structure of the EMPLOYEES table.

Examine the data in the ENAME and HIREDATE columns of the EMPLOYEES table:

ENAME       HIREDATE

SMITH       17-DEC-80

ALLEN       20-FEB-81

WARD        22-FEB-81

You want to generate a list of user IDs as follows:

USERID

Smi17DEC80

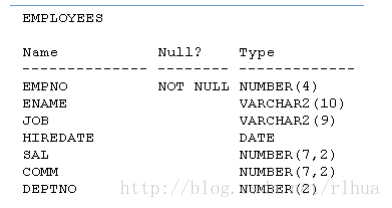
All20FEB81

War22FEB81

You issue the following query:

SQL>SELECT CONCAT(SUBSTR(INITCAP(ename),1,3), REPLACE(hiredate,'-')) "USERID" FROM employees;

What is the outcome?



A. It executes successfully and gives the correct output.

B. It executes successfully but does not give the correct output.

C. It generates an error because the REPLACE function is not valid.

D. It generates an error because the SUBSTR function cannot be nested in the CONCAT function.

Answer: A

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

题意输出的要求是名字的前三位并且首字母大写，日期不要短横线。

SELECT CONCAT(SUBSTR(INITCAP(FIRST\_NAME),1,3), REPLACE(HIRE\_DATE,'-')) "USERID" FROM employees;

 按照hr用户的employees来实验验证：

hr@TEST0924> select FIRST\_NAME,HIRE\_DATE from employees;

FIRST\_NAME           HIRE\_DATE

-------------------- ------------------

Donald               21-JUN-07

Douglas              13-JAN-08

Jennifer             17-SEP-03

REPLACE(HIRE\_DATE,'-')去掉日期中间的短横线，INITCAP(FIRST\_NAME)首字母大写，(SUBSTR(INITCAP(FIRST\_NAME),1,3)去前三位，并且首字母大写，

CONCAT(SUBSTR(INITCAP(FIRST\_NAME),1,3), REPLACE(HIRE\_DATE,'-'))两个字符串连接。

结果如下：

hr@TEST0924> SELECT CONCAT(SUBSTR(INITCAP(FIRST\_NAME),1,3), REPLACE(HIRE\_DATE,'-')) "USERID" FROM employees;

USERID

------------------------------

Don21JUN07

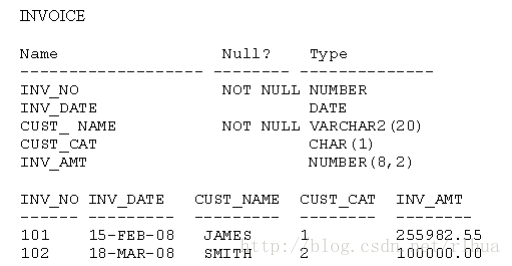
Dou13JAN08

Jen17SEP03

85. View the E xhibit and examine the structure and data in the INVOICE table.

Which statements are true regarding data type conversion in expressions used in queries?  (Choose all

that apply.)



A. inv\_amt ='0255982'   : requires explicit conversion

B. inv\_date > '01-02-2008'   : uses implicit conversion

C. CONCAT(inv\_amt,inv\_date)   : requires explicit conversion

D. inv\_date = '15-february-2008'    :  uses implicit conversion

E. inv\_no BETWEEN '101' AND '110'   :  uses implicit conversion

Answer: DE

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832019>

<http://blog.csdn.net/rlhua/article/details/12832885>

在**表达式**中，Oracle Server 可以自动执行以下转换：

**从**VARCHAR2 或CHAR隐式转换为NUMBER

**从VARCHAR2 或CHAR隐式转换为**DATE

A隐式转换为数值型，不用显示转换，错误。

B错误，只能用to\_date显示转换为特定的格式。

C错误，隐式转换为字符型，如下。

scott@TESTDB> select concat(sal,hiredate) from emp;

CONCAT(SAL,HIREDATE)

----------------------------------------------------

80017-DEC-80

160020-FEB-81

125022-FEB-81

297502-APR-81

 D正确，此格式玉默认的格式相融合，可隐式转换

E正确，可隐式转换为数值型。

86. Examine the structure and data of the CUST\_TRANS table:

CUST\_TRANS

Name       Null           Type

CUSTNO      NOT NULL       CHAR(2)

TRANSDATE                 DATE

TRANSAMT                  NUMBER(6,2)

CUSTNO TRANSDATE       TRANSAMT

11      01-JAN-07       1000

22      01-FEB-07       2000

33      01-MAR-07       3000

Dates are stored in the default date format dd-mon-rr in the CUST\_TRANS table.

Which SQL statements would execute successfully? (Choose three .)

A. SELECT transdate + '10' FROM cust\_trans;

B. SELECT \* FROM cust\_trans WHERE transdate  =   '01-01-07';

C. SELECT transamt FROM cust\_trans WHERE custno > '11';

D. SELECT \* FROM cust\_trans WHERE transdate='01-JANUARY-07';

E. SELECT custno + 'A' FROM cust\_trans WHERE transamt > 2000;

Answer: ACD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832019>

对于**表达式计算**，Oracle Server 可以自动执行以下转换：

**从NUMBER隐式转换为VARCHAR2 或CHAR**

**从 DATE隐式转换为VARCHAR2 或CHAR**

A，日期隐式转换为字符型。正确

B,  '01-01-07'为特定格式，需要用to\_date函数转换。

C，custno 为字符，字符与字符相比较，正确。

D, 01-JAN-07 等同于01-JANUARY-07，不需要转换函数，可隐式转换。正确。

E，错误，不能使用+。

87. You want to display the date for the first Mon day of the next  month and issue the following

command:

SQL>SELECT TO\_CHAR(NEXT\_DAY(LAST\_DAY(SYSDATE),'MON'),

‘dd "is the first Monday for" fmmonth rrrr')

 FROM DUAL;

What is the outcome?

A. It executes successfully and returns the correct result.

B. It executes successfully but does not return the correct result.

C. It generates an error because TO\_CHAR should be replaced with TO\_DATE.

D. It generates an error because rrrr should be replaced by rr in the format string.

E. It generates an error because fm and double quotation marks should not be used in the format string.

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12848395>

<http://blog.csdn.net/rlhua/article/details/12832885>

题意要输出下个月第一个星期一。

实验验证：

下个月的星期一是4号。由结果得知可以正确执行，并且结果是正确的。

*scott@TEST0924>  SELECT TO\_CHAR(NEXT\_DAY(LAST\_DAY(SYSDATE),'MON'),*

*2    'dd "is the first Monday for" fmmonth rrrr')*

*3   from dual;*

*TO\_CHAR(NEXT\_DAY(LAST\_DAY(SYSDATE),'MON'),'DD"ISTHEFIRSTMONDAYFOR"FM*

*--------------------------------------------------------------------*

*04 is the first Monday for november 2013*

SYSDATE输出当前日期，即18-OCT-13

*scott@TEST0924> select sysdate from dual;*

*SYSDATE*

*------------------*

*18-OCT-13*

LAST\_DAY(SYSDATE)输出当月最后一天，即10月31号。

 scott@TEST0924> select LAST\_DAY(SYSDATE) from dual;

*LAST\_DAY(SYSDATE)*

*------------------*

*31-OCT-13*

NEXT\_DAY(LAST\_DAY(SYSDATE)，输出指定日期的下一个日期，指定的日期是星期一，即是11月4号。

*scott@TEST0924> SELECT NEXT\_DAY(LAST\_DAY(SYSDATE),'MON') FROM DUAL;*

*NEXT\_DAY(LAST\_DAY(*

*------------------*

*04-NOV-13*

最后以特定的格式输出。

88. You need to calculate the number of days from 1st January 2007 till date.

Dates are stored in the default format of dd-mon-rr.

Which SQL statements would give the required output? (Choose two .)

A. SELECT SYSDATE - '01-JAN-2007' FROM DUAL;

B. SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;

C. SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL;

D. SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;

E. SELECT TO\_DATE(SYSDATE, 'DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL;

Answer: BC

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832885>

日期与字符串做表达式计算，隐式将日期转换为字符串，而字符串之间不能相减，所以ADE都是字符之间相减，所以错误。

BC之间相减是日期与日期之间的减，得出数字，是正确的。

89. You need to display the date 11-oct-2007 in words as 'Eleventh of October, Two Thousand Seven'.

Which SQL statement would give the required result?

A. SELECT TO\_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year')

FROM DUAL;

B. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdspth of month, year')

FROM DUAL;

C. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdthsp "of" Month, Year')

FROM DUAL;

D. SELECT TO\_DATE(TO\_CHAR('11-oct-2007','fmDdspth ''of'' Month, Year'))  错。

FROM DUAL;

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832885>

A错误，'11-oct-2007'本来就是字符，就不用再to\_char

B中'fmDdspth of month, year'的of需要用双引号引起来。

C正确。

D跟A一样的错误，'11-oct-2007'本来就是字符，就不用再to\_char。

实验验证：

A,

*scott@TEST0924> SELECT TO\_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year') FROM DUAL;*

*SELECT TO\_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year') FROM DUAL*

*\**

*ERROR at line 1:*

*ORA-01722: invalid number*

B

*scott@TEST0924> SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdspth of month, year') FROM DUAL;*

*SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdspth of month, year') FROM DUAL*

*\**

*ERROR at line 1:*

*ORA-01821: date format not recognized*

C

*scott@TEST0924> SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdthsp "of" Month, Year') FROM DUAL;*

*TO\_CHAR(TO\_DATE('11-OCT-2007'),'FMDDTHS*

*---------------------------------------*

*Eleventh of October, Two Thousand Seven*

D

*scott@TEST0924> SELECT TO\_DATE(TO\_CHAR('11-oct-2007','fmDdspth ''of'' Month, Year')) FROM DUAL;*

*SELECT TO\_DATE(TO\_CHAR('11-oct-2007','fmDdspth ''of'' Month, Year')) FROM DUAL*

*\**

*ERROR at line 1:*

*ORA-01722: invalid number*

90. Examine the structure and data in the PRICE\_LIST table:

name             Null       Type

PROD\_ID         NOT NULL     NUMBER(3)

PROD\_PRICE                  VARCHAR2(10)

PROD\_ID    PROD\_PRICE

100        $234.55

101        $6,509.75

102        $1,234

You plan to give a discount of 25% on the product price and need to display the discount amount in the

same format as the PROD\_PRICE.

Which SQL statement would give the required result?

A. SELECT TO\_CHAR(prod\_price\* .25,'$99,999.99')

FROM PRICE\_LIST;

B. SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\* .25,'$99,999.00')

FROM PRICE\_LIST;

C. SELECT TO\_CHAR(TO\_NUMBER(prod\_price,'$99,999.99')\* .25,'$99,999.00')

FROM PRICE\_LIST;

D. SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'$99,999.99')\* .25,'$99,999.00')

FROM PRICE\_LIST;

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832885>

实验环境：

*scott@TEST0924> create table price\_list*

*2   (prod\_id number(3) not null,*

*3   prod\_price varchar2(10));*

*Table created.*

*scott@TEST0924>  insert into price\_list values(100,'$234.55');*

*1 row created.*

*scott@TEST0924>  insert into price\_list values(101,'$6,509.75');*

*1 row created.*

*scott@TEST0924>  insert into price\_list values(102,'$1,234');*

*1 row created.*

*scott@TEST0924>  select \* from price\_list;*

*PROD\_ID PROD\_PRICE*

*---------- ----------*

*100 $234.55*

*101 $6,509.75*

*102 $1,234*

A，prod\_price为字符，不能\*.25

*scott@TEST0924> SELECT TO\_CHAR(prod\_price\* .25,'$99,999.99') FROM PRICE\_LIST;*

*SELECT TO\_CHAR(prod\_price\* .25,'$99,999.99') FROM PRICE\_LIST*

*\**

*ERROR at line 1:*

*ORA-01722: invalid number*

B，TO\_NUMBER(prod\_price)里面需要转换格式

*scott@TEST0924>  SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\* .25,'$99,999.00') FROM PRICE\_LIST;*

*SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\* .25,'$99,999.00') FROM PRICE\_LIST*

*\**

*ERROR at line 1:*

*ORA-01722: invalid number*

C正确

*scott@TEST0924> SELECT TO\_CHAR(TO\_NUMBER(prod\_price,'$99,999.99')\* .25,'$99,999.00')  FROM PRICE\_LIST;*

*TO\_CHAR(TO\_*

*-----------*

*$58.64*

*$1,627.44*

*$308.50*

D，本来(TO\_NUMBER(prod\_price,'$99,999.99')\* .25已经是数值型，就不用再TO\_NUMBER转换一回了。

*scott@TEST0924> SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'$99,999.99')\* .25,'$99,999.00') FROM PRICE\_LIST;*

*SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'$99,999.99')\* .25,'$99,999.00') FROM PRICE\_LIST*

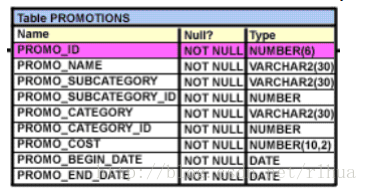
*\**

*ERROR at line 1:*

*ORA-01722: invalid number*

91. View the Exhibit and examine the structure of the PROMOTIONS table.

Which two SQL statements would execute successfully? (Choose two.)



A. UPDATE promotions

SET promo\_cost = promo\_cost+ 100

WHERE TO\_CHAR(promo\_end\_date, 'yyyy') > '2000';

B. SELECT promo\_begin\_date

FROM promotions

WHERE TO\_CHAR(promo\_begin\_date,'mon dd yy')='jul 01 98';

C. UPDATE promotions

SET promo\_cost = promo\_cost+ 100

WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8));

D. SELECT TO\_CHAR(promo\_begin\_date,'dd/month')

FROM promotions

WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'));

Answer: AB

答案解析：

A答案：

sh@TEST0924> UPDATE promotions

2 SET promo\_cost = promo\_cost+ 100

3 WHERE TO\_CHAR(promo\_end\_date, 'yyyy') > '2000';

42 rows updated.

B答案：

sh@TEST0924> SELECT promo\_begin\_date

2 FROM promotions

3 WHERE TO\_CHAR(promo\_begin\_date,'mon dd yy')='jul 01 98';

PROMO\_BEG

---------

01-JUL-98

01-JUL-98

01-JUL-98

01-JUL-98

01-JUL-98

01-JUL-98

6 rows selected.

C答案：格式不匹配，需要使用格式符转换

sh@TEST0924> UPDATE promotions

2 SET promo\_cost = promo\_cost+ 100

3 WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8));

WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8))

\*

ERROR at line 3:

ORA-01861: literal does not match format string

sh@TEST0924> select SUBSTR('01-JAN-2000',8) from dual;

SUBS

----

2000

sh@TEST0924> UPDATE promotions

2 SET promo\_cost = promo\_cost+ 100

3 WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8**),'yyyy'**);

84 rows updated.

D答案：TO\_DATE转换的日期如果与默认格式不一样，需要使用格式符转换

h@TEST0924> SELECT TO\_CHAR(promo\_begin\_date,'dd/month')

2 FROM promotions

3 WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'));

WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'))

\*

ERROR at line 3:

ORA-01858: a non-numeric character was found where a numeric was expected

sh@TEST0924> SELECT TO\_CHAR(promo\_begin\_date,'dd/month')

2 FROM promotions

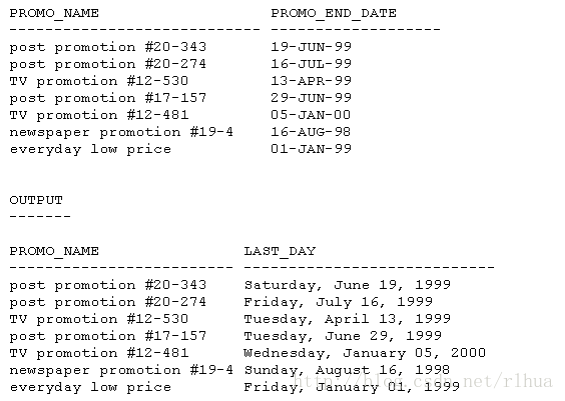
3 WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98',**'mon dd yy'**),TO\_DATE('JUL 01 98',**'mon dd yy'**));

no rows selected

92. View the E xhibit and examine the data in the PROMO\_NAME and PROMO\_END\_DATE columns of

the PROMOTIONS table, and the required output format.

Which two queries give the correct result? (Choose two.)



A. SELECT promo\_name, TO\_CHAR(promo\_end\_date,'Day')   ', '

TO\_CHAR(promo\_end\_date,'Month')   ' '

TO\_CHAR(promo\_end\_date,'DD, YYYY') AS last\_day

FROM promotions;

B. SELECT promo\_name,TO\_CHAR (promo\_end\_date,'fxDay')   ', '

TO\_CHAR(promo\_end\_date,'fxMonth')   ' '

TO\_CHAR(promo\_end\_date,'fxDD, YYYY') AS last\_day

FROM promotions;

C. SELECT promo\_name, TRIM(TO\_CHAR(promo\_end\_date,'Day'))   ', '

TRIM(TO\_CHAR(promo\_end\_date,'Month'))   ' '

TRIM(TO\_CHAR(promo\_end\_date,'DD, YYYY')) AS last\_day

FROM promotions;

D. SELECTpromo\_name,TO\_CHAR(promo\_end\_date,'fmDay')','

TO\_CHAR(promo\_end\_date,'fmMonth')   ' '

TO\_CHAR(promo\_end\_date,'fmDD, YYYY') AS last\_day

FROM promotions;

Answer: CD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12832885>

fm元素，用于删除填充的空格或隐藏前导零，TRIM也是截去字符串首字符或尾字符（或者两者都截去）

所有选CD

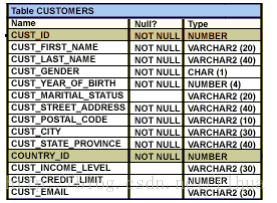
93. View the Exhibit and examine the structure of the CUSTOMERS table.

Using the CUSTOMERS table, y ou need to generate a report that shows  an increase in the credit limit

by 15% for all customers. Customers whose credit limit has not been entered should have the message "

Not Available"  displayed.

Which SQL statement would produce   the required result?



A. SELECT NVL(cust\_credit\_limit,'Not Available')\*.15 "NEW CREDIT"

FROM customers;

B. SELECT NVL(cust\_credit\_limit\*.15,'Not Available') "NEW CREDIT"

FROM customers;

C. SELECT TO\_CHAR(NVL(cust\_credit\_limit\*.15,'Not Available')) "NEW CREDIT"

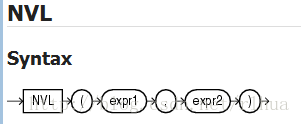
FROM customers;

D. SELECT NVL(TO\_CHAR(cust\_credit\_limit\*.15),'Not Available') "NEW CREDIT"

FROM customers;

Answer: D

**1、NVL**



官方解释：

Purpose

NVL lets you replace null (returned as a blank) with a string in the results of a query.

If expr1 is null, then NVL returns expr2. Ifexpr1 is not null, then NVL returns expr1.

如果expr1是null，则返回expr2，如果expr1 is not null，则返回expr1.

The arguments expr1 and expr2 can have any data type. If their data types are different, then Oracle Database implicitly converts one to the other.

If they cannot be converted implicitly, then the database returns an error.

expr1 and expr2 可以是任意的数据类型，但他们必须是同一数据类型，或者是隐式转换为同一数据类型，又或者是显示转换为同一数据类型。

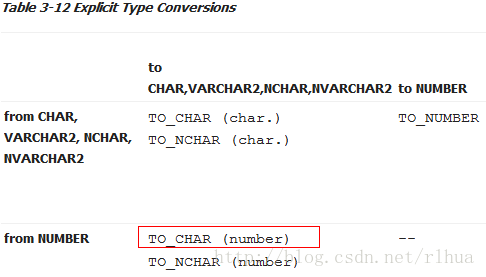
如果他们不是同一类型，则报错。

The implicit conversion is implemented as follows:

* If expr1 is character data, then Oracle Database converts expr2 to the data type ofexpr1 before comparing them and returns VARCHAR2 in the character set ofexpr1.
* 如果expr1 是字符类型，则expr2在比较前转换为expr1的数据类型，在进行比较。
* If expr1 is numeric, then Oracle Database determines which argument has the highest numeric precedence, implicitly converts the other argument to that data type, and returns that data type.
* 如果expr1是数字类型，则判断哪个参数的数据类型高就隐式转为哪个数据类型。

官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/functions119.htm#sthref1312>

**2、Data Conversion**



官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#sthref306>

此处考试的意图在与NVL函数里参数的数据类型的一致性以及数据类型之间的转换。

由题意可知，先用同to\_char函数显示将cust\_credit\_limit\*.15转换为字符类型，才能达成数据类型一致。

A答案：

sh@TEST0910> SELECT NVL(cust\_credit\_limit,'Not Available')\*.15 "NEW CREDIT" FROM customers;

SELECT NVL(cust\_credit\_limit,'Not Available')\*.15 "NEW CREDIT" FROM customers

                             \*

ERROR at line 1:

ORA-01722: invalid number

B答案：

sh@TEST0910> SELECT NVL(cust\_credit\_limit\*.15,'Not Available') "NEW CREDIT" FROM customers;

SELECT NVL(cust\_credit\_limit\*.15,'Not Available') "NEW CREDIT" FROM customers

                                 \*

ERROR at line 1:

ORA-01722: invalid number

C答案：

sh@TEST0910> SELECT TO\_CHAR(NVL(cust\_credit\_limit\*.15,'Not Available')) "NEW CREDIT" FROM customers;

SELECT TO\_CHAR(NVL(cust\_credit\_limit\*.15,'Not Available')) "NEW CREDIT" FROM customers

                                         \*

ERROR at line 1:

ORA-01722: invalid number

D答案：

sh@TEST0910> SELECT NVL(TO\_CHAR(cust\_credit\_limit\*.15),'Not Available') "NEW CREDIT"

  2  FROM customers where rownum<5;

NEW CREDIT

----------------------------------------

225

1050

1650

225

94. Examine the structure of the PROGRAMS table:

name            Null        Type

PROG\_ID         NOT NULL    NUMBER(3)

PROG\_COST                  NUMBER(8,2)

START\_DATE      NOT NULL    DATE

END\_DATE                   DATE

Which two SQL statements would execute successfully? (Choose two.)

