Databse test



**Câu 1**: Giải thích các hàm thông dụng sau

|  |  |  |
| --- | --- | --- |
| Stt | Tên Hàm | Mục đích sử dụng & nên sử dụng khi nào |
| 1 | Count() | Sử dụng khi đếm số lượng dữ liệu, hàng. Nên sử dụng khi đếm số lượng bản ghi |
| 2 | Sum() | Sử dụng khi trả về giá trị tổng của một cột, một tập dữ liệu hoặc một biểu thức. Nên sử dụng khi cần tính tổng. |
| 3 | MAX() | Sử dụng để trả về giá trị tối đa của một nhóm. Sử dụng tìm giá trị lớn nhất của dữ liệu đó. |
| 4 | MIN() | Sử dụng khi trả về giá trị nhỏ nhất của một nhóm. Sử dụng tìm giá trị nhỏ nhất của dữ liệu đó. |
| 5 | NVL() | Trả về một giá trị thay thế giá trị null. Nếu giá trị không phải null thì trả về chính giá trị(biểu thức) đó. Sử dụng cho thay thế giá trị null. |
| 6 | TO\_CHAR() | Là hàm chuyển đôi dữ liệu dạng số nguyên hoặc ngày tháng thành chuỗi String. Dùng để biển đổi dữ liệu về String. |
| 7 | T0\_DATE() | Là hàm chuyển đổi dữ liệu dạng số hoặc dạng chuỗi thành ngày tháng. Dùng để biến đổi dữ liệu về ngày tháng. |
| 8 | TO\_NUMBER() | Là hàm dung để chuyển đổi dữ liệu chuỗi hoặc dạng ngày tháng về dạng số. Dùng để biến đổi dữ liệu về số. |
| 9 | SUBSTR() | Là hàm dung để trích xuất một chuỗi có