A. SELECT NVL(ADD\_MONTHS(END\_DATE,1),SYSDATE)

FROM programs;

B. SELECT TO\_DATE(NVL(SYSDATE-END\_DATE,SYSDATE))

FROM programs;

C. SELECT NVL(MONTHS\_BETWEEN(start\_date,end\_date),'Ongoing')

FROM programs;

D. SELECT NVL(TO\_CHAR(MONTHS\_BETWEEN(start\_date,end\_date)),'Ongoing')

FROM programs;

Answer: AD

本题主要考NVL的参数的数据类型是否相同的问题。

NVL函数见<http://blog.csdn.net/rlhua/article/details/11805803>

1、根据题意创建PROGRAMS表

sh@TEST0910> create table PROGRAMS

  2  (prog\_id number(3) not null,

  3  prog\_cost number(8,2),

  4  start\_date date not null,

  5  end\_date date);

Table created.

2、插入一行数据测试用。

sh@TEST0910> select \* from  PROGRAMS;

   PROG\_ID  PROG\_COST START\_DAT END\_DATE

---------- ---------- --------- ---------

         1       10.1 18-SEP-13 19-SEP-13

3、开始测试：

A答案：

sh@TEST0910> SELECT NVL(ADD\_MONTHS(END\_DATE,1),SYSDATE) FROM programs;

NVL(ADD\_M

---------

19-OCT-13

ADD\_MONTHS returns the date date plus integer months.

ADD\_MONTHS 返回的是数据类型是Date和sysdate类型一致，故正确。

B答案：

sh@TEST0910> SELECT TO\_DATE(NVL(SYSDATE-END\_DATE,SYSDATE))

  2  FROM programs;

SELECT TO\_DATE(NVL(SYSDATE-END\_DATE,SYSDATE))

               \*

ERROR at line 1:

ORA-01858: a non-numeric character was found where a numeric was expected

sh@TEST0910> select SYSDATE-END\_DATE from  programs;

SYSDATE-END\_DATE

----------------

       .9937963

SYSDATE-END\_DATE 返回的是一个数值类型，与sysdate类型不一致。故错误。

C答案：

sh@TEST0910> SELECT NVL(MONTHS\_BETWEEN(start\_date,end\_date),'Ongoing')

  2   FROM programs;

SELECT NVL(MONTHS\_BETWEEN(start\_date,end\_date),'Ongoing')

                                               \*

ERROR at line 1:

ORA-01722: invalid number

sh@TEST0910> select MONTHS\_BETWEEN(start\_date,end\_date) from  programs;

MONTHS\_BETWEEN(START\_DATE,END\_DATE)

-----------------------------------

                         -.03225806

MONTHS\_BETWEEN(start\_date,end\_date)返回是数值类型，与'Ongoing'类型不一致，故错误

使用to\_char使他们类型一致，就是正确的了

sh@TEST0910> SELECT NVL(**to\_char**(MONTHS\_BETWEEN(start\_date,end\_date)),'Ongoing')

  2  FROM programs;

NVL(TO\_CHAR(MONTHS\_BETWEEN(START\_DATE,EN

----------------------------------------

-.03225806451612903225806451612903225806

D答案：NVL内的参数一致，故正确。

sh@TEST0910> SELECT NVL(TO\_CHAR(MONTHS\_BETWEEN(start\_date,end\_date)),'Ongoing')

  2   FROM programs;

NVL(TO\_CHAR(MONTHS\_BETWEEN(START\_DATE,EN

----------------------------------------

-.03225806451612903225806451612903225806

95. The  PRODUCTS table has the following structure:

name             Null            Type

PROD\_ID          NOT NULL        NUMBER(4)

PROD\_NAME                       VARCHAR2(25)

PROD\_EXPIRY\_DATE                DATE

Evaluate the following two SQL statements:

SQL>SELECT prod\_id, NVL2(prod\_expiry\_date, prod\_expiry\_date + 15,'') 如果prod\_expiry\_date为null，则返回空，否则返回prod\_expiry\_date + 15

FROM products;

SQL>SELECT prod\_id, NVL(prod\_expiry\_date, prod\_expiry\_date + 15) 如果prod\_expiry\_date为null，则返回prod\_expiry\_date + 15，否则返回prod\_expiry\_date

FROM products;

Which statement is true regarding the outcome?

A. Both the statements execute and give different results.

B. Both the statements execute and give the same result.

C. Only the first SQL statement executes successfully.

D. Only the second SQL statement executes successfully.

Answer: A

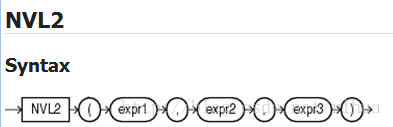
答案解析：

此题考的本意是NVL和NVL2之间逻辑计算的区别。

1、NVL

参考地址：<http://blog.csdn.net/rlhua/article/details/11805803>

2、NVL2



Purpose

NVL2 lets you determine the value returned by a query based on whether a specified expression is null or not null.

If expr1 is not null, then NVL2 returns expr2. If expr1 is null, then NVL2 returns expr3.

如果expr1非空，则返回 expr2，如果expr1是空值，则返回expr3

The argument expr1 can have any data type. The arguments expr2 and expr3 can have any data types except LONG.

expr1 可以是任意数据类型， expr2 and expr3 可以是任意数据类型，但不能是LONG类型，且数据类型要一致，或者隐式转换为一致，或者显示转换为一致。

If the data types of expr2 and expr3 are different, then Oracle Database implicitly converts one to the other.

如果expr2 and expr3 数据类型不同，则隐式转为相同

 If they cannot be converted implicitly, then the database returns an error.

如果不能隐式转换，则报错。

If expr2 is character or numeric data, then the implicit conversion is implemented as follows:

* If expr2 is character data, then Oracle Database converts expr3 to the data type of expr2 before returning a value unless expr3 is a null constant.
* In that case, a data type conversion is not necessary, and the database returns VARCHAR2 in the character set of expr2.
* 如果expr2 是字符类型，则将expr3 转换为expr2相同的数据类型。
* If expr2 is numeric data, then Oracle Database determines which argument has the highest numeric precedence, implicitly converts the other argument to that data type, and returns that data type.

官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/functions120.htm#sthref1315>

搭建环境：

1、创建表，插入数据。

sys@TEST0910> create table products

  2  (prod\_id number(4) not null,

  3  prod\_name varchar2(25),

  4  prod\_expiry\_date date);

Table created.

sys@TEST0910> insert into products values(1,'tomato',sysdate);

1 row created.

sys@TEST0910> select \* from products;

   PROD\_ID PROD\_NAME                 PROD\_EXPI

---------- ------------------------- ---------

         1 tomato                    18-SEP-13

2、开始测试：

sys@TEST0910> SELECT prod\_id, NVL2(prod\_expiry\_date, prod\_expiry\_date + 15,'') from products;

   PROD\_ID NVL2(PROD

---------- ---------

         1 03-OCT-13

prod\_expiry\_date非空，则返回prod\_expiry\_date + 15

sys@TEST0910> SELECT prod\_id, NVL(prod\_expiry\_date, prod\_expiry\_date + 15) from products;

   PROD\_ID NVL(PROD\_

---------- ---------

         1 18-SEP-13

prod\_expiry\_date非空，则返回prod\_expiry\_date

96. Examine the structure of the INVOICE table.

name           Null        Type

INV\_NO         NOT NULL    NUMBER(3)

INV\_DATE                  DATE

INV\_AMT                   NUMBER(10,2)

Which two SQL statements  would execute successfully?  (Choose two.)

A. SELECT inv\_no,NVL2(inv\_date,'Pending','Incomplete')

FROM invoice;

B. SELECT inv\_no,NVL2(inv\_amt,inv\_date,'Not Available')

FROM invoice;

C. SELECT inv\_no,NVL2(inv\_date,sysdate-inv\_date,sysdate)

FROM invoice;

D. SELECT inv\_no,NVL2(inv\_amt,inv\_amt\*.25,'Not Available')

FROM invoice;

Answer: AC

答案解析：

此题考NVL2参数类型是否一致的问题。

NVL2参考：<http://blog.csdn.net/rlhua/article/details/11808291>

搭建环境：

创建表，并插入数据

sys@TEST0910> create table invoice

  2  (inv\_no number(3) not null,

  3  inv\_date date,

  4  inv\_amt number(10,2));

Table created.

sys@TEST0910> insert into invoice values(1,sysdate-2,100.2);

1 row created.

sys@TEST0910> select \* from invoice;

    INV\_NO INV\_DATE     INV\_AMT

---------- --------- ----------

         1 16-SEP-13      100.2

A答案：

sys@TEST0910> SELECT inv\_no,NVL2(inv\_date,'Pending','Incomplete') FROM invoice;

    INV\_NO NVL2(INV\_D

---------- ----------

         1 Pending

inv\_date非空，则返回Pending。且'Pending','Incomplete'数据类型一致，故正确。

B答案：

sys@TEST0910> SELECT inv\_no,NVL2(inv\_amt,inv\_date,'Not Available')  FROM invoice;

SELECT inv\_no,NVL2(inv\_amt,inv\_date,'Not Available')  FROM invoice

                                    \*

ERROR at line 1:

ORA-01858: a non-numeric character was found where a numeric was expected

sys@TEST0910> SELECT inv\_no,NVL2(inv\_amt,to\_char(inv\_date),'Not Available')  FROM invoice;

    INV\_NO NVL2(INV\_AMT,TO\_CH

---------- ------------------

         1 16-SEP-13

inv\_amt非空，则返回inv\_date，但要保证inv\_date和'Not Available'数据类型一致，故用to\_char显示转换。

C答案：

sys@TEST0910> SELECT inv\_no,NVL2(inv\_date,sysdate-inv\_date,sysdate)  FROM invoice;

    INV\_NO NVL2(INV\_DATE,SYSDATE-INV\_DATE,SYSDATE)

---------- ---------------------------------------

         1                              2.00181713

sysdate可以隐式转为数字类型。故正确。

D答案：

sys@TEST0910> SELECT inv\_no,NVL2(inv\_amt,inv\_amt\*.25,'Not Available')

  2  FROM invoice;

SELECT inv\_no,NVL2(inv\_amt,inv\_amt\*.25,'Not Available')

                                       \*

ERROR at line 1:

ORA-01722: invalid number

sys@TEST0910> SELECT inv\_no,NVL2(inv\_amt,to\_char(inv\_amt\*.25),'Not Available') FROM invoice;

    INV\_NO NVL2(INV\_AMT,TO\_CHAR(INV\_AMT\*.25),'NOTAV

---------- ----------------------------------------

         1 25.05

inv\_amt非空，则返回inv\_amt\*.25，但要保证inv\_amt\*.25和'Not Available'数据类型一致，故用to\_char显示转换。

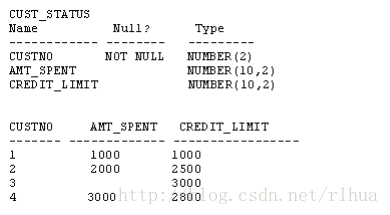
97. View the Exhibit and evaluate the structure and data in the CUST\_STATUS table.

You issue the following SQL statement:

SQL> SELECT custno, NVL2(NULLIF(amt\_spent, credit\_limit), 0, 1000)"BONUS"

FROM cust\_status;

Which statement is true regarding the execution of the above query?



A. It produces an error because the  AMT\_SPENT column contains a  null value.

B. It displays a bonus of 1000 for all customers whose AMT\_SPENT is less than CREDIT\_LIMIT.

C. It displays a bonus of 1000 for all customers whose AMT\_SPENT   equals CREDIT\_LIMIT, or

AMT\_SPENT is null .

D. It produces an error because the TO\_NUMBER function must be used to convert the result of the

NULLIF function before it can be used by the NVL2 function.

Answer: C

答案解析：

实验验证：

scott@TEST0924> create table cust\_status

  2  (custno number(2) not null,

  3  amt\_spent number(10,2),

  4  credit\_limit number(10,2)

  5  )

  6  /

Table created.

scott@TEST0924> insert into cust\_status values(1,1000,1000);

1 row created.

scott@TEST0924> insert into cust\_status values(2,2000,2500);

1 row created.

scott@TEST0924> insert into cust\_status values(3,'',3000);

1 row created.

scott@TEST0924> insert into cust\_status values(4,3000,2800);

1 row created.

scott@TEST0924> select \* from cust\_status;

    CUSTNO  AMT\_SPENT CREDIT\_LIMIT

---------- ---------- ------------

         1       1000         1000

         2       2000         2500

         3                    3000

         4       3000         2800

scott@TEST0924> select custno,nullif(amt\_spent,credit\_limit) from cust\_status;

    CUSTNO NULLIF(AMT\_SPENT,CREDIT\_LIMIT)

---------- ------------------------------

         1

         2                           2000

         3

         4                           3000

从以上sql语句得知，在amt\_spent,credit\_limit相等，或者AMT\_SPENT为空时，返回空值。

运行题中SQL语句

scott@TEST0924> select custno,nvl2(nullif(amt\_spent,credit\_limit),0,1000) "BONUS" from cust\_status;

    CUSTNO      BONUS

---------- ----------

         1       1000

         2          0

         3       1000

         4          0

从以上sql得知，当nullif函数返回空值时，"BONUS" 为1000，也即是说在amt\_spent,credit\_limit相等，或者AMT\_SPENT为空时，"BONUS" 为1000。选项C正好符合此意思。

98. Which statement is true regarding the COALESCE function?

A. It can have a maximum of five expressions in a list. 最多只能有5个表达式，错，无限个

B. It returns the highest NOT NULL  value in the list for all rows. 返回最高的非空值，错，返回第一个非空值

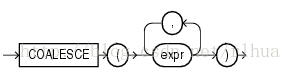
C. It requires that all expressions in the list must be of the same data type. 所有都要相同的类型

D. It requires that at least one of the expressions in the list must have a NOT NULL  value. 至少有一个不是空值，错，可以都是空值，则返回空值

Answer: C

答案解析：

Purpose



COALESCE (expression\_1, expression\_2, ...,expression\_n)

列表中第一个非空的表达式是函数的返回值，如果所有的表达式都是空值，最终将返回一个空值。

COALESCE returns the first non-null expr in the expression list. You must specify at least**two** expressions. If all occurrences of expr evaluate to null, then the function returns null.

Oracle Database uses short-circuit evaluation. The database evaluates each expr value and determines whether it is NULL, rather than evaluating all of the expr values before determining whether any of them is NULL.

If all occurrences of expr are numeric data type or any nonnumeric data type that can be implicitly converted to a numeric data type, then Oracle Database determines the argument with the highest numeric precedence, implicitly converts the remaining arguments to that data type, and returns that data type.

This function is a generalization of the NVL function.

You can also use COALESCE as a variety of the CASE expression. For example,

COALESCE(expr1, expr2)

is equivalent to:

CASE WHEN expr1 IS NOT NULL THEN expr1 ELSE expr2 END

Similarly,

COALESCE(expr1, expr2, ..., exprn)

where n >= 3, is equivalent to:

CASE WHEN expr1 IS NOT NULL THEN expr1

   ELSE COALESCE (expr2, ..., exprn) END

官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/functions030.htm#sthref995>

99. View the Exhibit and examine the structure of the PROMOTIONS table.

Using the PROMOTIONS table,  you need to find out the average cost for all promos in the ranges

$0-2000 and $2000-5000 in category A

You   issue the following SQL statement:

SQL>SELECT AVG(CASE

WHEN promo\_cost BETWEEN 0 AND 2000 AND promo\_category='A'

then promo\_cost

ELSE null END) "CAT\_2000A",

AVG(CASE

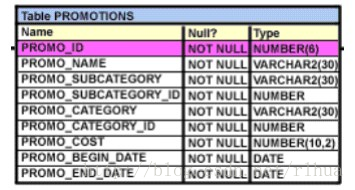
WHEN promo\_cost BETWEEN 2001 AND 5000 AND promo\_category='A'

THEN promo\_cost

ELSE null END) "CAT\_5000A"

FROM promotions;

What would be the outcome?



A. It executes successfully and gives the required result.

B. It generates an error because NULL cannot be specified as a return value.

C. It generates an error because CASE cannot be used with group functions.

D. It generates an error because multiple conditions cannot be specified for the WHEN clause.

Answer: A

答案解析：

在原表中 promo\_category='A'没有，故这里换成'TV'，一样的测试效果。

sh@TEST0910> select promo\_cost from promotions where promo\_category='A';

no rows selected

sh@TEST0910>  select promo\_cost from promotions where promo\_category='TV';

PROMO\_COST

----------

      1100

      1500

      3100

      4800

      5200

...

115 rows selected.

先按条件找出区间范围的值。

sh@TEST0910> SELECT CASE WHEN promo\_cost BETWEEN 0 AND 2000 AND promo\_category='TV' then promo\_cost ELSE null END "CAT\_2000A",

  2  CASE WHEN promo\_cost BETWEEN 2001 AND 5000 AND promo\_category='TV' THEN promo\_cost ELSE null END "CAT\_5000A"

  3  FROM promotions;

 CAT\_2000A  CAT\_5000A

---------- ----------

      1100

      1500

                 3100

                 4800

......

503 rows selected.

再算出平均值。

sh@TEST0910> SELECT AVG(CASE WHEN promo\_cost BETWEEN 0 AND 2000 AND promo\_category='TV' then promo\_cost ELSE null END) "CAT\_2000A",

  2  AVG(CASE WHEN promo\_cost BETWEEN 2001 AND 5000 AND promo\_category='TV' THEN promo\_cost ELSE null END) "CAT\_5000A"

  3  FROM promotions;

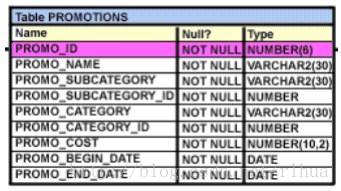
 CAT\_2000A  CAT\_5000A

---------- ----------

      1300       3950

100. View the Exhibit and examine the structure of the PROMOTIONS table.

Which SQL statements are valid?  (Choose all that apply.)



A. SELECT promo\_id, DECODE(NVL(promo\_cost,0), promo\_cost,

promo\_cost \* 0.25, 100) "Discount"

FROM promotions;

B. SELECT promo\_id, DECODE(promo\_cost, 10000,

DECODE(promo\_category, 'G1', promo\_cost \*.25, NULL),

NULL) "Catcost"

FROM promotions;

C. SELECT promo\_id, DECODE(NULLIF(promo\_cost, 10000),

NULL, promo\_cost\*.25, 'N/A') "Catcost"

FROM promotions;

D. SELECT promo\_id, DECODE(promo\_cost, >10000, 'High',

<10000, 'Low') "Range"

FROM promotions;

Answer: AB

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12868497>

<http://blog.csdn.net/rlhua/article/details/12860467>

C不正确，因为把’N/A’隐式转换成数值时不成功。D语法不正确。

DECODE(expr,search1,result1[,search2,result2……,default])：比较expr与search，如果等于search1则返回result1，如果等于search2则返回result2，依次类推，如果都不等于，如果有default则返回default，否则返回NULL.

ORACLE在比较之前，会自动把expr和每一个search隐式转换成第一个search（search1）的数据类型。自动把返回值转换成第一个result（result1）的数据类型。如果第一个result的数据类型为CHAR或者值是null，则Oracle转换返回值为VARCHAR2.

在DECODE函数中,NULL是相等的，如果expr为空，则Oracle将会返回第一个为NULL的search所对应的result。DECODE列表中的最大表达式个数为255个。

101. Examine the data in the PROMO\_BEGIN\_DATE column of the PROMOTIONS table:

PROMO\_BEGIN \_DATE

04-jan-00

10-jan-00

15-dec-99

18-oct-98

22-aug-99

You want to display the number of promotions started in 1999 and 2000.

Which query gives the correct output?

A. SELECT SUM(DECODE(SUBSTR(promo\_begin\_date,8),'00',1,0)) "2000",

SUM(DECODE(SUBSTR(promo\_begin\_date,8),'99',1,0)) "1999"

FROM promotions;

B. SELECT SUM(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '99' THEN 1

ELSE 0 END) "1999",SUM(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '00' THEN 1

ELSE 0 END) "2000"

FROM promotions;

C. SELECT COUNT(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '99' THEN 1

ELSE 0 END) "1999",COUNT(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '00' THEN 1

ELSE 0 END) "2000"

FROM promotions;

D. SELECT COUNT(DECODE(SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8), '1999', 1, 0)) "1999",

COUNT(DECODE(SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8),'2000', 1,

0)) "2000"

FROM promotions;

Answer: A

答案解析：

分析A答案

SUBSTR(promo\_begin\_date,8)，即是取日期的最后两个值。

(DECODE(SUBSTR(promo\_begin\_date,8),'00',1,0)如果最后两个值是00，则值为1，否则为0

(DECODE(SUBSTR(promo\_begin\_date,8),'99',1,0)如果最后两个值是99，则值为1，否则为0

 SUM(DECODE(SUBSTR(promo\_begin\_date,8),'00',1,0)) 加起来有多少个1，即有多少个2000年的

SUM(DECODE(SUBSTR(promo\_begin\_date,8),'99',1,0))加起来有多少个1，即有多少个1999年的

B，条件不对，应该 TO\_CHAR(promo\_begin\_date,'yyyy') 改成 TO\_CHAR(promo\_begin\_date,'yy')

C,与B同样错误

D,截取不对，SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8)改成SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 1)

102. Examine the structure of the TRANSACTIONS table:

name            Null            Type

TRANS\_ID        NOT NULL        NUMBER(3)

CUST\_NAME                      VARCHAR2(30)

TRANS\_DATE                     TIMESTAMP

TRANS\_AMT  NUMBER(10,2)

You want to display the date, time, and transaction amount of transactions that where done before 12

noon. The value zero should be displayed for transactions where the transaction amount has not been

entered.

Which query gives the required result?

A. SELECT TO\_CHAR(trans\_date,'dd-mon-yyyy hh24:mi:ss'),

TO\_CHAR(trans\_amt,'$99999999D99')

FROM transactions

WHERE TO\_NUMBER(TO\_DATE(trans\_date,'hh24')) < 12 AND COALESCE(trans\_amt,NULL)<>NULL;

B. SELECT TO\_CHAR(trans\_date,'dd-mon-yyyy hh24:mi:ss'),

NVL(TO\_CHAR(trans\_amt,'$99999999D99'),0)

FROM transactions

WHERE TO\_CHAR(trans\_date,'hh24') < 12;

C. SELECT TO\_CHAR(trans\_date,'dd-mon-yyyy hh24:mi:ss'),

COALESCE(TO\_NUMBER(trans\_amt,'$99999999.99'),0)  已经是number类型，不需要转换

FROM transactions

WHERE TO\_DATE(trans\_date,'hh24') < 12;

D. SELECT TO\_DATE (trans\_date,'dd-mon-yyyy hh24:mi:ss'),

NVL2(trans\_amt,TO\_NUMBER(trans\_amt,'$99999999.99'), 0)  已经是number类型，不需要转换

FROM transactions

WHERE TO\_DATE(trans\_date,'hh24') < 12;

Answer: B

答案解析：

A,TO\_DATE(trans\_date,'hh24')，trans\_date已经为日期类型，不用再to\_date

scott@TESTDB> select to\_date(HIREDATE,'yy24') from emp;

select to\_date(HIREDATE,'yy24') from emp

                        \*

ERROR at line 1:

ORA-01821: date format not recognized

B,正确，这里0隐式转换为字符型

C,(TO\_NUMBER(trans\_amt,'$99999999.99')，trans\_amt已经是数值型，不用再to\_number

D,错误同C一样。

103. Examine the structure of the TRANSACTIONS table:

name          Null          Type

TRANS\_ID      NOT NULL      NUMBER(3)

CUST\_NAME                  VARCHAR2(30)

TRANS\_DATE                 DATE

TRANS\_AMT                  NUMBER(10,2)

You want to display the transaction date and specify whether it is a weekday or weekend.

Evaluate the following two queries:

SQL>SELECT TRANS\_DATE,CASE

WHEN TRIM(TO\_CHAR(trans\_date,'DAY')) IN ('SATURDAY','SUNDAY') THEN 'weekend'

ELSE 'weekday'

END "Day Type"

FROM transactions;

SQL>SELECT TRANS\_DATE, CASE

WHEN TO\_CHAR(trans\_date,'DAY') BETWEEN 'MONDAY' AND 'FRIDAY' THEN 'weekday'

ELSE   'weekend'

END "Day Type"FROM transactions;

Which statement is true regarding the above queries?

A. Both give wrong results.

B. Both give the correct  result.

C. Only the first query gives the correct result.

D. Only the  second query gives the correct result.

Answer: C

 答案解析：

此处 BETWEEN 'MONDAY' AND 'FRIDAY'是指MONDA\*\*\*\*后面的值到FRIDAY

这里是按照字符的ASCII码来排序的，此处的排序永远为假，即输出的都是weekend。

SQL>SELECT HIREDATE,CASE

WHEN TRIM(TO\_CHAR(HIREDATE,'DAY')) IN ('SATURDAY','SUNDAY') THEN 'weekend' ELSE 'weekday' END "Day Type"

FROM emp;

实验验证，此处使用scott的emp表下的HIREDATE字段来做测试

*scott@TEST0924> SELECT HIREDATE,CASE*

*2  WHEN TRIM(TO\_CHAR(HIREDATE,'DAY')) IN ('SATURDAY','SUNDAY') THEN 'weekend' ELSE 'weekday' END "Day Type"*

*3  FROM emp;*

*HIREDATE           Day Typ*

*------------------ -------*

*17-DEC-80          weekday*

*20-FEB-81          weekday*

*22-FEB-81          weekend*

*02-APR-81          weekday*

*28-SEP-81          weekday*

*01-MAY-81          weekday*

*09-JUN-81          weekday*

*19-APR-87          weekend*

*17-NOV-81          weekday*

*08-SEP-81          weekday*

*23-MAY-87          weekend*

*03-DEC-81          weekday*

*03-DEC-81          weekday*

*23-JAN-82          weekend*

*14 rows selected.*

*scott@TEST0924> SELECT HIREDATE, CASE*

*2   WHEN TO\_CHAR(HIREDATE,'DAY') BETWEEN 'MONDAY' AND 'FRIDAY' THEN 'weekday'  ELSE   'weekend'*

*3  END "Day Type"FROM emp;*

*HIREDATE           Day Typ*

*------------------ -------*

*17-DEC-80          weekend*

*20-FEB-81          weekend*

*22-FEB-81          weekend*

*02-APR-81          weekend*

*28-SEP-81          weekend*

*01-MAY-81          weekend*

*09-JUN-81          weekend*

*19-APR-87          weekend*

*17-NOV-81          weekend*

*08-SEP-81          weekend*

*23-MAY-87          weekend*

*03-DEC-81          weekend*

*03-DEC-81          weekend*

*23-JAN-82          weekend*

*14 rows selected.*

104. Examine the structure of the PROMOS table:

name              Null        Type

PROMO\_ID          NOT NULL    NUMBER(3)

PROMO\_NAME                   VARCHAR2(30)

PROMO\_START\_DATE NOT NULL    DATE

PROMO\_END\_DATE               DATE

You want to generate a report showing promo names and their duration (number of days). If the

PROMO\_END\_DATE has not been entered, the message 'ONGOING' should be displayed.

Which queries give the correct output? (Choose all that apply.)

A. SELECT promo\_name, TO\_CHAR(NVL(promo\_end\_date -promo\_start\_date,'ONGOING'))

FROM promos;

B. SELECT promo\_name,COALESCE(TO\_CHAR(promo\_end\_date - promo\_start\_date),'ONGOING')

FROM promos;

C. SELECT promo\_name, NVL(TO\_CHAR(promo\_end\_date -promo\_start\_date),'ONGOING')

FROM promos;

D. SELECT promo\_name, DECODE(promo\_end\_date -promo\_start\_date,NULL,'ONGOING',promo\_end\_date - promo\_start\_date)

FROM promos;

E. SELECT promo\_name, decode(coalesce(promo\_end\_date,promo\_start\_date),null,'ONGOING',

promo\_end\_date - promo\_start\_date)

FROM promos;

Answer: BCD

 答案解析：

ANVL要求括号内参数的类型需一致，,NVL(promo\_end\_date -promo\_start\_date,'ONGOING')此处类型不一致，一个为数值型，一个为字符型

B, COALESCE 需括号内参数类型需一致，TO\_CHAR(promo\_end\_date - promo\_start\_date),'ONGOING') ，此处使用to\_char转换，两个参数都为字符型，可以比较。正确。

C, NVL(TO\_CHAR(promo\_end\_date -promo\_start\_date),'ONGOING')，NVL括号内两个参数都为字符型，正确。

D,表达式正确

E,不满足题意，coalesce返回第一个不为空的值，因为promo\_start\_date肯定不为空，所以即使promo\_end\_date为空，coalesce也会返回promo\_start\_dat，所以都会显示promo\_end\_date - promo\_start\_date值，即当promo\_end\_date为空时，还是会显示空（注意，如果你直接指定NULL-sysdate会报转换数据类型错，而在字段里就可以减，并返回空。

105. Examine the structure of the PROMOS table:

name              Null        Type

PROMO\_ID          NOT NULL    NUMBER(3)

PROMO\_NAME                   VARCHAR2(30)

PROMO\_START\_DATE NOT NULL    DATE

PROMO\_END\_DATE    NOT NULL    DATE

You want to display the list of promo names with the message 'Same Day' for promos that started and

ended on the same day.

Which query gives the correct output?

A. SELECT promo\_name, NVL(NULLIF(promo\_start\_date, promo\_end\_date), 'Same Day') FROM promos;

B. SELECT promo\_name, NVL(TRUNC(promo\_end\_date - promo\_start\_date), 'Same Day') FROM promos;

C. SELECT promo\_name, NVL2(TO\_CHAR(TRUNC(promo\_end\_date-promo\_start\_date)), NULL,'SameDay') FROM promos;

D. SELECT promo\_name, DECODE((NULLIF(promo\_start\_date, promo\_end\_date)), NULL,'Same day') FROM promos;

Answer: D

答案解析：

A。NULLIF里面额数据类型为date与 'Same Day'字符串类型不匹配，错误。

B。TRUNC将小数部分截断，但类型不一致，错误。

C。TO\_CHAR(TRUNC(promo\_end\_date-promo\_end\_date))，如果相等，TRUNC为0，to\_char后，字符为0，即始终有值，NVL2返回第二个值，即NULL

sh@TEST0924>  SELECT promo\_name, NVL2(TO\_CHAR(TRUNC(promo\_end\_date-promo\_end\_date)), NULL,'SameDay') FROM PROMOTIONS;

PROMO\_NAME                     NVL2(TO

------------------------------ -------

NO PROMOTION #

newspaper promotion #16-108

post promotion #20-232

newspaper promotion #16-349

internet promotion #14-471

D。实验验证：正确的。

scott@TESTDB> SELECT  DECODE((NULLIF('11-mon-01','11-mon-01')),null,'same') from dual;

DECO

----

same

scott@TESTDB> SELECT  DECODE(null,null,'same') from dual;

DECO

----

same

106. Examine the data in the LIST\_PRICE and MIN\_PRICE columns of the PRODUCTS table:

LIST\_PRICE     MIN\_PRICE

10000          8000

20000

30000          30000

Which two expressions give the same output? (Choose two.)

A. NVL(NULLIF(list\_price, min\_price), 0)

B. NVL(COALESCE(list\_price, min\_price), 0)  如果list\_price为空，则返回min\_price，如果min\_price为空，则返回空值，如果是空值，则返回0

C. NVL2(COALESCE(list\_price, min\_price), min\_price, 0)

D. COALESCE(NVL2(list\_price, list\_price, min\_price), 0) 如果list\_price为空，则返回min\_price，如果min\_price为空，则返回0，否则返回list\_price，

Answer: BD

答案解析：

LIST\_PRICE 从表中看，没有空值，故

B的结果为list\_price，D的结果为list\_price

C的结果为min\_price

A分情况，不等时为list\_price，相等时为0

1.NVL参考：<http://blog.csdn.net/rlhua/article/details/11805803>

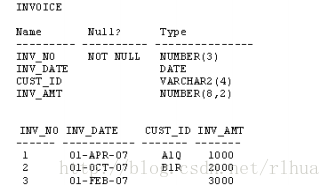
2.NVL2参考：<http://blog.csdn.net/rlhua/article/details/11808291>

3.NULLIF参考：<http://blog.csdn.net/rlhua/article/details/11810737>

4.COALESCE参考：<http://blog.csdn.net/rlhua/article/details/11919341>

107. View the Exhibit and examine the structure and data in the INVOICE table.

Which two SQL statements would execute successfully?  (Choose two.)



A. SELECT AVG(inv\_date )  类型不一致，不能对date求平均值

FROM invoice;

B. SELECT MAX(inv\_date),MIN(cust\_id)

FROM invoice;

C. SELECT MAX(AVG(SYSDATE - inv\_date))  缺少group by

FROM invoice;

D. SELECT AVG( inv\_date - SYSDATE), AVG(inv\_amt)

FROM invoice;

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12868639>

A，不能对date类型求平均数

B,MAX,MIN可以对数值和date求最大值和最小值

C,组合函数，需要分组，错误

D,AVG对数值求平均数，正确。

首先创建测试表：

sh@TEST0924> create table invoice

  2  (inv\_no number(3) not null,

  3  inv\_date date,

  4  cust\_id varchar2(4),

  5  inv\_amt number(8,2)

  6  );

Table created.

插入数据后查询：

sh@TEST0924> select \* from invoice;

    INV\_NO INV\_DATE  CUST     INV\_AMT

----------     ---------        ----         ----------

         1 01-APR-07     A10        1000

         2 01-OCT-07     B1R        2000

         3 01-FEB-07                     3000

开始测试：

A答案：AGV函数需要一个数值类型或者可以隐式转为数值类型的参数

sh@TEST0924> SELECT AVG(inv\_date ) FROM invoice;

SELECT AVG(inv\_date ) FROM invoice

           \*

ERROR at line 1:

ORA-00932: inconsistent datatypes: expected NUMBER got DATE

AGV  function takes as an argument any numeric data type or any nonnumeric data type that can be implicitly converted to a numeric data type. The function returns the same data type as the numeric data type of the argument.

B答案：

sh@TEST0924>  SELECT MAX(inv\_date),MIN(cust\_id)

  2  FROM invoice;

MAX(INV\_D MIN(

--------- ----

01-OCT-07 A10

C答案：缺少group 不要函数

sh@TEST0924> SELECT MAX(AVG(SYSDATE - inv\_date)) FROM invoice;

SELECT MAX(AVG(SYSDATE - inv\_date)) FROM invoice

           \*

ERROR at line 1:

ORA-00978: nested group function without GROUP BY

sh@TEST0924> SELECT MAX(AVG(SYSDATE - inv\_date)) FROM invoice

  2  group by cust\_id;

MAX(AVG(SYSDATE-INV\_DATE))

--------------------------

                2427.62578

可以单独求平均值和最大值。

sh@TEST0924> SELECT AVG(SYSDATE - inv\_date)FROM invoice;

AVG(SYSDATE-INV\_DATE)

---------------------

           2327.29273

sh@TEST0924>  SELECT MAX(SYSDATE - inv\_date) FROM invoice;

MAX(SYSDATE-INV\_DATE)

---------------------

           2427.62655

D答案：

sh@TEST0924> SELECT AVG( inv\_date - SYSDATE), AVG(inv\_amt) FROM invoice;

AVG(INV\_DATE-SYSDATE)     AVG(INV\_AMT)

---------------------                         -----------

           -2327.2942                        2000

108. Which two statements are true regarding the COUNT function? (Choose two.)

A. The  COUNT function can be used only for CHAR, VARCHAR2,  and NUMBER data types.

B. COUNT(\*) returns the number of rows including duplicate rows and rows containing NULL  value in

any of the columns.

C. COUNT(cust\_id) returns the number of rows including rows with duplicate customer IDs and NULL

value in the CUST\_ID column.

D. COUNT(DISTINCT inv\_amt)returns the number of rows excluding排除在外 rows containing duplicates and

NULL values in the INV\_AMT column.

E. A SELECT statement using the  COUNT function with a  DISTINCT keyword   cannot have a

WHERE clause.

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12868639>

COUNT函数有以下三种格式：

• COUNT(\*)

• COUNT(expr)

• COUNT(DISTINCT expr)

COUNT(\*)用于返回表中符合SELECT语句标准的行数，包括重复行和在任何列中含有空值的行。

如果SELECT语句中包含WHERE子句，则COUNT(\*)会返回符合WHERE子句中条件的行数。

相反，COUNT(expr)返回由expr标识的列中非空值的数量。

COUNT(DISTINCT expr)返回由expr标识的列中不同非空值的数量。

测试实验如下表：有两组一样的row，

scott@TEST0924> select \* from test;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO

----------     ----------      ---------        ----------      --------           - ---------- ---------- ----------

      7369 SMITH      CLERK           7902 17-DEC-80               800                        20

      7499 ALLEN      SALESMAN        7698 20-FEB-81          1600        300         30

      7521 WARD       SALESMAN        7698 22-FEB-81         1250        500         30

      7566 JONES      MANAGER         7839 02-APR-81         2975                         20

      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30

      7566 JONES      MANAGER         7839 02-APR-81         2975                         20

      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30

7 rows selected.

count(\*)时，返回所有的行数，包括重复的，以及有空值的

scott@TEST0924> select count(\*) from test;

  COUNT(\*)

----------

         7

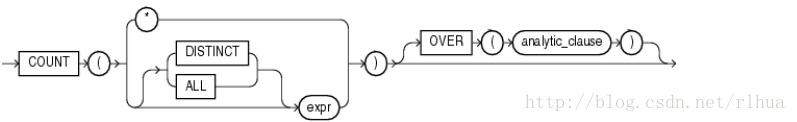
count(distinct comm) ，首先去掉重复的和空值的，然后总计行[数](mailto:scott@TEST0924)

scott@TEST0924> select count(distinct comm) from test;

COUNT(DISTINCTCOMM)

-------------------

                  3



COUNT returns the number of rows returned by the query. You can use it as an aggregate or analytic function.你可以用它作为一个聚合函数或解析函数

If you specify DISTINCT, then you can specify only the query\_partition\_clause of theanalytic\_clause. Theorder\_by\_clause and windowing\_clause are not allowed.

If you specify expr, then COUNT returns the number of rows whereexpr is not null. You can count either all rows, or only distinct values ofexpr.

If you specify the asterisk (\*), then this function returns all rows, including duplicates and nulls. COUNT never returns null.

109. Examine the structure of the MARKS table:

name            Null        Type

STUDENT\_ID      NOT NULL    VARCHAR2(4)

STUDENT\_NAME               VARCHAR2(25)

SUBJECT1                   NUMBER(3)

SUBJECT2                    NUMBER(3)

SUBJECT3                    NUMBER(3)

Which two statements would execute successfully?  (Choose two.)

A. SELECT student\_name,subject1

FROM marks

WHERE subject1 > AVG(subject1); where不能跟表达式

B. SELECT student\_name,SUM(subject1)

FROM marks

WHERE student\_name LIKE 'R%';  少了group by

C. SELECT SUM(subject1+subject2+subject3)

FROM marks

WHERE student\_name IS NULL;

D. SELECT SUM(DISTINCT NVL(subject1,0)), MAX(subject1)

FROM marks

WHERE subject1 > subject2;

Answer: CD

答案解析：

实验测试：

首先创建marks表

scott@TEST0924> create table marks

  2  (student\_id varchar2(4) not null,

  3  student\_name varchar(25),

  4  subject1 number(3),

  5  subject2 number(3),

  6  subject3 number(3)

  7  );

插入几条数据后查询如下：

scott@TEST0924> select \* from marks;

STUD STUDENT\_NAME                SUBJECT1   SUBJECT2   SUBJECT3

---- -------------------------                     ---------- ---------- ----------

1                        aa                                88         89         90

2                        bb                                85         85         80

3                        cc                                87         81         95

A答案：AGV分组函数不能用在where子句里

scott@TEST0924> SELECT student\_name,subject1

  2  FROM marks

  3  WHERE subject1 > AVG(subject1);

WHERE subject1 > AVG(subject1)

                 \*

ERROR at line 3:

ORA-00934: group function is not allowed here

scott@TEST0924> SELECT student\_name,subject1

  2  FROM marks

  3  WHERE subject1>(select AVG(subject1) from marks);

STUDENT\_NAME                SUBJECT1

------------------------- ----------

aa                                88

cc                                87

B答案：缺少group by

scott@TEST0924> SELECT student\_name,SUM(subject1)

  2  FROM marks

  3  WHERE student\_name LIKE 'a%';

SELECT student\_name,SUM(subject1)

       \*

ERROR at line 1:

ORA-00937: not a single-group group function

scott@TEST0924> SELECT student\_name,SUM(subject1)

  2  FROM marks

  3  WHERE student\_name LIKE 'a%'

  4  group by student\_name;

STUDENT\_NAME              SUM(SUBJECT1)

------------------------- -------------

aa                                   88

C答案：因为测试数据里没有student\_name没有为空的，所有无返回值。

cott@TEST0924> SELECT SUM(subject1+subject2+subject3)

  2  FROM marks

  3  WHERE student\_name IS NULL;

SUM(SUBJECT1+SUBJECT2+SUBJECT3)

-------------------------------

故用is not null 一样的测试效果。

scott@TEST0924>

scott@TEST0924> SELECT SUM(subject1+subject2+subject3)

  2  FROM marks

  3  WHERE student\_name IS not NULL;

SUM(SUBJECT1+SUBJECT2+SUBJECT3)

-------------------------------

                            780

D答案：

cott@TEST0924> SELECT SUM(DISTINCT NVL(subject1,0)), MAX(subject1)

  2  FROM marks

  3  WHERE subject1 > subject2;

SUM(DISTINCTNVL(SUBJECT1,0)) MAX(SUBJECT1)

---------------------------- -------------

                          87            87

110. View the Exhibit and examine the structure of the CUSTOMERS table.

Using the CUSTOMERS table, you need to generate a report that shows   the average credit limit for

customers in WASHINGTON and NEW YORK.

Which SQL statement would produce   the required result?

A. SELECT cust\_city, AVG(cust\_credit\_limit)

FROM customers

WHERE cust\_city IN ('WASHINGTON','NEW YORK')

GROUP BY cust\_credit\_limit, cust\_city;

B. SELECT cust\_city, AVG(cust\_credit\_limit)

FROM customers

WHERE cust\_city IN ('WASHINGTON','NEW YORK')

GROUP BY cust\_city,cust\_credit\_limit;

C. SELECT cust\_city, AVG(cust\_credit\_limit)

FROM customers

WHERE cust\_city IN ('WASHINGTON','NEW YORK')

GROUP BY cust\_city; 按照城市来取平均值

D. SELECT cust\_city, AVG(NVL(cust\_credit\_limit,0))

FROM customers

WHERE cust\_city IN ('WASHINGTON','NEW YORK');

Answer: C

答案解析：

A答案：此处是按照cust\_credit\_limit, cust\_city;这两个一起分组的。此处换另外两个城市来测试。

sh@TEST0924> SELECT cust\_city,AVG(cust\_credit\_limit) FROM customers WHERE cust\_city IN ('Duncan','Norman')

  2  GROUP BY cust\_credit\_limit, cust\_city;

CUST\_CITY                      AVG(CUST\_CREDIT\_LIMIT)

------------------------------ ----------------------

Norman                                           1500

Duncan                                           9000

Duncan                                          11000

Norman                                           3000

Norman                                           5000

Duncan                                           3000

Norman                                          11000

Duncan                                           5000

Norman                                          10000

Duncan                                           1500

Duncan                                           7000

Norman                                           9000

Duncan                                          10000

Duncan                                          15000

Norman                                           7000

Norman                                          15000

16 rows selected.

B答案，与A答案一样，按照cust\_credit\_limit, cust\_city;这两个一起分组的。

C答案，按照cust\_city来分组，与题意复合

h@TEST0924> SELECT cust\_city,AVG(cust\_credit\_limit) FROM customers WHERE cust\_city IN ('Duncan','Norman')

  2  GROUP BY cust\_city;

CUST\_CITY                      AVG(CUST\_CREDIT\_LIMIT)

------------------------------ ----------------------

Duncan                                           6550

Norman                                     6634.92063

D答案：缺少group by

sh@TEST0924> SELECT cust\_city, AVG(NVL(cust\_credit\_limit,0)) FROM customers WHERE cust\_city IN ('Duncan','Norman');

SELECT cust\_city, AVG(NVL(cust\_credit\_limit,0)) FROM customers WHERE cust\_city IN ('Duncan','Norman')

       \*

ERROR at line 1:

ORA-00937: not a single-group group function

sh@TEST0924> SELECT cust\_city, AVG(NVL(cust\_credit\_limit,0)) FROM customers WHERE cust\_city IN ('Duncan','Norman')

  2  GROUP BY cust\_city;

CUST\_CITY                      AVG(NVL(CUST\_CREDIT\_LIMIT,0))

------------------------------ -----------------------------

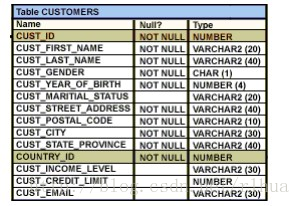
Duncan                                                  6550

Norman                                            6634.92063

111. View the Exhibit and examine the structure of the CUSTOMERS table.

Which statement would display the highest credit limit available in each income level in each city in   the

CUSTOMERS table?



A. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit )

FROM customers

GROUP BY cust\_city, cust\_income\_level, cust\_credit\_limit;

B. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)

FROM customers

GROUP BY cust\_city, cust\_income\_level; 分组先按照city再按照income level

C. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)

FROM customers

GROUP BY cust\_credit\_limit, cust\_income\_level, cust\_city ;  顺序错了

D. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)

FROM customers

GROUP BY cust\_city, cust\_income\_level, MAX(cust\_credit\_limit);

Answer: B

答案解析：

按照题意，先按照每个城市后按照每个城市中的收入水平来分组查看最高的credit limit

故在group by时应为 group by cust\_city, cust\_income\_level;

A答案，没有涉及到cust\_credit\_limit的分组。

B答案与题意相符，正确，数据太多，取前5行。

sh@TEST0924> SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)

  2   FROM customers where rownum<=5

  3  GROUP BY cust\_city, cust\_income\_level;

CUST\_CITY                      CUST\_INCOME\_LEVEL              MAX(CUST\_CREDIT\_LIMIT)

------------------------------ ------------------------------ ----------------------

Joinville                      G: 130,000 - 149,999                             9000

Ede                            G: 130,000 - 149,999                             1500

Schimmert                      G: 130,000 - 149,999                            11000

Scheveningen                   G: 130,000 - 149,999                             1500

Hoofddorp                      G: 130,000 - 149,999                             7000

C答案，没有涉及cust\_credit\_limit分组，且cust\_income\_level, cust\_city 顺序错了，应为group by cust\_city, cust\_income\_level;

D答案，语法错误

sh@TEST0924> SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)

  2  FROM customers

  3  GROUP BY cust\_city, cust\_income\_level, MAX(cust\_credit\_limit);

GROUP BY cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)

                                       \*

ERROR at line 3:

ORA-00934: group function is not allowed here

112. View the Exhibit and examine the structure of the PROMOTIONS table.

Evaluate the following SQL statement:

SQL>SELECT promo\_category, AVG(promo\_cost) Avg\_Cost, AVG(promo\_cost)\*.25 Avg\_Overhead

FROM promotions

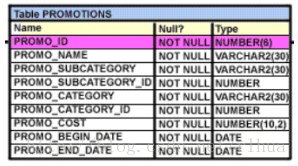
WHERE UPPER(promo\_category) IN ('TV', 'INTERNET','POST')

GROUP BY Avg\_Cost

ORDER BY Avg\_Overhead;

The above query generates an error on execution.

Which clause in the above SQL statement causes the error?



A. WHERE

B. SELECT

C. GROUP BY

D. ORDER BY

Answer: C

答案解析：

group by不能跟列别名

sh@TEST0924> SELECT promo\_category, AVG(promo\_cost) Avg\_Cost, AVG(promo\_cost)\*.25 Avg\_Overhead

  2  FROM promotions

  3  WHERE UPPER(promo\_category) IN ('TV', 'INTERNET','POST')

  4  GROUP BY Avg\_Cost

  5  ORDER BY Avg\_Overhead;

GROUP BY Avg\_Cost

         \*

ERROR at line 4:

ORA-00904: "AVG\_COST": invalid identifier

113. Examine the structure of the ORDERS table:

Name             Null      Type

ORDER\_ID        NOT NULL    NUMBER(12)

ORDER\_DATE      NOT NULL    TIMESTAMP(6)

CUSTOMER\_ID     NOT NULL    NUMBER(6)

ORDER\_STATUS               NUMBER(2)

ORDER\_TOTAL                NUMBER(8,2)

You want to find the total value of all the orders for each year and issue the following command:

SQL>SELECT TO\_CHAR(order\_date,'rr'), SUM(order\_total)

FROM orders

GROUP BY TO\_CHAR(order\_date,'yyyy');

Which statement is true regarding the outcome?

A. It executes successfully and gives the correct output.

B. It gives an error because the TO\_CHAR function is not valid.

C. It executes successfully but does not give the correct output.

D. It gives an error because the data type conversion in the SELECT list does not match the data type

conversion in the GROUP BY clause. 格式不一致

Answer: D

答案解析：

报错，因为SELECT列表中的数据类型转换与GROUP BY子句中的数据类型转换不匹配。

rr和yyyy的格式不一致，改为一致后可以得出数据。

scott@TEST0924> select to\_char(HIREDATE,'rr'),sum(sal)

  2  from emp

  3  group by to\_char(HIREDATE,'yyyy');

select to\_char(HIREDATE,'rr'),sum(sal)

               \*

ERROR at line 1:

ORA-00979: not a GROUP BY expression

scott@TEST0924>  select to\_char(HIREDATE,'yyyy'),sum(sal)

  2  from emp

  3  group by to\_char(HIREDATE,'yyyy');

TO\_C   SUM(SAL)

---- ----------

1987       4100

1980        800

1982       1300

1981      22825

scott@TEST0924> select to\_char(HIREDATE,'rr'),sum(sal)

  2  from emp

  3  group by to\_char(HIREDATE,'rr');

TO   SUM(SAL)

-- ----------

87       4100

81      22825

82       1300

114. View the Exhibit and examine the structure of the SALES table.

The following query is written to retrieve all those product ID s from the SALES table that have more than

55000 sold  and have been ordered more than 10 times.

SQL> SELECT prod\_id

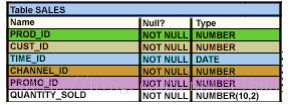
FROM sales

WHERE quantity\_sold > 55000 AND COUNT(\*)>10

GROUP BY prod\_id

HAVING COUNT(\*)>10;

Which statement is true regarding this SQL statement?



A. It executes successfully and generates the required result.

B. It produces an error because COUNT(\*) should be specified in the SELECT clause also.

C. It produces an error because COUNT(\*) should be only in the HAVING clause and not in the WHERE

clause.

D. It executes successfully but produces no result because COUNT(prod\_id) should be used instead of

COUNT(\*).

Answer: C

答案解析：

sh@TEST0924> SELECT prod\_id  FROM sales  WHERE quantity\_sold > 55000

  2  AND COUNT(\*)>10

  3  GROUP BY prod\_id

  4  HAVING COUNT(\*)>10;

AND COUNT(\*)>10

    \*

ERROR at line 2:

ORA-00934: group function is not allowed here

 因为COUNT(\*)只能用在HAVING子句中，不能在WHERE子句中

 where子句不能带组函数

115. View the Exhibit and examine the structure of the CUSTOMERS table.

Evaluate the following SQL statement:

SQL> SELECT cust\_city, COUNT(cust\_last\_name)

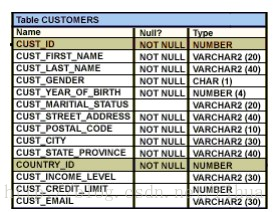
FROM customers

WHERE cust\_credit\_limit > 1000

GROUP BY cust\_city

HAVING AVG(cust\_credit\_limit) BETWEEN 5000 AND 6000;

Which statement is true regarding the outcome of the above query?



A. It executes successfully.

B. It returns an error because the BETWEEN operator cannot be used in the HAVING clause.

C. It returns an error because WHERE and HAVING clauses cannot be used in the same SELECT

statement.

D. It returns an error because WHERE and HAVING clauses cannot be used to apply conditions on the

same column.

Answer: A

答案解析：

sh@TEST0924> SELECT cust\_city, COUNT(cust\_last\_name) FROM customers

  2  WHERE cust\_credit\_limit > 1000

  3  GROUP BY cust\_city

  4  HAVING AVG(cust\_credit\_limit) BETWEEN 5000 AND 6000;

CUST\_CITY                      COUNT(CUST\_LAST\_NAME)

------------------------------ ---------------------

Joinville                                         98

Neuss                                            241

Clermont-l'Herault                      202

Ingolstadt                                     159

...

Sliedrecht                                                      17

207 rows selected.

116. Examine the data in the ORD\_ITEMS table:

ORD\_NO     ITEM\_NO    QTY

1          111        10

1          222        20

1          333        30

2          333        30

2          444        40

3          111        40

You want to find out if there is any item in the table for which the average maximum quantity is more than

50.

You issue the following query:

SQL> SELECT AVG(MAX(qty))

FROM ord\_items

GROUP BY item\_no

HAVING AVG(MAX(qty))>50;

Which statement is true regarding the outcome of this query?

A. It executes successfully and gives the correct output.

B. It gives an error because the HAVING clause is not valid.

C. It executes successfully but does not give the correct output.

D. It gives an error because the GROUP BY expression is not valid.

Answer: B

答案解析：

sh@TESTDB> select \* from ord\_items;

    ORD\_NO    ITEM\_NO        QTY

---------- ---------- ----------

         1        111         10

         1        222         20

         1        333         30

         2        333         30

         2        444         40

         3        111         40

6 rows selected.

sh@TESTDB> select max(qty) from ord\_items group by item\_no;

  MAX(QTY)

----------

        40

        20

        30

        40

sh@TESTDB> select avg(max(qty)) from ord\_items group by item\_no;

AVG(MAX(QTY))

-------------

         32.5

sh@TESTDB> select avg(max(qty)) from ord\_items group by item\_no having avg(max(qty))>50;

select avg(max(qty)) from ord\_items group by item\_no having avg(max(qty))>50

                                                                \*

ERROR at line 1:

ORA-00935: group function is nested too deeply

 HAVING子句后面不能有好几个组函数嵌套，只能使用一个组函数。

117. Which statements are true regarding the WHERE and HAVING clauses in a SELECT statement?

(Choose all that apply.)

A. The HAVING clause can be used with aggregate functions in subqueries.

B. The WHERE clause can be used to exclude rows after dividing them into groups.

C. The WHERE clause can be used to exclude rows before dividing them into groups.

D. The aggregate functions and columns used in the HAVING clause must be specified in the SELECT list

of the query.

E. The WHERE and HAVING clauses can be used in the same statement only if they are applied to

different columns in the table.

Answer: AC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12871951>

A,having可以用在聚合函数的子句里。正确

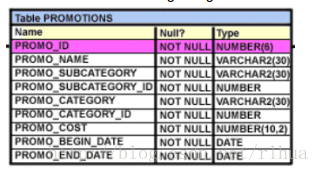
B,分组之后不能使用where来限制，因为where只能限制行，不能限制组，错误

C,分组之前可以使用where来限制所要输出的行，正确

D,在having子句的聚合函数或者字段不一定要在select列表中，错误。

E,在同一句子中，where和having可以都使用，可以应用在不同的字段，也可以相同的字段。

118. View the Exhibit and examine the structure of the PROMOTIONS table.   
Examine the following two SQL statements:   
Statement 1   
SQL>SELECT promo\_category,SUM(promo\_cost)       
FROM promotions       
WHERE promo\_end\_date-promo\_begin\_date > 30       
GROUP BY promo\_category;   
Statement 2   
SQL>SELECT promo\_category,sum(promo\_cost)       
FROM promotions       
GROUP BY promo\_category       
HAVING MIN(promo\_end\_date-promo\_begin\_date)>30;   
Which statement is true regarding the above two SQL statements?



A. statement 1 gives an error, statement 2 executes successfully   
B. statement 2 gives an error, statement 1 executes successfully   
C. statement 1 and statement 2 execute successfully and give the same output   
D. statement 1 and statement 2 execute successfully and give a different output

答案解析：

sh@TESTDB> select promo\_category,SUM(promo\_cost)

  2  from promotions

  3  where promo\_end\_date-promo\_begin\_date>30

  4  group by promo\_category;

PROMO\_CATEGORY                 SUM(PROMO\_COST)

------------------------------ ---------------

ad news                                      0

internet                               3703800

magazine                               2889600

flyer                                  2079800

newspaper                              5113600

post                                   3031500

TV                                     4850500

radio                                  1095000

8 rows selected.

sh@TESTDB> select promo\_category,SUM(promo\_cost)

  2  from promotions

  3  group by promo\_category

  4  having min(promo\_end\_date-promo\_begin\_date)>30;

PROMO\_CATEGORY                 SUM(PROMO\_COST)

------------------------------ ---------------

ad news                                      0

radio                                  1095000

两个语句都能执行成功。

找**所有PROMO\_CATEGORY 中的promo\_end\_date-promo\_begin\_date大于30的所有的promo\_cost的按promo\_category分组求和**

sh@TEST0910> select promo\_category,SUM(promo\_cost),promo\_end\_date-promo\_begin\_date

  2  from promotions

  3  where promo\_end\_date-promo\_begin\_date>30

  4  group by promo\_category,promo\_end\_date-promo\_begin\_date

  5  order by promo\_category;

PROMO\_CATEGORY                 SUM(PROMO\_COST) PROMO\_END\_DATE-PROMO\_BEGIN\_DATE

------------------------------ --------------- -------------------------------

TV                                     1267200                              31

TV                                     1333500                              61

TV                                      254700                              62

TV                                      904200                              90

TV                                      308000                              91

TV                                      782900                              92

ad news                                      0                             214

flyer                                   680000                              31

flyer                                   450300                              61

flyer                                   214500                              62

flyer                                   260700                              90

flyer                                   240200                              91

flyer                                   234100                              92

internet                                589800                              31

internet                               1195200                              61

internet                                249000                              62

internet                                568500                              90

internet                                398200                              91

internet                                703100                              92

magazine                                514900                              31

magazine                                781300                              61

magazine                                144000                              62

magazine                                404700                              91

magazine                               1044700                              92

newspaper                              1361500                              31

newspaper                              1325800                              61

newspaper                               664700                              62

newspaper                               473800                              90

newspaper                               707600                              91

newspaper                               580200                              92

post                                    838800                              31

post                                    587900                              61

post                                    277300                              62

post                                    156300                              90

post                                    296100                              91

post                                    875100                              92

radio                                   465700                              31

radio                                   145900                              61

radio                                   275800                              62

radio                                   137200                              91

radio                                    70400                              92

41 rows selected.

 min(promo\_end\_date-promo\_begin\_date)>30，找所有PROMO\_CATEGORY 中最小的promo\_end\_date-promo\_begin\_date，大于30。最小值不大于30，丢掉。

从下面看，TV,**flyer ,**internet ,magazine ,newspaper,post最小值均为29，不满足条件。

**radio  最小值是31满足大于31的条件，ad news  最小值是214，满足大于31的条件，故只有两个值。**

sh@TEST0910> select promo\_category,SUM(promo\_cost),promo\_end\_date-promo\_begin\_date

  2  from promotions

  3   group by promo\_category,promo\_end\_date-promo\_begin\_date

  4  order by promo\_category;

PROMO\_CATEGORY                 SUM(PROMO\_COST) PROMO\_END\_DATE-PROMO\_BEGIN\_DATE

------------------------------ --------------- -------------------------------

NO PROMOTION                                 0                               0

**TV                                      430600                              29**

TV                                       38900                              30

TV                                     1267200                              31

TV                                     1333500                              61

TV                                      254700                              62

TV                                      904200                              90

TV                                      308000                              91

TV                                      782900                              92

**ad news                                      0                             214**

**flyer                                    83600                              29**

flyer                                   680000                              31

flyer                                   450300                              61

flyer                                   214500                              62

flyer                                   260700                              90

flyer                                   240200                              91

flyer                                   234100                              92

internet                                526700                              29

internet                                 25700                              30

internet                                589800                              31

internet                               1195200                              61

internet                                249000                              62

internet                                568500                              90

internet                                398200                              91

internet                                703100                              92

magazine                                408500                              29

magazine                                104600                              30

magazine                                514900                              31

magazine                                781300                              61

magazine                                144000                              62

magazine                                404700                              91

magazine                               1044700                              92

newspaper                               406900                              29

newspaper                               160000                              30

newspaper                              1361500                              31

newspaper                              1325800                              61

newspaper                               664700                              62

newspaper                               473800                              90

newspaper                               707600                              91

newspaper                               580200                              92

post                                    196700                              29

post                                     84900                              30

post                                    838800                              31

post                                    587900                              61

post                                    277300                              62

post                                    156300                              90

post                                    296100                              91

post                                    875100                              92

**radio                                   465700                              31**

radio                                   145900                              61

radio                                   275800                              62

radio                                   137200                              91

radio                                    70400                              92

53 rows selected.

119. Examine the data in the ORD\_ITEMS table:

ORD\_NO     ITEM\_NO    QTY

1          111        10

1          222        20

1          333        30

2          333        30

2          444        40

3          111        40

Evaluate the following query:

SQL>SELECT item\_no, AVG(qty)

FROM ord\_items

HAVING AVG(qty) > MIN(qty) \* 2

GROUP BY item\_no;

Which statement is true regarding the outcome of the above query?

A. It gives an error because the HAVING clause should be specified after the GROUP BY clause.

B. It gives an error because all the aggregate functions used in the HAVING clause must be specified in

the SELECT list.

C. It displays the item nos with their average quantity where the average quantity is more than double the

minimum quantity of that item in the table.

D. It displays the item nos with their average quantity where the average quantity is more than double the

overall minimum quantity of all the items in the table.

Answer: C

答案解析：

scott@TESTDB> select \* from ord\_items;

    ORD\_NO    ITEM\_NO        QTY

---------- ---------- ----------

         1        111         10

         1        222         20

         1        333         30

         2        333         30

         2        444         40

         3        111         40

6 rows selected.

scott@TESTDB> select item\_no,avg(qty) from ord\_items group by item\_no;

   ITEM\_NO   AVG(QTY)

---------- ----------

       222         20

       444         40

       333         30

       111         25

scott@TESTDB> select item\_no,avg(qty) from ord\_items having avg(qty)>min(qty)\*2 group by item\_no;

   ITEM\_NO   AVG(QTY)

---------- ----------

       111         25

120. View the Exhibits  and examine the structures of the PRODUCTS,  SALES,  and CUSTOMERS   
tables.   
You issue the following query:   
SQL>SELECT p.prod\_id,prod\_name,prod\_list\_price,              
quantity\_sold,cust\_last\_name   
FROM products p NATURAL JOIN sales s NATURAL JOIN customers c   
WHERE prod\_id =148;   
Which statement is true regarding the outcome of this query?   
A. It executes successfully.   
B. It produces an error because the NATURAL join can be used only with two tables.   
C. It produces an error because a  column used in the  NATURAL join cannot have a qualifier.

D. It produces an error because all columns used in the NATURAL join should have a qualifier.   
Answer: C

sh@TEST0910> SELECT **p.prod\_id**,prod\_name,prod\_list\_price,quantity\_sold,cust\_last\_name

  2  FROM products p NATURAL JOIN sales s NATURAL JOIN customers c

  3  WHERE prod\_id =148 and rownum<6;

SELECT p.prod\_id,prod\_name,prod\_list\_price,quantity\_sold,cust\_last\_name

       \*

ERROR at line 1:

ORA-25155: column used in NATURAL join cannot have qualifier

**列用于自然连接不能有限定符。**

**去掉限定符后：**

sh@TEST0910> SELECT prod\_id,prod\_name,prod\_list\_price,quantity\_sold,cust\_last\_name

  2   FROM products p NATURAL JOIN sales s NATURAL JOIN customers c

  3  WHERE prod\_id =148 and rownum<6;

   PROD\_ID PROD\_NAME                      PROD\_LIST\_PRICE QUANTITY\_SOLD CUST\_LAST\_NAME

---------- ------------------------------ --------------- ------------- ------------------------------

       148 Xtend Memory                             20.99             1 Llyles

       148 Xtend Memory                             20.99             1 Lake

       148 Xtend Memory                             20.99             1 Koch

       148 Xtend Memory                             20.99             1 Skillman

       148 Xtend Memory                             20.99             1 Speer

**列不用于自然连接，可以使用限定符。**

sh@TEST0910>  SELECT prod\_id,**p.prod\_name**,prod\_list\_price,quantity\_sold,cust\_last\_name

  2  FROM products p NATURAL JOIN sales s NATURAL JOIN customers c

  3  WHERE prod\_id =148 and rownum<6;

   PROD\_ID PROD\_NAME                      PROD\_LIST\_PRICE QUANTITY\_SOLD CUST\_LAST\_NAME

---------- ------------------------------ --------------- ------------- ------------------------------

       148 Xtend Memory                             20.99             1 Llyles

       148 Xtend Memory                             20.99             1 Lake

       148 Xtend Memory                             20.99             1 Koch

       148 Xtend Memory                             20.99             1 Skillman

       148 Xtend Memory                             20.99             1 Speer

**自然连接:**

自然连接是在两张表中寻找那些数据类型和列名都相同的字段，然后自动地将他们连接起来，并返回所有符合条件按的结果。

来看一下自然连接的例子。

Select emp.ename,dept.dname

From emp natural join dept;

这里我们并没有指定连接的条件，实际上oracle为我们自作主张的将，emp中的deptno和dept中的deptno做了连接。

也就是实际上相当于

Select emp.ename,dept.dname

From emp join dept on emp.deptno = dept.deptno;

因为这两张表的这两个字段deptno的类型个名称完全相同。所以使用natural join时被自然的连接在一起了。

另外：

1.如果做自然连接的两个表的有多个字段都满足有相同名称个类型，那么他们会被作为自然连接的条件。

2.如果自然连接的两个表仅是字段名称相同，但数据类型不同，那么将会返回一个错误。

3.由于oracle中可以进行这种非常简单的natural join，我们在设计表时，应该尽量在不同表中具有相同含义的字段使用相同的名字和数据类型。以方便以后使用natural join。

121. Which two  statements are true regarding the USING clause in table joins? (Choose two .)

A. It can be used to join a maximum of three tables.

B. It can be used to restrict the number of columns used in a NATURAL join.

C. It can be used to access data from tables through equijoins as well as nonequijoins.

D. It can be used to join tables that have columns with the same name and compatible data types.

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

A,最多可以连接三个表，错误，可以连接三个以上的表

B,限制自然连接的等值连接数，正确，因为当有多个列相匹配时，使用USING子句可仅与一列相匹配，NATURAL JOIN和USING语句是互相排斥的。

C,错误，USING只能用在等值连接，不能用在非等值连接。

D,列的名称相同和数据类型匹配，则可以连接表，正确。

122. View the Exhibit for the structure of the STUDENT and FACULTY tables.

You need to display the faculty name followed by the number of students handled by the faculty at the

base location.

Examine the following two SQL statements:

Statement 1

SQL>SELECT faculty\_name,COUNT(student\_id)

FROM student JOIN faculty

USING (faculty\_id, location\_id)

GROUP BY faculty\_name;

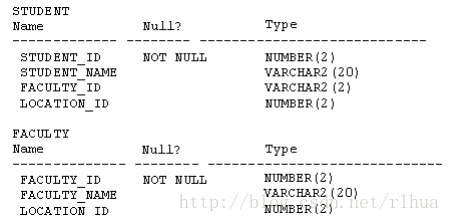
Statement 2

SQL>SELECT faculty\_name,COUNT(student\_id)

FROM student NATURAL JOIN faculty

GROUP BY faculty\_name;

Which statement is true regarding the outcome?



A. Only s tatement 1 executes successfully and gives the required result.

B. Only statement 2 executes successfully and gives the required result.

C. Both statements 1 and 2 execute successfully and give different results.

D. Both statements  1 and 2 execute successfully and give the same required result.

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

实验验证：

1、首先创建这两张表

[scott@TEST0924](mailto:scott@TEST0924)> create table student  
  2  (student\_id number(2) not null,  
  3  student\_name varchar2(20),  
  4  faculty\_id varchar2(2),  
  5  location\_id number(2)  
  6  );

Table created.

[scott@TEST0924](mailto:scott@TEST0924)> create table faculty  
  2  (faculty\_id number(2) not null,  
  3  faculty\_name varchar2(20),  
  4  location\_id number(2)  
  5  );

Table created.

2、向这两张表中插入数据

[scott@TEST0924](mailto:scott@TEST0924)> insert into student values(2,'zhansan3','20',11);

1 row created.

[scott@TEST0924](mailto:scott@TEST0924)> insert into faculty values(20,'zhansan2',11);

1 row created.

3、执行查询语句，可成功执行

[scott@TEST0924](mailto:scott@TEST0924)> SELECT faculty\_name,COUNT(student\_id) FROM student JOIN faculty   
  2  USING (faculty\_id, location\_id) GROUP BY faculty\_name;

FACULTY\_NAME         COUNT(STUDENT\_ID)  
-------------------- -----------------  
zhansan2                             1

[scott@TEST0924](mailto:scott@TEST0924)> SELECT faculty\_name,COUNT(student\_id) FROM student NATURAL JOIN faculty  
  2  GROUP BY faculty\_name;

FACULTY\_NAME         COUNT(STUDENT\_ID)  
-------------------- -----------------  
zhansan2                             1

4，如果向student表中的faculty\_id插入字符类。

[scott@TEST0924](mailto:scott@TEST0924)> insert into student values(2,'zhansan3','a1',11);

1 row created.

5、查询失败

[scott@TEST0924](mailto:scott@TEST0924)> SELECT faculty\_name,COUNT(student\_id) FROM student NATURAL JOIN faculty  
  2  GROUP BY faculty\_name;  
SELECT faculty\_name,COUNT(student\_id) FROM student NATURAL JOIN faculty  
\*  
ERROR at line 1:  
ORA-01722: invalid number

[scott@TEST0924](mailto:scott@TEST0924)> SELECT faculty\_name,COUNT(student\_id) FROM student JOIN faculty  
  2  USING (faculty\_id, location\_id) GROUP BY faculty\_name;  
USING (faculty\_id, location\_id) GROUP BY faculty\_name  
       \*  
ERROR at line 2:  
ORA-01722: invalid number

总结，当且仅当student表中的faculty\_id列能隐式转换为数字类型时，这两个查询语句才可以查询成功。

因为NATURAL JOIN是将两个表中具有相同名称的所有列都连接起来，即FACULTY\_ID和LOCATION\_ID,而USING子句也使用了这两个字段来连接，故结果相同。

123. View the Exhibits  and examine the structures of the PRODUCTS, SALES,  and CUSTOMERS

tables.

You need to generate a report that gives details of the customer's last name, name of the product,  and

the quantity sold for all customers in ' Tokyo' .

Which two queries give the required result? (Choose two.)

A. SELECT c.cust\_last\_name,p.prod\_name, s.quantity\_sold

FROM sales s JOIN products p

USING(prod\_id)

JOIN customers c

USING(cust\_id)

WHERE c.cust\_city='Tokyo';

B. SELECT c.cust\_last\_name, p.prod\_name, s.quantity\_sold

FROM products p JOIN sales s JOIN customers c

ON(p.prod\_id=s.prod\_id)

ON(s.cust\_id=c.cust\_id)

WHERE c.cust\_city='Tokyo';

C. SELECT c.cust\_last\_name, p.prod\_name, s.quantity\_sold

FROM products p JOIN sales s

ON(p.prod\_id=s.prod\_id)

JOIN customers c

ON(s.cust\_id=c.cust\_id)

AND c.cust\_city='Tokyo';

D. SELECT c.cust\_id,c.cust\_last\_name,p.prod\_id, p.prod\_name, s.quantity\_sold

FROM products p JOIN sales s

USING(prod\_id)

JOIN customers c

USING(cust\_id)

WHERE c.cust\_city='Tokyo';

Answer: AC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

B语法错误

D中p.prod\_id不应该跟限定的表别名。

**不要对USING子句中使用的列加以限定。**

USING子句中引用的那些列不能在SQL 语句的任何位置使用限定词（表名或别名）。

**如果在SQL 语句的另一个位置使用了同一列，则不要对其设置别名。**

124. View the Exhibit and examine the structure of the PROMOTIONS, SALES, and CUSTOMER tables.

You need to generate a report showing the promo name along with the customer name for all products

that were sold during their promo campaign and before 30th October 2007.

You issue the following query:

SQL> SELECT promo\_name,cust\_name

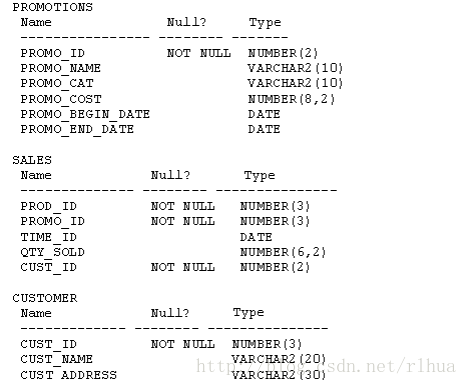
FROM promotions p JOIN sales s

ON(time\_id BETWEEN promo\_begin\_date AND promo\_end\_date)

JOIN customer c

ON (s.cust\_id = c.cust\_id) AND time\_id < '30-oct-2007';

Which statement is true regarding the above query?



A. It executes successfully and gives the required result.

B. It executes successfully but does not give the required result.

C. It produces an error because the join order of the tables is incorrect.

D. It produces an error because equijoin and nonequijoin conditions cannot be used in the same SELECT

statement.

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

题意要求：需要一个报表，显示所有产品的promo name和customer name，销售的产品是在promo campaign之内，并且在2007年10月30日以前

 缺少条件， promotions p JOIN sales s  之间应该有个PROMO\_ID的等值连接，才能得出正确结果。

125. Examine the structure of the CUSTOMERS table:

name                Null           Type

CUSTNO              NOT NULL       NUMBER(3)

CUSTNAME            NOT NULL       VARCHAR2(25)

CUSTADDRESS                       VARCHAR2(35)

CUST\_CREDIT\_LIMIT                 NUMBER(5)

CUSTNO is the PRIMARY KEY in the table. You want to find out if any customers' details have been

entered more than once using different CUSTNO, by listing all the duplicate names.

Which two methods can you use to get the required result? (Choose two.)

A. self-join

B. subquery

C. full outer-join with self-join

D. left outer-join with self-join

E. right outer-join with self-join

Answer: AB

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

题意：找到是否存在客户信息使用不同的CUSTNO录入了多次，显示所有重复的客户名称

自连接即是查询两次自身的表，首先根据客户信息来查看CUSTNO大于1的记录，然后再根据CUSTNO来显示客户名称。子查询也是一样。

126. View the Exhibit and examine the data in the PROJ\_TASK\_DETAILS table.

The PROJ\_TASK\_DETAILS table stores information about tasks involved in a project and the relation

between them.

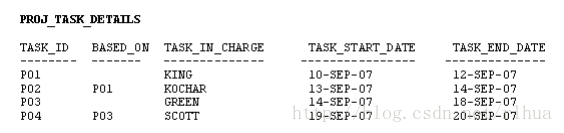
The BASED\_ON column indicates dependencies between tasks. Some tasks do not depend on the

completion of any other tasks.

You need to generate a report showing all task IDs, the corresponding task ID they are dependent on, and

the name of the employee in charge of the task it depends on.

Which query would give the required result?



A. SELECT p.task\_id, p.based\_on, d.task\_in\_charge

FROM proj\_task\_details p JOIN proj\_task\_details d

ON (p.based\_on = d.task\_id);

B. SELECT p.task\_id, p.based\_on, d.task\_in\_charge

FROM proj\_task\_details p LEFT OUTER JOIN proj\_task\_details d

ON (p.based\_on = d.task\_id);

C. SELECT p.task\_id, p.based\_on, d.task\_in\_charge

FROM proj\_task\_details p FULL OUTER JOIN proj\_task\_details d

ON (p.based\_on = d.task\_id);

D. SELECT p.task\_id, p.based\_on, d.task\_in\_charge

FROM proj\_task\_details p JOIN proj\_task\_details d

ON (p.task\_id = d.task\_id);

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

显示所有任务的ID，依赖对应任务的ID，和它依赖任务负责的员工姓名

A为只显示有依赖任务的task\_id

B是LEFT OUTER JOIN也就是左连表的所有task\_id全部显示出来，如果右边表没有匹配的信息，则显示空

127. Examine the data in the CUSTOMERS table:

CUSTNO    CUSTNAME    CITY

1         KING        SEATTLE

2         GREEN       BOSTON

3         KOCHAR      SEATTLE

4         SMITH       NEW YORK

You want to list all cities that have more than one customer along with the customer details.

Evaluate the following query:

SQL>SELECT c1.custname, c1.city

FROM Customers c1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Customers c2

ON (c1.city=c2.city AND c1.custname<>c2.custname);

Which two JOIN options can be used in the blank in the above query to give the correct output? (Choose

two.)

A. JOIN

B. NATURAL JOIN

C. LEFT OUTER JOIN

D. FULL OUTER JOIN

E. RIGHT OUTER JOIN

Answer: AE

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

*scott@TESTDB> SELECT \* FROM CUSTOMERS;*

*CUSTNO CUSTNAME   CITY*

*---------- ---------- ----------*

*1 KING       SEATTLE*

*2 GREEN      BOSTON*

*3 KOCHAR     SEATTLE*

*4 SMITH      NEW YORK*

你想要列出所有的城市,有一个以上的客户以及客户的细节

A

*scott@TESTDB>  select c1.custname,c1.city from customers c1 join customers c2*

*2  on (c1.city=c2.city and c1.custname<>c2.custname);*

*CUSTNAME   CITY*

*---------- ----------*

*KOCHAR     SEATTLE*

*KING       SEATTLE*

B，select列表中c1.city有限定

*scott@TESTDB>  select c1.custname,c1.city from customers c1 natural join customers c2*

*2  on (c1.city=c2.city and c1.custname<>c2.custname);*

*on (c1.city=c2.city and c1.custname<>c2.custname)*

*\**

*ERROR at line 2:*

*ORA-00933: SQL command not properly ended*

 C

*scott@TESTDB>  select c1.custname,c1.city from customers c1 left outer join customers c2*

*2  on (c1.city=c2.city and c1.custname<>c2.custname);*

*CUSTNAME   CITY*

*---------- ----------*

*KOCHAR     SEATTLE*

*KING       SEATTLE*

*SMITH      NEW YORK*

*GREEN      BOSTON*

 D

*scott@TESTDB>  select c1.custname,c1.city from customers c1 full outer join customers c2*

*2  on (c1.city=c2.city and c1.custname<>c2.custname);*

*CUSTNAME   CITY*

*---------- ----------*

*KOCHAR     SEATTLE*

*KING       SEATTLE*

*SMITH      NEW YORK*

*GREEN      BOSTON*

E

*scott@TESTDB>  select c1.custname,c1.city from customers c1 right outer join customers c2*

*2  on (c1.city=c2.city and c1.custname<>c2.custname);*

*CUSTNAME   CITY*

*---------- ----------*

*KING       SEATTLE*

*KOCHAR     SEATTLE*

128. View the Exhibits  and examine the structures of the  CUSTOMERS,  SALES,  and COUNTRIES

tables.

You need to generate a report that shows all country names,  with corresponding customers (if any) and

sales details (if any), for all customers.

Which FROM clause gives the required result?

A. FROM sales JOIN customers USING (cust\_id)

FULL OUTER JOIN countries USING (country\_id);

B. FROM sales JOIN customers USING (cust\_id)

RIGHT OUTER JOIN countries USING (country\_id);

C. FROM customers LEFT OUTER JOIN sales USING (cust\_id)

RIGHT OUTER JOIN countries USING (country\_id);

D. FROM customers LEFT OUTER JOIN sales USING (cust\_id)

LEFT OUTER JOIN countries USING (country\_id);

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

显示所有客户的所有country names，对应的customers（如果有）和sales details（如果有）

即是左外连接customers ，显示所有客户，右外连接countries ，显示所有 country names。

129. View the Exhibits  and examine the structures of the PROMOTIONS and SALES tables.

Evaluate the following SQL statement:

SQL>SELECT p.promo\_id, p.promo\_name, s.prod\_id

FROM sales s RIGHT OUTER JOIN promotions p

ON (s.promo\_id = p.promo\_id);

Which statement is true regarding the output of the above query?

A. It gives the details of promos for which there have been sales.

B. It gives the details of promos for which there have been no sales.

C. It gives details of all promos irrespectiveof whether they have resulted in a sale or not.

D. It gives details of product ID s that have been sold irrespective of whether they had a promo or not.

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

此处是右连接promotions ，即显示所有promotions 表中的信息，不管sales 表中promo\_id 是否有。

130. View the Exhibit and examine the data in the EMPLOYEES table:

You want to display all the employee names and their corresponding manager names.

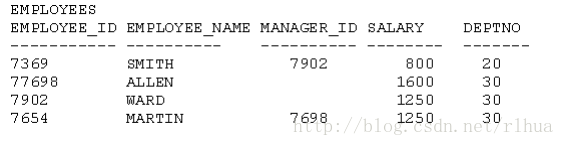
Evaluate the following query:

SQL> SELECT e.employee\_name "EMP NAME", m.employee\_name "MGR NAME"

FROM employees e \_\_\_\_\_\_\_\_\_\_\_\_\_\_ employees m

ON e.manager\_id = m.employee\_id;

Which JOIN option can be used in the blank in the above query to get the required output?



A. o nly inner JOIN

B. only FULL OUTER JOIN

C. only LEFT OUTER JOIN

D. only RIGHT OUTER JOIN

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

显示所有employee names和对应的manager names

此处e代表员工表，而m代表管理人员表，所有只有左连接才能显示 all the employee names

131. View the Exhibit and examine the structure of the PRODUCT, COMPONENT, and PDT\_COMP

tables.

In PRODUCT table, PDTNO is the primary key.

In COMPONENT table, COMPNO is the primary key.

In PDT\_COMP table, (PDTNO,COMPNO) is the primary key, PDTNO is the foreign key referencing

PDTNO in PRODUCT table and COMPNO is the foreign key referencing the COMPNO in COMPONENT

table.

You want to generate a report listing the product names and their corresponding component names, if the

component names and product names exist.

Evaluate the following query:

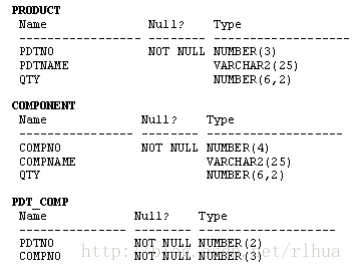
SQL>SELECT pdtno,pdtname, compno,compname

FROM product \_\_\_\_\_\_\_\_\_\_\_\_\_ pdt\_comp

USING (pdtno) \_\_\_\_\_\_\_\_\_\_\_ component USING(compno)

WHERE compname IS NOT NULL;

Which combination of joins used in the blanks in the above query gives the correct output?



A. JOIN; JOIN

B. FULL OUTER JOIN; FULL OUTER JOIN

C. RIGHT OUTER JOIN; LEFT OUTER JOIN 外键的关系

D. LEFT OUTER JOIN; RIGHT OUTER JOIN

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

listing the product names and their corresponding component names, if the

component names and product names exist

报表显示的是如果component names和product names存在，则显示product names和对应的component names

即显示PDT\_COMP表中所有的值

132. View the Exhibit and examine the structure of the SALES and PRODUCTS tables.

In the SALES table, PROD\_ID is the foreign key referencing PROD\_ID in the PRODUCTS table,

You want to list each product ID and the number of times it has been sold.

Evaluate the following query:

SQL>SELECT p.prod\_id, COUNT(s.prod\_id)

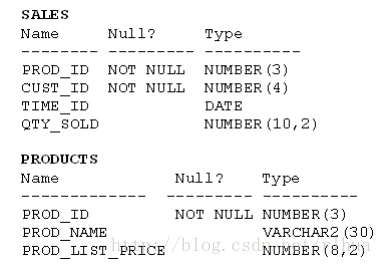
FROM products p \_\_\_\_\_\_\_\_\_\_\_\_\_ sales s

ON p.prod\_id  = s.prod\_id

GROUP BY p.prod\_id;

Which two JOIN options can be used in the blank in the above query to get the required output? (Choose

two.)



A. JOIN

B. FULL OUTER JOIN

C. LEFT OUTER JOIN

D. RIGHT OUTER JOIN

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12877591>

列出每一个产品ID和产品已销售的次数。

列出每一个产品ID，所以C正确，由于s表的prod\_id是products表的外键，所以是products的子集，所以B也正确。

133. Which two statements are true regarding subqueries? (Choose two.)

A. A subquery can retrieve zero or more rows.

B. Only two subqueries can be placed at one level.

C. A subquery can be used only in SQL query statements.

D. A subquery can appear on either side of a comparison operator

E. There is no limit on the number of subquery levels in the WHERE clause of a SELECT statement.

Answer: AD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

A,子查询可以返回一行或者多行，也可以返回0行。

B,只能有两个子查询被放在同一层，这句话不是很懂。

C,子查询不仅可以用在查询语句，也可以用在插入和更新语句。

D,子查询可以出现在任何一方的一个比较运算符。

E,WHERE子句中的嵌套子查询最多能嵌套255层

134. Where can subqueries be used? (Choose all that apply.)

A. field  names in the  SELECT statement  标量子查询

B. the  FROM clause in the  SELECT statement  内联视图

C. the  HAVING clause in the  SELECT statement   select的having子句

D. the  GROUP BY clause in the  SELECT statement

E. the  WHERE clause in only the  SELECT statement

F. the  WHERE clause in SELECT as well as all DML statements  select的where子句与所有的DML操作。

Answer: ABCF

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

子查询是一个SELECT语句，它嵌入到另一个SELECT语句的子句中。通过使用子查询，可以用简单的语句构建功能强大的语句。当需要从表中选择行，而选择条件却取决于该表自身中的数据时，子查询非常有用。

可以在许多SQL 子句中使用子查询，其中包括以下子句：

• WHERE子句

• HAVING子句

• FROM子句

在该语法中：

operator包括比较条件，例如>、=或IN

注：比较条件分为以下两类：单行运算符（>、=、>=、<、<>、<=）和多行运算符（IN、ANY、ALL、EXISTS）。

子查询通常被称为嵌套SELECT语句、子SELECT语句或内部SELECT语句。通常先执行子查询，然后使用其输出来完善主查询（即外部查询）的查询条件。

A,SELECT语句中的字段名称

B,SELECT语句的FROM子句中

C,SELECT语句的HAVING子句中

D,SELECT语句的GROUP BY子句中，不行

E,只能用于WHERE子句,错误，还可以用在HAVING和FROM子句。

F,SELECT与DML语句的WHERE子句中

135. Which three statements are true regarding subqueries? (Choose three.)

A. Subqueries can contain GROUP BY and ORDER BY clauses. 对，可以包含group by 和order by

B. Main query and subquery can get data from different tables. 对，可以分别拿不同表的数据

C. Main query and subquery must get data from the same tables. 错。看B选项

D. Subqueries can contain ORDER BY but not the  GROUP BY clause. 错，看A选项

E. Only one column or expression can be compared between the main query and subquery.

F. Multiple columns or expressions can be compared between the main query and subquery.

Answer: ABF

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

A,子查询可以包含GROUP BY和ORDER BY子句

B,主查询和子查询能从不同的表里获取数据

C,主查询和子查询必须从同一个表获取数据,错误，可以从不同表获取数据

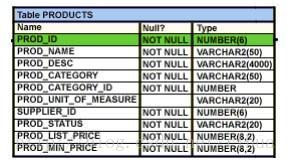
D,子查询能包含ORDER BY 子句，但是不能包含GROUP BY子句,错误，可以包含GROUP BY子句

E,主查询与子查询之间只能比较一个列或一个表达式，错误，可以比较多行，使用any，all等

F,主查询与子查询之间可以比较多个列或多个表达式.

136. View the Exhibit and examine the structure of the PRODUCTS table.

Which two tasks would require subqueries? (Choose two.)



A. Display the minimum list price for each product status.

B. Display all suppliers whose list price is less than 1000.

C. Display the number of products whose list price is more than the average list price.

D. Display the total number of products supplied by supplier 102 and have product status as 'obsolete'.

E. Display all products whose minimum list price is more than the average list price of products and have

the status 'orderable'.

Answer: CE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

题意要求两个需要使用子查询的。

A,显示每个product status的minimum list price,可以使用group by product status来找出 minimum list price

B,显示所有价格低于1000的供应商，找出供应商的价格，选择价格低于1000的，不需要使用子查询

C,显示产品价格大于平均价格的产品数量，首先需要计算出平均价格，然后在找出产品价格大于平均价格的数量，需要使用子查询。

D,显示102供应商供应的并且产品状态为'obsolete'的全部产品数量，直接查询，不需要使用子查询

E,显示产品的最低价格大于平均价格并且产品状态是'orderable'的所有产品,首先计算出平均价格，然后使用min（）函数找出最低价格来比较品均价格以及状态是'orderable'的所有产品，需要使用到子查询。

137. View the Exhibits and examine PRODUCTS and SALES tables.

You issue the following query to display product name and the number of times the product has been

sold:

SQL>SELECT p.prod\_name, i.item\_cnt

FROM (SELECT  prod\_id, COUNT(\*) item\_cnt

FROM sales

GROUP BY prod\_id) i RIGHT OUTER JOIN  products p

ON   i.prod\_id  = p.prod\_id;

What happens when the above statement is executed?

A. The statement executes successfully and produces the required output.

B. The statement produces an error because ITEM\_CNT cannot be displayed in the outer query.

C. The statement produces an error because a subquery in the FROM clause and outer-joins cannot be

used together.

D. The statement produces an error because the GROUP BY clause cannot be used in a subquery in the

FROM clause.

Answer: A

答案解析：

通过子查询，查询出产品卖出去的次数。

在通过右连接产品表，显示出所有产品。

实验验证：

sh@TEST0924> SELECT p.prod\_name, i.item\_cnt

  2  FROM (SELECT  prod\_id, COUNT(\*) item\_cnt FROM sales GROUP BY prod\_id)

  3   i RIGHT OUTER JOIN  products p

  4  ON   i.prod\_id  = p.prod\_id;

PROD\_NAME                                            ITEM\_CNT

-------------------------------------------------- ----------

Envoy External Keyboard                                  3441

SIMM- 8MB PCMCIAII card                                 19557

Mouse Pad                                               29282

External 6X CD-ROM                                      13043

...

72 rows selected.

138. Which statement is true regarding subqueries?

A. The LIKE operator cannot be used with single- row subqueries.

B. The NOT IN operator is equivalent to IS NULL with single- row subqueries.

C. =ANY and =ALL operators have the same functionality in multiple- row subqueries.

D. The NOT operator can be used with IN,  ANY,  and ALL operators in multiple- row subqueries.

Answer: D

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

A,LIKE操作符不能用在单行子查询中，错误，LIKE可以用于单行函数

B，NOT IN操作符相当于单选子查询的IS NULL，次两个不等价，NOT IN运算符等同于<> ALL，

C，多行子查询中的=ANY and =ALL操作符具有相同的功能，=ANY是等于任意一个，=ALL是等于所有

D，NOT操作可以用在多行子查询中的IN,ANY和ALL操作符上。

其实就是把NOT放到整个表达式之前：

WHERE NOT col IN (SELECT ...)  （也可以是WHERE col not IN (SELECT ...)）

WHERE NOT col = ANY (SELECT ...)  （大于、小于都可以）

WHERE NOT col = ALL (SELECT ...)  （大于、小于都可以）

139. Which three statements are true about multiple-row subqueries? (Choose three.)

A. They can contain a subquery within a subquery.

B. They can return multiple columns as well as rows.

C. They cannot contain a subquery within a subquery.

D. They can return only one column but multiple rows.

E. They can contain group functions and  GROUP BY and HAVING clauses.

F. They can contain group functions and the GROUP BY clause, but not the HAVING clause.

Answer: ABE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

A,子查询中能包含子查询,可以嵌套，但最多255层。

B，能返回多列和多行，正确。

C，子查询中不能包含子查询，错误，子查询可以嵌套

D，只能返回一列多行，错误，可以返回多行多列

E，可以包含组函数和GROUP BY,HAVING子句，正确

F，可以包含组函数和GROUP BY，但不能包含HAVING子句，错误，是可以包含HAVING子句的。

140. Examine the structure of the PRODUCTS table:

name          Null        Type

PROD\_ID       NOT NULL    NUMBER(4)

PROD\_NAME                VARCHAR2(20)

PROD\_STATUS              VARCHAR2(6)

QTY\_IN\_HAND              NUMBER(8,2)

UNIT\_PRICE               NUMBER(10,2)

You want to display the names of the products that have the highest total value for UNIT\_PRICE \*

QTY\_IN\_HAND.

Which SQL statement gives the required output?

A. SELECT prod\_name

FROM products

WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand)

FROM products);

B. SELECT prod\_name

FROM products

WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand)

FROM products

GROUP BY prod\_name);

C. SELECT prod\_name

FROM products

GROUP BY prod\_name

HAVING MAX(unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand)

FROM products

GROUP BY prod\_name);

D. SELECT prod\_name

FROM products

WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(SUM(unit\_price \* qty\_in\_hand))

FROM products)

GROUP BY prod\_name;

Answer: A

答案解析：

BC都犯了一个错误，都会返回一个错误single-row subquery returns more than one row。

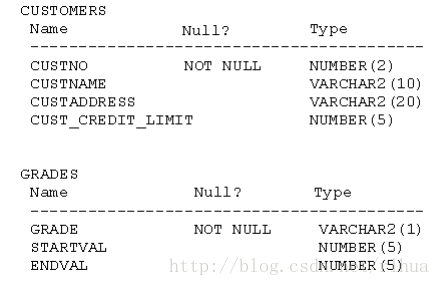
即是单行子查询返回了多行。该WHERE子句包含一个等于(=) 运算符，该运算符是一个只需要一个值的单行比较运算符。=运算符无法接受子查询中的多个值，因此产生了错误。

D的查询条件不满足题意。

141. View the Exhibit and examine the structure of CUSTOMERS and GRADES tables.

You need to display names and grades of customers who have the highest credit limit.

Which two SQL statements would accomplish the task? (Choose two.)



A. SELECT custname, grade

FROM customers, grades

WHERE (SELECT MAX(cust\_credit\_limit)

FROM customers) BETWEEN startval and endval;

B. SELECT custname, grade

FROM customers, grades

WHERE (SELECT MAX(cust\_credit\_limit)

FROM customers) BETWEEN startval and endval

AND cust\_credit\_limit BETWEEN startval AND endval;

C. SELECT custname, grade

FROM customers, grades

WHERE cust\_credit\_limit = (SELECT MAX(cust\_credit\_limit)

FROM customers)

AND cust\_credit\_limit BETWEEN startval AND endval;

D. SELECT custname, grade

FROM customers , grades

WHERE cust\_credit\_limit IN (SELECT MAX(cust\_credit\_limit)

FROM customers)

AND MAX(cust\_credit\_limit) BETWEEN startval AND endval;

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

题意要求显示有最高credit limit的用户的名称和等级。

A，缺少cust\_credit\_limit BETWEEN startval AND endval这个条件。

BC，使用的非等值连接

D有语法错误，WHERE子句里不能使用组函数

142. View the Exhibit and examine the structure of the PRODUCTS table.

Evaluate the following query:

SQL> SELECT prod\_name

FROM products

WHERE prod\_id IN (SELECT prod\_id FROM products

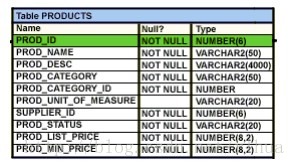
WHERE prod\_list\_price =

(SELECT MAX(prod\_list\_price)FROM products

WHERE prod\_list\_price <

(SELECT MAX(prod\_list\_price)FROM products)));

What would be the outcome of executing the above SQL statement?



A. It produces an error.

B. It shows the names of all products in the table.

C. It shows the names of products whose list price is the second highest in the table.

D. It shows the names of all  products whose list price is less than the maximum list price.

Answer: C

答案解析：

1、首先来运行一下上面的sql语句，是可以出结果的。

sh@TESTDB> SELECT prod\_name FROM products

  2  WHERE prod\_id IN (SELECT prod\_id FROM products

  3  WHERE prod\_list\_price =

  4  (SELECT MAX(prod\_list\_price)FROM products

  5  WHERE prod\_list\_price <

  6  (SELECT MAX(prod\_list\_price)FROM products)));

PROD\_NAME

--------------------------------------------------

Mini DV Camcorder with 3.5" Swivel LCD

2、拆分上面sql语句，首先运行子句查询出价格最高的产品的价格

sh@TESTDB> SELECT MAX(prod\_list\_price)FROM products;

MAX(PROD\_LIST\_PRICE)

--------------------

             1299.99

3、低于价格最高的，即价格次高的产品。按题意，故选C.

sh@TESTDB> SELECT MAX(prod\_list\_price)FROM products

  2  WHERE prod\_list\_price < 1299.99;

MAX(PROD\_LIST\_PRICE)

--------------------

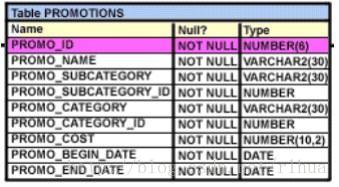
             1099.99

143. View the Exhibit and examine the structure of the PROMOTIONS table.

You have to generate a report that displays the promo name and start date for all promos that started after

the last promo in the 'INTERNET' category.

Which query would give you the required output?



A. SELECT promo\_name, promo\_begin\_date FROM promotions

WHERE promo\_begin\_date > ALL (SELECT MAX(promo\_begin\_date)

FROM promotions )AND

promo\_category = 'INTERNET';

B. SELECT promo\_name, promo\_begin\_date FROM promotions

WHERE promo\_begin\_date   IN (SELECT promo\_begin\_date

FROM promotions

WHERE promo\_category='INTERNET');

C. SELECT promo\_name, promo\_begin\_date FROM promotions

WHERE promo\_begin\_date  > ALL (SELECT promo\_begin\_date

FROM promotions

WHERE promo\_category = 'INTERNET');

D. SELECT promo\_name, promo\_begin\_date FROM promotions

WHERE promo\_begin\_date > ANY (SELECT promo\_begin\_date

FROM promotions

WHERE promo\_category = 'INTERNET');

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12879585>

显示在最后一个promo之后并且category为INTERNET开始的所有promos的promo name和start date。

A,promo\_category = 'INTERNET'应该再子查询里面。

B,条件不满足，条件要求找出最后一个promo之后并且category为INTERNET开始的所有promos的promo name和start date。

C,正确，大于ALL，即是大于最大值。

D,错误，大于ANY,即是大于最小值。

144. View the Exhibit and examine the structure of the PRODUCTS table.

You want to display the category with the maximum number of items.

You issue the following query:

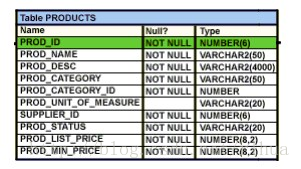
SQL>SELECT COUNT(\*),prod\_category\_id

FROM products

GROUP BY prod\_category\_id

HAVING COUNT(\*) = (SELECT MAX(COUNT(\*)) FROM products); 没有group by

What is the outcome?



A. It executes successfully and gives the correct output.

B. It executes successfully but does not give the correct output.

C. It generates an error because the subquery does not have a GROUP BY clause.

D. It generates an error because = is not valid and should be replaced  by the   IN operator.

Answer: C

答案解析：

1、首先来运行前半句，可见语法是正确的。

sh@TESTDB> SELECT COUNT(\*),prod\_category\_id

  2  FROM products

  3  GROUP BY prod\_category\_id

  4  HAVING COUNT(\*)=2;

  COUNT(\*) PROD\_CATEGORY\_ID

---------- ----------------

         2              202

2、再来运行子查询，报错，缺少group by函数。故选C，子查询缺少group by函数。

sh@TESTDB> SELECT MAX(COUNT(\*)) FROM products;

SELECT MAX(COUNT(\*)) FROM products

           \*

ERROR at line 1:

ORA-00978: nested group function without GROUP BY

145. View the Exhibit and examine the structure of the CUSTOMERS table.

You issue the following SQL statement on the CUSTOMERS table to display the customers who are in the

same country as customers with the last name 'KING' and whose credit limit is less than the maximum  小于最大值

credit limit in countries that have customers with the last name 'KING':

SQL> SELECT cust\_id,cust\_last\_name

FROM customers

WHERE country\_id IN(SELECT country\_id

FROM customers

WHERE cust\_last\_name ='King')

AND cust\_credit\_limit < (SELECT MAX(cust\_credit\_limit)

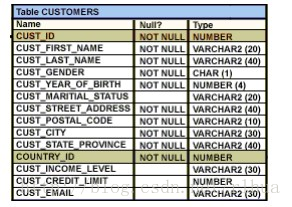
FROM customers

WHERE country\_id IN(SELECT country\_id

FROM customers

WHERE cust\_last\_name='King'));

Which statement is true regarding the outcome of the above query?



A. It executes and shows the required result.

B. It produces an error and the < operator should be replaced by < ALL to get the required output.

C. It produces an error and the < operator should be replaced by < ANY to get the required output.

D. It produces an error and the   IN operator should be replaced by = in the WHERE clause of the main

query to get the required output.

Answer: A

答案解析：

1、首先找出the last name 'KING'所在countries

sh@TEST0910> select country\_id from customers where cust\_last\_name='King';

COUNTRY\_ID

----------

     52772

     52790

     52790

     52779

2、再找出KING所在城市的最大的cust\_credit\_limit

sh@TEST0910> select MAX(cust\_credit\_limit) from customers where country\_id in

  2  (select country\_id from customers where cust\_last\_name='King');

MAX(CUST\_CREDIT\_LIMIT)

----------------------

                 15000

3、再找出和KING所在一样城市的客户的id和姓名，他们的cust\_credit\_limit要小于KING所在城市的最大的cust\_credit\_limit的其余的值。

值太多，去前5个。按题意，选A。

sh@TEST0910> SELECT cust\_id,cust\_last\_name FROM customers WHERE country\_id IN

  2  (select country\_id from customers where cust\_last\_name='King') and cust\_credit\_limit <

  3  (select MAX(cust\_credit\_limit) from customers where country\_id in

  4  (select country\_id from customers where cust\_last\_name='King')) and rownum<6;

   CUST\_ID CUST\_LAST\_NAME

---------- ----------------------------------------

     49671 Ruddy

      3228 Ruddy

      6783 Ruddy

     10338 Ruddy

     13894 Ruddy

146. Evaluate the following SQL statement:

SQL> SELECT cust\_id, cust\_last\_name

FROM customers

WHERE cust\_credit\_limit IN

(select cust\_credit\_limit

FROM customers

WHERE cust\_city ='Singapore');

Which statement is true regarding the above query if one of the values generated by the subquery is

NULL?

A. It produces an error.

B. It executes but returns no rows.

C. It generates output for NULL as well as the other values produced by the subquery.

D. It ignores the NULL value and generates output for the other values produced by the subquery.

Answer: C

答案解析：

这道题的意思是说：如果子查询出现的值是null会怎样？

1、首先运行子查询里面的sql语句，结果为null

sh@TEST0910> select cust\_credit\_limit

  2  FROM customers

  3  WHERE cust\_city ='Singapore';

no rows selected

2、如果子查询为null，那么结果也为null。

sh@TEST0910> SELECT cust\_id, cust\_last\_name FROM customers WHERE cust\_credit\_limit IN

  2  (select cust\_credit\_limit  FROM customers WHERE cust\_city ='Singapore');

no rows selected

147. View the Exhibit and examine the structure of the PROMOTIONS table.

Evaluate the following SQL statement:

SQL>SELECT promo\_name,CASE

WHEN promo\_cost >=(SELECT AVG(promo\_cost)

FROM promotions

WHERE promo\_category='TV')

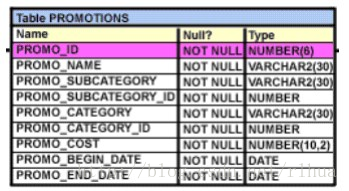
then 'HIGH'

else 'LOW'

END COST\_REMARK

FROM promotions;

Which statement is true regarding the outcome of the above query?



A. It shows COST\_REMARK for all the promos in the table. 分类显示所有值

B. It produces an error because the subquery gives an error.

C. It shows COST\_REMARK for all the promos in the promo category 'TV'.

D. It produces an error because subqueries cannot be used with the CASE expression.

Answer: A

答案解析：

1、首先运行子查询，得出一个数值

sh@TEST0910> SELECT AVG(promo\_cost) FROM promotions  WHERE promo\_category='TV';

AVG(PROMO\_COST)

---------------

     46260.8696

2、所有的promos 根据上面计算的数值，来进行分类。故选择A

sh@TEST0910> SELECT promo\_name,CASE WHEN promo\_cost >=

  2  (SELECT AVG(promo\_cost) FROM promotions  WHERE promo\_category='TV')

  3  then 'HIGH'  else 'LOW'END COST\_REMARK

  4  FROM promotions;

PROMO\_NAME                     COST

------------------------------ ----

NO PROMOTION #                 LOW

newspaper promotion #16-108    LOW

post promotion #20-232         LOW

newspaper promotion #16-349    LOW

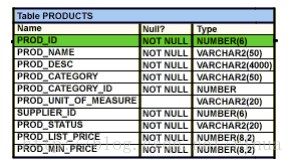
internet promotion #14-471     LOW

148. View the Exhibit and examine the structure of the PRODUCTS tables.

You want to generate a report that displays the average list price of product categories where the average

list price is less than half the maximum in each category.

Which query would give the correct output?



A. SELECT prod\_category,avg(prod\_list\_price)

FROM products

GROUP BY prod\_category

HAVING avg(prod\_list\_price) < ALL   小于所有，小于最小值

(SELECT max(prod\_list\_price)/2

FROM products

GROUP BY prod\_category);

B. SELECT prod\_category,avg(prod\_list\_price)

FROM products

GROUP BY prod\_category

HAVING avg(prod\_list\_price) > ANY

(SELECT max(prod\_list\_price)/2

FROM products

GROUP BY prod\_category);

C. SELECT prod\_category,avg(prod\_list\_price)

FROM products

HAVING avg(prod\_list\_price) < ALL

(SELECT max(prod\_list\_price)/2

FROM products

GROUP BY prod\_category);

D. SELECT prod\_category,avg(prod\_list\_price)

FROM products

GROUP BY prod\_category

HAVING avg(prod\_list\_price) > ANY

(SELECT max(prod\_list\_price)/2

FROM products);

Answer: A

答案解析：

1、首先查询子查询中出每个PROD\_CATEGORY的prod\_list\_price的最大值的1/2的值。

sh@TEST0924> SELECT max(prod\_list\_price)/2 FROM products

  2  GROUP BY prod\_category;

MAX(PROD\_LIST\_PRICE)/2

----------------------

                99.995

               649.995

               299.995

               549.995

               499.995

2、找出prod\_list\_price的平均值

sh@TEST0924> SELECT prod\_category,avg(prod\_list\_price)

  2  FROM products

  3  GROUP BY prod\_category;

PROD\_CATEGORY                                      AVG(PROD\_LIST\_PRICE)

-------------------------------------------------- --------------------

Software/Other                                               31.4130769

Hardware                                                        1149.99

Electronics                                                  121.682308

Photo                                                            230.19

Peripherals and Accessories                                  145.085238

3、平均值小于1/2最大值的所有值 ，即小于最小值

sh@TEST0924> SELECT prod\_category,avg(prod\_list\_price) FROM products GROUP BY prod\_category

  2  having avg(prod\_list\_price)<all(SELECT max(prod\_list\_price)/2 FROM products GROUP BY prod\_category);

PROD\_CATEGORY                                      AVG(PROD\_LIST\_PRICE)

-------------------------------------------------- --------------------

Software/Other                                               31.4130769

故选A答案。

B与题意不符，> ANY大于最小值

sh@TEST0924> SELECT prod\_category,avg(prod\_list\_price)

  2   FROM products

  3  GROUP BY prod\_category

  4  HAVING avg(prod\_list\_price) > ANY

  5  (SELECT max(prod\_list\_price)/2

  6  FROM products

  7  GROUP BY prod\_category);

PROD\_CATEGORY                                      AVG(PROD\_LIST\_PRICE)

-------------------------------------------------- --------------------

Hardware                                                        1149.99

Electronics                                                  121.682308

Photo                                                            230.19

Peripherals and Accessories                                  145.085238

C语法错误，缺少group by

sh@TEST0924> SELECT prod\_category,avg(prod\_list\_price)

  2  FROM products

  3  HAVING avg(prod\_list\_price) < ALL

  4  (SELECT max(prod\_list\_price)/2

  5  FROM products

  6  GROUP BY prod\_category);

SELECT prod\_category,avg(prod\_list\_price)

       \*

ERROR at line 1:

ORA-00937: not a single-group group function

D与题意不符，没有in each category分组。

sh@TEST0924> SELECT max(prod\_list\_price)/2 FROM products;

MAX(PROD\_LIST\_PRICE)/2

----------------------

               649.995

sh@TEST0924> SELECT prod\_category,avg(prod\_list\_price)

  2  FROM products

  3  GROUP BY prod\_category

  4  HAVING avg(prod\_list\_price) > ANY

  5  (SELECT max(prod\_list\_price)/2

  6  FROM products);

PROD\_CATEGORY                                      AVG(PROD\_LIST\_PRICE)

-------------------------------------------------- --------------------

Hardware                                                        1149.99

ALL参考：<http://blog.csdn.net/rlhua/article/details/12006433>

ANY参考：<http://blog.csdn.net/rlhua/article/details/12007309>

149. View the Exhibits and examine the structures of the COSTS and PROMOTIONS tables.

Evaluate the following SQL statement:

SQL> SELECT prod\_id FROM costs

WHERE promo\_id IN  (SELECT promo\_id FROM promotions

WHERE promo\_cost < ALL

(SELECT MAX(promo\_cost) FROM promotions

GROUP BY (promo\_end\_date-

promo\_begin\_date)));

What would be the outcome of the above SQL statement?

A. It displays prod IDs in the promo with the lowest cost.

B. It displays prod IDs in the promos with the lowest cost in the same time interval.

C. It displays prod IDs in the promos with the highest cost in the same time interval.

D. It displays prod IDs in the promos with cost less than the highest cost in the same time interval.

Answer: D

答案解析：

1、首先查询出promo\_end\_date- promo\_begin\_date期间promo\_cost的最大值

sh@TEST0924> SELECT MAX(promo\_cost) FROM promotions

  2  GROUP BY (promo\_end\_date- promo\_begin\_date);

MAX(PROMO\_COST)

---------------

              0

          98500

              0

          99900

         100000

          99000

          99300

          95900

          99600

          98000

10 rows selected.

2、<ALL，即小于1查询出来所有的的值，即小于最小值。

3、按题中查询，即查询出小于MAX(promo\_cost) 的所有的值。

sh@TEST0924>  SELECT prod\_id FROM costs

  2  WHERE promo\_id IN  (SELECT promo\_id FROM promotions

  3  WHERE promo\_cost < ALL

  4  (SELECT MAX(promo\_cost) FROM promotions

  5  GROUP BY (promo\_end\_date-

  6  promo\_begin\_date)));

no rows selected

ALL参考：<http://blog.csdn.net/rlhua/article/details/12006433>

ANY参考：<http://blog.csdn.net/rlhua/article/details/12007309>

此题与148题相似：<http://blog.csdn.net/rlhua/article/details/12011533>

150. View the Exhibit and examine the data in the PROMOTIONS table.

You need to display all promo categories that do not have 'discount' in their subcategory.

Which two SQL statements give the required result? (Choose two.)

A. SELECT promo\_category

FROM promotions

MINUS

SELECT promo\_category

FROM promotions

WHERE promo\_subcategory = 'discount';

所有promo\_category 减去包含promo\_subcategory = 'discount'的，符合题意。

B. SELECT promo\_category

FROM promotions

INTERSECT 交集

SELECT promo\_category

FROM promotions

WHERE promo\_subcategory = 'discount';

所有promo\_category 中包含promo\_subcategory = 'discount'的，与题意相反。

C. SELECT promo\_category

FROM promotions

MINUS

SELECT promo\_category

FROM promotions

WHERE promo\_subcategory <> 'discount';

所有promo\_category 中减去不包含promo\_subcategory = 'discount'的，与题意相反。

D. SELECT promo\_category

FROM promotions

INTERSECT

SELECT promo\_category

FROM promotions

WHERE promo\_subcategory <> 'discount';

所有promo\_category 交集不包含promo\_subcategory = 'discount'的，符合题意。

Answer: AD

答案解析：

本题意是让找出不包含promo\_subcategory = 'discount'的产品。

MINUS 返回第一个查询有，第二个查询无的

which returns only unique rows returned by the first query but not by the second

INTERSECT 返回共同有的

which returns only those unique rows returned by both queries

151. View the Exhibit and examine the structure of the CUSTOMERS and CUST\_HISTORY tables.

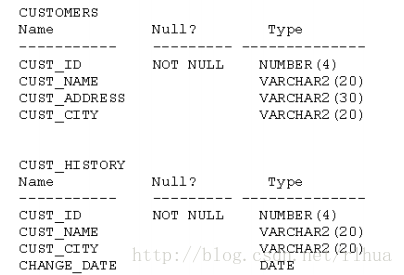
The CUSTOMERS  table  contains  the  current  location  of  all  currently  active  customers. The

CUST\_HISTORY table stores historical details relating to any changes in the location of all current as well

as previous customers who are no longer active with the company.

You need to find those customers who have never changed their address.

Which SET operator would you use to get the required output?



A. MINUS

B. UNION

C. INTERSECT

D. UNION ALL

Answer: A

答案解析：

本题意是让找出没有改变过地址的客户。即从CUSTOMERS  减去改变过地址的CUST\_HISTORY 的客户即可。

UNION Example The following statement combines the results of two queries with the UNION operator, which eliminates duplicate selected rows. This statement shows that you must match data type (using the TO\_CHAR function) when columns do not exist in one or the other table

所有的行都包括

UNION ALL Example The UNION operator returns only distinct rows that appear in either result, while the UNION ALL operator returns all rows. The UNIONALL operator does not eliminate duplicate selected rows

去除重复的行

INTERSECT Example The following statement combines the results with the INTERSECT operator, which returns only those unique rows returned by both queries

返回共同有的唯一的行

MINUS Example The following statement combines results with the MINUS operator, which returns only unique rows returned by the first query but not by the second

返回第一个有，第二个表无的行

官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/queries004.htm#i2054381>

152. Which statement is true regarding the UNION operator?

A. By default,   the output is not sorted.

B. NULL values are not ignored during duplicate checking.

C. Names of all columns must be identical across all SELECT statements.

D. The number of columns selected in all SELECT statements need not be the same.

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12883007>

UNION运算符用于返回由任一查询选定的所有行。使用UNION运算符可以返回多个表中的所有行，但不包括重复行。

准则

• 所选列的数量必须相同。

• 所选列的数据类型必须属于相同的数据类型组（如数字或字符）。

• 列名不必相同。

• UNION将对所有选定的列执行操作。

• 在重复项检查过程中不会忽略NULL值。

• 默认情况下，输出按SELECT子句中列的升序进行排序。

A,默认情况下，结果不排序，错误，在默认的情况下，输出按SELECT子句中列的升序进行排序。

B，在重复项检查过程中不会忽略NULL值，正确。

C，列名在所有SELECT语句中必须是相同的，错误，列名不必相同。

D，在所有SELECT语句中选择的列的数量不需要相同，错误，所选列的数量必须相同。

153. View the Exhibits and examine the structures of the PRODUCTS and SALES tables.

Which two SQL statements would give the same output? (Choose two.)

A. SELECT prod\_id FROM products

INTERSECT

SELECT prod\_id FROM sales;

B. SELECT prod\_id FROM products

MINUS

SELECT prod\_id FROM sales;

C. SELECT DISTINCT  p.prod\_id

FROM products p JOIN sales s

ON p.prod\_id=s.prod\_id;

D. SELECT DISTINCT p.prod\_id

FROM products p JOIN sales s

ON p.prod\_id <> s.prod\_id;

Answer: AC

答案解析：

A：所有的产品和已卖出的产品相交，即返回已经卖出去的产品的id。

sh@TEST0924> SELECT prod\_id FROM products

  2  INTERSECT

  3  SELECT prod\_id FROM sales;

   PROD\_ID

----------

        13

        14

        15

        16

...

       148

72 rows selected.默认升序排列。

B:所有的产品和已卖出去的产品相减，即返回没有卖出去的产品。

sh@TEST0924> SELECT prod\_id FROM products

  2  MINUS

  3  SELECT prod\_id FROM sales;

no rows selected

C:返回已卖出去的产品的id

sh@TEST0924> SELECT DISTINCT  p.prod\_id

  2   FROM products p JOIN sales s

  3  ON p.prod\_id=s.prod\_id;

   PROD\_ID

----------

        22

        25

...

       139

72 rows selected.

D答案：会把所有的都显示出来

sh@TEST0924> SELECT DISTINCT p.prod\_id

  2  FROM products p JOIN sales s

  3  ON p.prod\_id <> s.prod\_id;

   PROD\_ID

----------

        13

        14

...

       148

72 rows selected.

INTERSECT Example The following statement combines the results with the INTERSECT operator, which returns only those unique rows returned by both queries

返回共同有的唯一的行

MINUS Example The following statement combines results with the MINUS operator, which returns only unique rows returned by the first query but not by the second

返回第一个有，第二个表无的行

官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/queries004.htm#i2054381>

154. View the Exhibit and evaluate structures of the SALES, PRODUCTS, and COSTS tables.

Evaluate the following SQL statement:

SQL>SELECT prod\_id FROM products

INTERSECT

SELECT prod\_id FROM sales

MINUS

SELECT prod\_id FROM costs;

Which statement is true regarding the above compound query?

A. It produces an error.

B. It shows products that were sold  and have a cost recorded.

C. It shows products that were sold  but have  no cost recorded.

D. It shows products that  have a cost recorded irrespective of sales.

Answer: C

答案解析：

按照题意，查询出在sales表里的已经卖出去的但是不在costs表里的产品id。

INTERSECT Example The following statement combines the results with the INTERSECT operator, which returns only those unique rows returned by both queries

返回共同有的唯一的行

MINUS Example The following statement combines results with the MINUS operator, which returns only unique rows returned by the first query but not by the second

返回第一个有，第二个表无的行

官方参考：<http://docs.oracle.com/cd/E11882_01/server.112/e41084/queries004.htm#i2054381>

155. Evaluate the following SQL statement:

SQL> SELECT promo\_id, promo\_category

FROM promotions

WHERE promo\_category = 'Internet' **ORDER BY 2 DESC**

UNION

SELECT promo\_id, promo\_category

FROM promotions

WHERE promo\_category = 'TV'

UNION

SELECT promo\_id, promo\_category

FROM promotions

WHERE promo\_category ='Radio';

Which statement is true regarding the outcome of the above query?

A. It executes successfully and displays rows in the descending order of PROMO\_CATEGORY.

B. It produces an error because positional notation cannot be used in the  ORDER BY clause with SET

operators.

C. It executes successfully but ignores the ORDER BY clause because it is not located at the end of the

compound statement.

D. It produces an error because the ORDER BY clause should appear only at the end of a compound

query-that is, with the last SELECT statement.  ORDER BY必须要在句子最后

Answer: D

答案解析：

ORDER BY 子句应该放在复合sql查询语句的最后。

sh@TEST0924> SELECT promo\_id, promo\_category

  2  FROM promotions

  3  WHERE promo\_category = 'Internet' ORDER BY 2 DESC

  4  UNION

  5  SELECT promo\_id, promo\_category

  6  FROM promotions

  7  WHERE promo\_category = 'TV'

  8  UNION

  9  SELECT promo\_id, promo\_category

 10  FROM promotions

 11  WHERE promo\_category ='Radio';

UNION

\*

ERROR at line 4:

ORA-00933: SQL command not properly ended

正确的是：

sh@TEST0924> SELECT promo\_id, promo\_category

  2  FROM promotions

  3  WHERE promo\_category = 'Internet'

  4  UNION

  5  SELECT promo\_id, promo\_category

  6  FROM promotions

  7  WHERE promo\_category = 'TV'

  8  UNION

  9  SELECT promo\_id, promo\_category

 10  FROM promotions

 11  WHERE promo\_category ='Radio'

 12  ORDER BY 2 DESC;

  PROMO\_ID PROMO\_CATEGORY

---------- ------------------------------

        35 TV

        42 TV

        47 TV

        49 TV

        54 TV

...

       532 TV

       534 TV

115 rows selected.

156. Evaluate the following SQL statement:

SQL> SELECT cust\_id, cust\_last\_name "Last Name"

FROM customers

WHERE country\_id = 10

UNION

SELECT cust\_id CUST\_NO, cust\_last\_name

FROM customers

WHERE country\_id = 30;

Which ORDER BY clauses are valid for the above query? (Choose all that apply.)

A. ORDER BY 2,1

B. ORDER BY CUST\_NO

C. ORDER BY 2,cust\_id

D. ORDER BY "CUST\_NO"

E. ORDER BY "Last Name"

Answer: ACE

答案解析：

ORDER BY 后面可以跟数字，可以完全一摸一样的列名,经过union后，字段名根据第一个查询语句显示，变为cust\_id和Last Name。

sh@TEST0924> SELECT cust\_id, cust\_last\_name "Last Name" from customers where country\_id=52770 and rownum<5

  2  union

  3  select cust\_id CUST\_NO,cust\_last\_name from customers where country\_id=52775 and rownum<5;

   CUST\_ID Last Name

---------- ----------------------------------------

      3228 Ruddy

      6783 Ruddy

     10338 Ruddy

     13894 Ruddy

     21005 Ruddy

     37004 Embrey

     40559 Embrey

     49671 Ruddy

8 rows selected.

A答案：

sh@TEST0924> SELECT cust\_id, cust\_last\_name "Last Name" from customers where country\_id=52770 and rownum<5

  2  union

  3  select cust\_id CUST\_NO,cust\_last\_name from customers where country\_id=52775 and rownum<5

  4  ORDER BY 2,1;

   CUST\_ID Last Name

---------- ----------------------------------------

     37004 Embrey

     40559 Embrey

      3228 Ruddy

      6783 Ruddy

     10338 Ruddy

     13894 Ruddy

     21005 Ruddy

     49671 Ruddy

8 rows selected.

B答案：报错，没有CUST\_NO字段。

sh@TEST0924> SELECT cust\_id, cust\_last\_name "Last Name" from customers where country\_id=52770 and rownum<5

  2  union

  3   select cust\_id CUST\_NO,cust\_last\_name from customers where country\_id=52775 and rownum<5

  4  ORDER BY CUST\_NO;

ORDER BY CUST\_NO

         \*

ERROR at line 4:

ORA-00904: "CUST\_NO": invalid identifier

C答案：

sh@TEST0924> SELECT cust\_id, cust\_last\_name "Last Name" from customers where country\_id=52770 and rownum<5

  2  union

  3  select cust\_id CUST\_NO,cust\_last\_name from customers where country\_id=52775 and rownum<5

  4  ORDER BY 2,cust\_id

  5  ;

   CUST\_ID Last Name

---------- ----------------------------------------

     37004 Embrey

     40559 Embrey

      3228 Ruddy

      6783 Ruddy

     10338 Ruddy

     13894 Ruddy

     21005 Ruddy

     49671 Ruddy

8 rows selected.

D答案：报错

sh@TEST0924> SELECT cust\_id, cust\_last\_name "Last Name" from customers where country\_id=52770 and rownum<5

  2  union

  3  select cust\_id CUST\_NO,cust\_last\_name from customers where country\_id=52775 and rownum<5

  4  ORDER BY "CUST\_NO"

  5  ;

ORDER BY "CUST\_NO"

         \*

ERROR at line 4:

ORA-00904: "CUST\_NO": invalid identifier

E答案：

sh@TEST0924> SELECT cust\_id, cust\_last\_name "Last Name" from customers where country\_id=52770 and rownum<5

  2  union

  3  select cust\_id CUST\_NO,cust\_last\_name from customers where country\_id=52775 and rownum<5

  4  ORDER BY "Last Name"

  5  ;

   CUST\_ID Last Name

---------- ----------------------------------------

     37004 Embrey

     40559 Embrey

      3228 Ruddy

      6783 Ruddy

     10338 Ruddy

     13894 Ruddy

     21005 Ruddy

     49671 Ruddy

8 rows selected.

157. View the Exhibit and examine the structure of the ORDERS and CUSTOMERS tables.

Evaluate the following SQL command:

SQL> SELECT o.order\_id, c.cust\_name, o.order\_total, c.credit\_limit

FROM orders o JOIN customers c

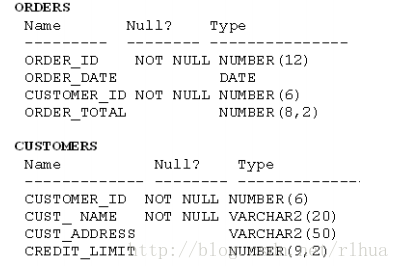
USING (customer\_id)

WHERE o.order\_total > c.credit\_limit

FOR UPDATE

ORDER BY o.order\_id;

Which two statements are true regarding the outcome of the above query? (Choose two.)



A. It locks all the rows that satisfy the condition in the statement.

B. It locks only the columns that satisfy the condition in both the tables.

C. The locks are released only when a COMMIT or ROLLBACK is issued.

D. The locks are released after  a DML statement   is executed on the locked rows.

Answer: AC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

For update of 实际列名（可以加of，也可以省略of，如果加上of就是锁定指定的表，而列并不重要，只要是该表中的实际列名，不可以使用列别名，如果不加of，则锁定满足条件的所有行）

A,锁定语句中满足条件的所有行,正确

B,只锁定两个表中满足条件的列。错误，是满足条件的所有行，不是列。

C，只有执行COMMIT或ROLLBACK后，锁才会释放，正确。

D，锁定的行上执行DML语句后，锁被释放，错误。如C。

158. Which statements are true regarding the FOR UPDATE clause in a SELECT statement? (Choose all

that apply.)

A. It locks only the columns specified in the SELECT list.

B. It locks the rows that satisfy the condition in the SELECT statement.

C. It can be used only in SELECT statements that are based on a single table.

D. It can be used in SELECT statements that are based on a single or multiple tables.

E. After   it is enforced by a SELECT statement, no other query can access the same rows until a

COMMIT or ROLLBACK is issued.

Answer: BD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

A，只锁定SELECT列表中指定的列，错误，是指定的行。

B，锁定SELECT语句中满足条件的行，正确。

C，只能用于基于单个表的SELECT语句中，错误。可以针对多个表在SELECT语句中使用FOR UPDATE子句。

D，可以用于基于单个表或多个表的SELECT语句中，正确。

E，用SELECT语句执行之后，其它查询直到执行COMMIT或ROLLBACK后才能访问相同的行，错误，即使被锁定，也是可以查询的。

159. View the Exhibit and examine the structure of the CUSTOMERS table.

NEW\_CUSTOMERS is a new table with the columns CUST\_ID, CUST\_NAME and CUST\_CITY that

have the same data types and size as the corresponding columns in the CUSTOMERS table.

Evaluate the following INSERT statement:

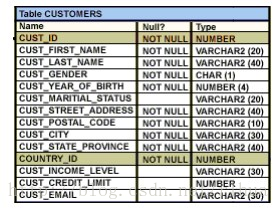
INSERT INTO new\_customers (cust\_id, cust\_name, cust\_city)

VALUES(SELECT cust\_id,cust\_first\_name' 'cust\_last\_name,cust\_city    去掉

FROM customers

WHERE cust\_id > 23004);

The INSERT statement fails when executed. What could be the reason?



A. The VALUES clause cannot be used in an INSERT with a subquery. values后面不能跟子查询，或者直接不要VALUES

B. Column names in the NEW\_CUSTOMERS and CUSTOMERS tables do not match.

C. The WHERE clause cannot be used in a subquery embedded in an INSERT statement.

D. The total number of columns in the NEW\_CUSTOMERS table does not match the total number of

columns in the CUSTOMERS table.

Answer: A

答案解析：

即insert into后面的如果有子句，则不用values

sh@TESTDB> create table new\_sustomers as select cust\_id,cust\_last\_name,cust\_city

  2  from customers where 1=0;

Table created.

sh@TESTDB> insert into new\_sustomers(CUST\_ID, CUST\_LAST\_NAME,CUST\_CITY)

  2  values

  3  (select cust\_id,cust\_first\_name||cust\_last\_name,cust\_city  from customers where cust\_id>23004)

  4  ;

(select cust\_id,cust\_first\_name||cust\_last\_name,cust\_city  from customers where cust\_id>23004)

\*

ERROR at line 3:

ORA-00936: missing expression

去掉values后正常

sh@TESTDB> insert into new\_sustomers(CUST\_ID, CUST\_LAST\_NAME,CUST\_CITY)

  2  (select cust\_id,cust\_first\_name||cust\_last\_name,cust\_city  from customers where cust\_id>23004)

  3  ;

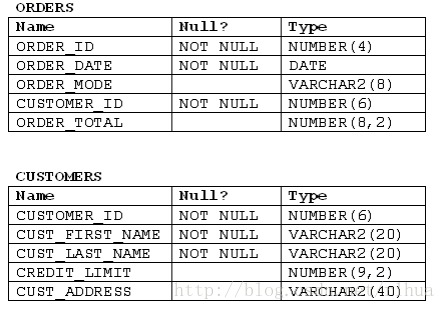
32496 rows created.

160. View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables.

There is only one customer with the cust\_last\_name column having value Roberts. Which INSERT

statement  should  be  used  to  add  a  row  into  the ORDERS  table  for  the  customer whose

CUST\_LAST\_NAME is Roberts and CREDIT\_LIMIT is 600?



A. INSERT INTO  orders

VALUES (1,'10-mar-2007', 'direct',

(SELECT customer\_id

FROM customers

WHERE cust\_last\_name='Roberts' AND

credit\_limit=600), 1000);

B. INSERT INTO orders (order\_id,order\_date,order\_mode,

(SELECT customer\_id

FROM customers

WHERE cust\_last\_name='Roberts' AND

credit\_limit=600),order\_total)

VALUES(1,'10-mar-2007', 'direct', &&customer\_id, 1000);

C. INSERT INTO(SELECT o.order\_id, o.order\_date,o.order\_mode,c.customer\_id, o.order\_total

FROM orders o, customers c

WHERE o.customer\_id = c.customer\_id

AND c.cust\_last\_name='Roberts' ANDc.credit\_limit=600 )

VALUES (1,'10-mar-2007', 'direct',(SELECT customer\_id

FROM customers

WHERE cust\_last\_name='Roberts' AND

credit\_limit=600), 1000);

D. INSERT INTO orders (order\_id,order\_date,order\_mode,

(SELECT customer\_id

FROM customers

WHERE      cust\_last\_name='Roberts' AND

credit\_limit=600),order\_total)

VALUES(1,'10-mar-2007', 'direct', &customer\_id, 1000);

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

只有一个customer的cust\_last\_name列的值是Roberts。

题意问哪一个INSERT语句能被用于给ORDERS添加一行，customer的cust\_last\_name为Roberts，并且CREDIT\_LIMIT为600，只有A正确

BCD语法错误。

161. View the exhibit and examine the description for the SALES and CHANNELS tables.

You issued the following SQL statement to insert a row in the SALES table:

INSERT INTO sales VALUES

(23, 2300, SYSDATE, (SELECT channel\_id

FROM channels

WHERE channel\_desc='Direct Sales'), 12, 1, 500);

Which statement is true regarding the execution of the above statement?

A. The statement will execute and the new row will be inserted in the SALES   table.

B. The statement will fail because subquery cannot be used in the VALUES clause.

C. The statement will fail because the  VALUES clause is not required with subquery.

D. The statement will fail because subquery in the VALUES clause is not enclosed with in single quotation

marks .

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

可以在values括号中使用子查询

  该语句可以执行成功，前提是子查询的结果是一个值。

 这里需要注意的是：图中只有6个字段，但实际在sh用户下sales表中确有7个字段，此题是按照7个字段来的。

[sh@TEST1107](mailto:sh@TEST1107)> desc sales;  
 Name                                                  Null?    Type  
 ----------------------------------------------------- -------- ------------------------------------  
 PROD\_ID                                               NOT NULL NUMBER  
 CUST\_ID                                               NOT NULL NUMBER  
 TIME\_ID                                               NOT NULL DATE  
 CHANNEL\_ID                                            NOT NULL NUMBER  
 PROMO\_ID                                              NOT NULL NUMBER  
 QUANTITY\_SOLD                                         NOT NULL NUMBER(10,2)  
 AMOUNT\_SOLD                                           NOT NULL NUMBER(10,2)

162. View the Exhibit and examine the structure of the PRODUCTS, SALES, and SALE\_SUMMARY

tables.

SALE\_VW   is a view created using the following command :

SQL>CREATE VIEW sale\_vw AS

SELECT prod\_id, SUM(quantity\_sold) QTY\_SOLD

FROM sales GROUP BY prod\_id;

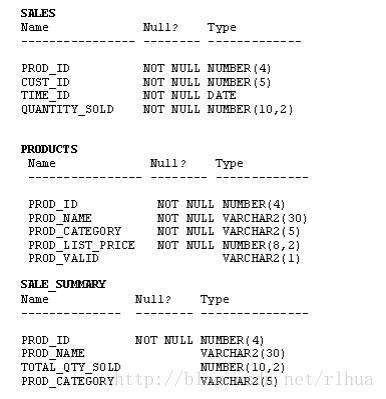
You issue the following command to add a row to the  SALE\_SUMMARY   table :

SQL>INSERT INTO sale\_summary

SELECT prod\_id, prod\_name, qty\_sold FROM sale\_vw JOIN products

USING (prod\_id) WHERE prod\_id = 16;

What is the outcome?



A. It executes successfully.

B. It gives an error because a complex view cannot be used to add data into the SALE\_SUMMARY table.

C. It gives an error because the column names in the subquery and the SALE\_SUMMARY table do not

match.

D. It gives an error because the number of columns to be inserted does not match with the number of

columns in the SALE\_SUMMARY table.

Answer: D

答案解析：

参考：http://blog.csdn.net/rlhua/article/details/12885143

由于列的数量不匹配造成的，如果数量一致，会执行成功。

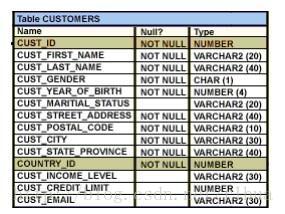
 带有子查询的INSERT语句，勿使用VALUES子句，并且使INSERT子句中的列数与子查询中的列数匹配。

163. View the Exhibit and examine the description for the CUSTOMERS table.

You want  to update  the CUST\_CREDIT\_LIMIT column  to NULL for all  the customers, where

CUST\_INCOME\_LEVEL has NULL in the CUSTOMERS table. Which SQL statement will accomplish the

task?



A. UPDATE customers

SET cust\_credit\_limit = NULL

WHERE CUST\_INCOME\_LEVEL = NULL;

B. UPDATE customers

SET cust\_credit\_limit = NULL

WHERE cust\_income\_level IS NULL;

C. UPDATE customers

SET cust\_credit\_limit = TO\_NUMBER(NULL)

WHERE cust\_income\_level = TO\_NUMBER(NULL);

D. UPDATE customers

SET cust\_credit\_limit = TO\_NUMBER(' ',9999)

WHERE cust\_income\_level IS NULL;

Answer: B

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

题意要求：更新CUSTOMERS表中所有customers的CUST\_CREDIT\_LIMIT值，当CUST\_INCOME\_LEVEL为NULL时更新CUST\_CREDIT\_LIMIT为空。

AC的where条件都为假，所有都不更新。

D的set语句有语法问题，故错误。

 实验验证：

A,

*sh@TESTDB> UPDATE customers*

*2  SET cust\_credit\_limit = NULL*

*3  WHERE CUST\_INCOME\_LEVEL = NULL;*

*0 rows updated.*

 B,

*sh@TESTDB> UPDATE customers*

*2  SET cust\_credit\_limit = NULL*

*3  WHERE cust\_income\_level IS NULL*

*4  ;*

*41 rows updated.*

 C,

*sh@TESTDB> UPDATE customers*

*2  SET cust\_credit\_limit = TO\_NUMBER(NULL)*

*3  WHERE cust\_income\_level = TO\_NUMBER(NULL);*

*0 rows updated.*

 D,

sh@TESTDB> UPDATE customers

  2  SET cust\_credit\_limit = TO\_NUMBER(' ',9999)

  3  WHERE cust\_income\_level IS NULL;

SET cust\_credit\_limit = TO\_NUMBER(' ',9999)

                                  \*

ERROR at line 2:

ORA-01722: invalid number

164. View the Exhibit and examine the structure of CUSTOMERS and SALES tables.

Evaluate the following SQL statement:

UPDATE (SELECT prod\_id, cust\_id, quantity\_sold, time\_id

FROM sales)

SET time\_id = '22-MAR-2007'

WHERE cust\_id = (SELECT cust\_id

FROM customers

WHERE cust\_last\_name = 'Roberts' AND

credit\_limit = 600);

Which statement is true regarding the execution of the above UPDATE statement?

A. It would not execute because two tables cannot be used in a single UPDATE statement.

B. It would not execute because the SELECT statement cannot be used in place of the table name.

C. It would execute and restrict modifications to only the columns specified in the SELECT statement.

D. It would not execute because a subquery cannot be used in the WHERE clause of an UPDATE

statement.

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

UPDATE后面的SELECT语句相当于一个简单视图，执行并限制只能修改SELECT语句指定的列，前提确保子查询的的结果为一个值。

165. View the Exhibit and  examine the description for the CUSTOMERS table.

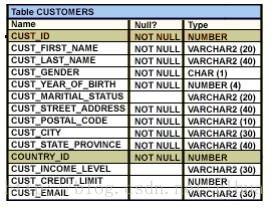
You want to update the CUST\_INCOME\_LEVEL and CUST\_CREDIT\_LIMIT columns for the customer

with the CUST\_ID 2360. You want the value for the CUST\_INCOME\_LEVEL to have the same value as

that of the customer with the CUST\_ID 2560 and the CUST\_CREDIT\_LIMIT to have the same value as

that of the customer with CUST\_ID 2566.

Which UPDATE statement will accomplish the task?



A. UPDATE customers

SET cust\_income\_level = (SELECT cust\_income\_level

FROM customers

WHERE cust\_id = 2560),

cust\_credit\_limit = (SELECT cust\_credit\_limit

FROM customers

WHERE cust\_id = 2566)

WHERE cust\_id=2360;

B. UPDATE customers

SET (cust\_income\_level,cust\_credit\_limit) = (SELECT

cust\_income\_level, cust\_credit\_limit

FROM customers

WHERE cust\_id=2560 OR cust\_id=2566)

WHERE cust\_id=2360;

C. UPDATE customers

SET (cust\_income\_level,cust\_credit\_limit) = (SELECT

cust\_income\_level, cust\_credit\_limit

FROM customers

WHERE cust\_id IN(2560, 2566)

WHERE cust\_id=2360;

D. UPDATE customers

SET (cust\_income\_level,cust\_credit\_limit) = (SELECT

cust\_income\_level, cust\_credit\_limit

FROM customers

WHERE cust\_id=2560 AND cust\_id=2566)

WHERE cust\_id=2360;

Answer: A

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

BC的子查询返回的是多行，所有报错。

D的子查询条件不正确，

题意说：更新CUST\_ID为2360的CUST\_INCOME\_LEVEL和CUST\_CREDIT\_LIMIT列值。你想让CUST\_INCOME\_LEVEL的值与CUST\_ID为2560的值一样，让CUST\_CREDIT\_LIMIT的值与CUST\_ID为2566的值一样

按题意，只有A正确。

实验验证：

A

*UPDATE customers*

*2  SET cust\_income\_level = (SELECT cust\_income\_level*

*FROM customers*

*WHERE cust\_id = 2560),*

*cust\_credit\_limit = (SELECT cust\_credit\_limit*

*FROM customers*

*WHERE cust\_id = 2566)*

*8  WHERE cust\_id=2360;*

*1 row updated.*

B

*sh@TESTDB> UPDATE customers*

*2  SET (cust\_income\_level,cust\_credit\_limit) = (SELECT*

*cust\_income\_level, cust\_credit\_limit*

*FROM customers*

*WHERE cust\_id=2560 OR cust\_id=2566)*

*6  WHERE cust\_id=2360;*

*SET (cust\_income\_level,cust\_credit\_limit) = (SELECT*

*\**

*ERROR at line 2:*

*ORA-01427: single-row subquery returns more than one row*

C

*sh@TESTDB> UPDATE customers*

*2  SET (cust\_income\_level,cust\_credit\_limit) = (SELECT*

*cust\_income\_level, cust\_credit\_limit*

*FROM customers*

*WHERE cust\_id IN(2560, 2566)*

*6  WHERE cust\_id=2360;*

*WHERE cust\_id=2360*

*\**

*ERROR at line 6:*

*ORA-00907: missing right parenthesis*

缺失右括号

添加上 右括号也报错误 single-row subquery returns more than one row

*sh@TESTDB>  UPDATE customers*

*2  SET (cust\_income\_level,cust\_credit\_limit) =*

*3  (SELECT*

*4  cust\_income\_level, cust\_credit\_limit FROM customers*

*5  WHERE cust\_id IN(2560, 2566))*

*6   WHERE cust\_id=2360;*

*(SELECT*

*\**

*ERROR at line 3:*

*ORA-01427: single-row subquery returns more than one row*

D

*sh@TEST0924> UPDATE customers*

*2  SET (cust\_income\_level,cust\_credit\_limit) = (SELECT*

*3  cust\_income\_level, cust\_credit\_limit*

*4  FROM customers*

*5  WHERE cust\_id=2560 AND cust\_id=2566)*

*6  WHERE cust\_id=2360;*

*1 row updated.*

166. View the Exhibit and examine the structures of the EMPLOYEES and DEPARTMENTS tables.

You want to update the EMPLOYEES table as follows:4 ? 4;

-Update only those employees who work in Boston or Seattle (locations 2900 and 2700).

-Set department\_id for these employees to the department\_id corresponding to London (location\_id

2100).

-Set the employees' salary in location\_id 2100 to 1.1 times the average salary of their department.

-Set the employees' commission in location\_id 2100 to 1.5 times the average commission of their

department.

You issue the following command:

SQL>UPDATE employees

SET department\_id =

(SELECT department\_id

FROM departments

WHERE location\_id = 2100),

(salary, commission) =

(SELECT 1.1\*AVG(salary), 1.5\*AVG(commission)

FROM employees, departments

WHERE departments.location\_id IN(2900,2700,2100))

WHERE department\_id IN

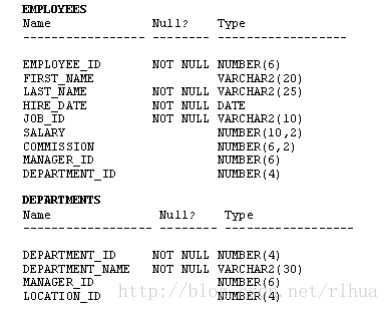
(SELECT department\_id

FROM departments

WHERE location\_id = 2900

OR location\_id = 2700)

What is the outcome?



A. It executes successfully and gives the correct result.

B. It executes successfully but does not give the correct result.

C. It generates an error because a subquery cannot have a join condition in an UPDATE statement.

D. It generates an error because multiple columns (SALARY, COMMISION) cannot be specified together

in an UPDATE statement.

Answer: B

 答案解析：

题意要求更新条件：

1.只更新那些在Boston或Seattle工作的employees（locations为2900或者2700）

2.设置这些employees的department\_id为London（location\_id 2100）对应的department\_id

3.设置location\_id 2100的employees' salary为他们部门的平均薪水的1.1倍

4.设置location\_id 2100的employees' commission为他们部门的平均提成的1.5倍

UPDATE employees

SET department\_id =

(SELECT department\_id FROM departments WHERE location\_id = 2100),

(salary, commission) = (SELECT 1.1\*AVG(salary), 1.5\*AVG(commission)  FROM employees, departments

WHERE departments.location\_id IN(2900,2700,2100))

WHERE department\_id IN (SELECT department\_id FROM departments WHERE location\_id = 2900 OR location\_id = 2700)

WHERE departments.location\_id IN(2900,2700,2100))        应该使用部门ID进行关联。

167. Evaluate the following DELETE statement:

DELETE FROM sales;

There are no other uncommitted transactions on the SALES table.

Which statement is true about the DELETE statement?

A. It would not remove the rows if the table has a primary key.

B. It removes all the rows as well as the structure of the table.

C. It removes all the rows in the table and deleted rows can be rolled back.

D. It removes all the rows in the table and deleted rows cannot be rolled back.

Answer: C

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

A，如果表中有主键则不能移除行，错误，有主键是可以删除的

B，移除表中所有行及表结构，错误，表结构是保留的

C，移除表中所有行，并且删除的行可以回滚，正确，删除的数据保留在undo段，可以回滚。

D，移除表中所有行，并且删除的行不可以回滚，错误，删除的行是可以回滚的。

168. View the Exhibit and examine the description of SALES and PROMOTIONS tables.

You want to delete rows from the SALES table, where the PROMO\_NAME column in the PROMOTIONS

table has either blowout sale or everyday low price as values.

Which DELETE statements are valid? (Choose all that apply.)

delete from sales

where promo\_id=(select promo\_id from promotions where)

A. DELETE FROM sales

WHERE promo\_id = (SELECT promo\_id

FROM promotions

WHERE promo\_name = 'blowout sale')

AND promo\_id = (SELECT promo\_id

FROM promotions

WHERE promo\_name = 'everyday low price');

B. DELETE FROM sales

WHERE promo\_id = (SELECT promo\_id

FROM promotions

WHERE promo\_name = 'blowout sale')

OR promo\_id = (SELECT promo\_id

FROM promotions

WHERE promo\_name = 'everyday low price');

C. DELETE FROM sales

WHERE promo\_id IN (SELECT promo\_id

FROM promotions

WHERE promo\_name = 'blowout sale'

OR promo\_name = 'everyday low price');

D. DELETE FROM sales

WHERE promo\_id IN (SELECT promo\_id

FROM promotions

WHERE promo\_name IN ('blowout sale','everyday low price'));

Answer: BCD

答案解析：

题意要求：从SALES表中删除行，条件为PROMOTIONS表中PROMO\_NAM列的值为blowout sale或者everyday low price）

BCD这三个选项是等价的。

而A的where条件中的AND不满足题意要求。

169. View the Exhibit and examine the description for the PRODUCTS and SALES table.

PROD\_ID is a primary key in the PRODUCTS table and foreign key in the SALES table. You want to

remove all the rows from the PRODUCTS table for which no sale was done for the last three years.

Which is the valid DELETE statement?

A. DELETE

FROM products

WHERE prod\_id = (SELECT prod\_id

FROM sales

WHERE time\_id - 3\*365 = SYSDATE );

B. DELETE

FROM products

WHERE prod\_id = (SELECT prod\_id

FROM sales

WHERE SYSDATE >= time\_id - 3\*365 );

C. DELETE

FROM products

WHERE prod\_id IN (SELECT prod\_id

FROM sales

WHERE SYSDATE - 3\*365 >= time\_id);

D. DELETE

FROM products

WHERE prod\_id IN (SELECT prod\_id

FROM sales

WHERE time\_id >= SYSDATE - 3\*365 );

Answer: C

 答案解析：

题意要求：从PRODUCTS表移除所有过去三年没有销售过的产品行

SYSDATE - 3\*365 <= time\_id三年前之前的时间，三年前多，

SYSDATE - 3\*365 >= time\_id三年前之后的时间，三年前往后面数，即应该用这个

即选C

170. Which two statements are true regarding the DELETE and TRUNCATE commands? (Choose two.)

A. DELETE can be used to remove only rows from only one table at a time.

B. DELETE can be used to remove only rows from multiple tables at a time.

C. DELETE can be used only on a table that is a parent of a referential integrity constraint. 父级的参照完整性约束错？

D. DELETE can be used to remove data from specific columns as well as complete rows. 指定行删除

E. DELETE and TRUNCATE can be used on a table that is a parent of a referential integrity constraint

having ON DELETE rule .

Answer: AE

答案解析：

A，DELETE一次只能用于移除一个表的行，正确

B，DELETE一次只能用于移除多个表的行，错误，一次只能一个表

C，DELETE只能用于删除有引用完整性约束的父表，错误，有完整性约束的父表不能被删除，除非禁用约束等

D，DELETE可以用于删除指定列的数据以及完整的行，错误，只能删除行，不能删除列

E，DELETE和TRUNCATE可以用于有引用完整性约束ON DELETE规则的父表。正确。

建表时可以使用ON DELETE CASCADE（当删除父表数据时，子表数据也一起删除）或ON DELETE CASCAD SET NULL（当删除父表数据时，子表相关的列设置为NULL）子句，可以在SQL Language Reference里的Constraints部分查找到ON DELETE子句

171. Which three statements/commands would cause a transaction to end? (Choose three.)

A. COMMIT

B. SELECT

C. CREATE

D. ROLLBACK

E. SAVEPOINTAnswer: ACD

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

事务处理在遇到第一条DML 语句时开始，在发生以下事件之一时结束：

• 发出COMMIT或ROLLBACK语句。

• 发出DDL 语句，例如CREATE。

• 发出DCL 语句。

• 用户退出SQL Developer 或SQL\*Plus。

• 计算机出现故障或系统崩溃。

一个事务处理结束后，下一个可执行的SQL 语句会自动启动下一个事务处理。DDL 语句或DCL 语句是自动提交的，因此会隐式结束一个事务处理

172. The SQL statements executed in a user session are as follows:

SQL> CREATE TABLE product

(pcode  NUMBER(2),

pname  VARCHAR2(10));

SQL> INSERT INTO product  VALUES (1, 'pen');

SQL> INSERT INTO product  VALUES (2,'pencil');

SQL> SAVEPOINT a;

SQL> UPDATE product SET pcode = 10 WHERE pcode = 1;

SQL> SAVEPOINT b;

SQL> DELETE FROM product  WHERE pcode = 2;

SQL> COMMIT;

SQL> DELETE FROM product WHERE pcode=10;

Which two statements describe the consequences of issuing the ROLLBACK TO SAVE POINT a

command in the session? (Choose two.)

A. The rollback generates an error.

B. No SQL statements are rolled back.

C. Only the DELETE statements are rolled back.

D. Only the second DELETE statement is rolled back.

E. Both the DELETE statements and the UPDATE statement are rolled back.

Answer: AB

 答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

由于COMMIT命令提交后之前建立的保存点都变成无效的了。

ROLLBACK TO SAVEPOINT a;命令报错后，只会影响本身这条语句，不会影响其它语句，所以第二个DELETE语句不会回滚。

实验验证：

sh@TEST0910> create table product

  2  (pcode number(2),

  3  pname varchar2(10));

Table created.

sh@TEST0910> INSERT INTO product  VALUES (1, 'pen');

1 row created.

sh@TEST0910> INSERT INTO product  VALUES (2,'pencil');

1 row created.

sh@TEST0910>  SAVEPOINT a;

Savepoint created.

sh@TEST0910> UPDATE product SET pcode = 10 WHERE pcode = 1;

1 row updated.

sh@TEST0910>  SAVEPOINT b;

Savepoint created.

sh@TEST0910> DELETE FROM product  WHERE pcode = 2;

1 row deleted.

sh@TEST0910>  COMMIT;

Commit complete.

sh@TEST0910> select \* from product;

     PCODE PNAME

---------- ----------

        10 pen

sh@TEST0910>  DELETE FROM product WHERE pcode=10;

1 row deleted.

sh@TEST0910> ROLLBACK TO SAVE POINT a

  2  ;

ROLLBACK TO SAVE POINT a

                 \*

ERROR at line 1:

ORA-00933: SQL command not properly ended

173. When does a transaction complete? (Choose all that apply.)

A. when a DELETE statement is executed

B. when a ROLLBACK command is executed

C. when a  PL/SQL anonymous block is executed

D. when a data definition language ( DDL)  statement is executed

E. when a TRUNCATE statement is executed after the pending transaction

Answer: BDE

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

事务处理在遇到第一条DML 语句时开始，在发生以下事件之一时结束：

• 发出COMMIT或ROLLBACK语句。

• 发出DDL 语句，例如CREATE,TRUNCATE等。

• 发出DCL 语句。

• 用户退出SQL Developer 或SQL\*Plus。

• 计算机出现故障或系统崩溃。

一个事务处理结束后，下一个可执行的SQL 语句会自动启动下一个事务处理。DDL 语句或DCL 语句是自动提交的，因此会隐式结束一个事务处理。

174. Which statement is true regarding transactions? (Choose all that apply.)

A. A transaction can consist only of a set of DML and DDL statements.

B. A p art or an entire   transaction can be undone by using  ROLLBACK command .

C. A transaction consists of a set of DML or DCL statements.

D. A part or an entire transaction can be made permanent with a COMMIT.

E. A transaction can consist of only a set of queries or DML or DDL statements.

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

A，一个事务只能由一组DML和DDL语句组成，

错误，事务由第一条DML开始，由事件之一时结束：

• 发出COMMIT或ROLLBACK语句。

• 发出DDL 语句，例如CREATE,TRUNCATE等。

• 发出DCL 语句。

• 用户退出SQL Developer 或SQL\*Plus。

• 计算机出现故障或系统崩溃。

一个事务处理结束后，下一个可执行的SQL 语句会自动启动下一个事务处理。DDL 语句或DCL 语句是自动提交的，因此会隐式结束一个事务处理。

B，部分或整个事务可以使用ROLLBACK命令回滚，正确，可以用rollback来回滚部分或者全部事务，回滚部分需要保存点。

C，一个事务由一组DML或DCL语句组成，正确。

D，部分或整个事务可以使用COMMIT命令永久提交，错误。因为COMMIT把整个事务都进行提交，不能提交部分事务。

E，一个事务只能由一组查询或DML或DDL语句组成，错误。查询不是事务。

175. Which two statements are true regarding savepoints? (Choose two.)

A. Savepoints are effective only for COMMIT.

B. Savepoints may be used to ROLLBACK.

C. Savepoints can be used for only DML statements.

D. Savepoints are effective for both COMMIT and ROLLBACK.

E. Savepoints can be used for both DML and DDL statements.

Answer: BC

答案解析：

参考：<http://blog.csdn.net/rlhua/article/details/12885143>

A，保存点只对COMMIT有效，错误，只对ROLLBACK

B，保存点可以用于ROLLBACK，正确。

C，保存点只能用于DML语句，正确。

D，保存点对于COMMIT和ROLLBACK都有效，错误。只对ROLLBACK。

E，保存点可以用于DML和DDL语句，错误。DDL语句不行，DDL相当于commit，commit后不能回滚。

保存点只能用于DML语句，并且只用于ROLLBACK。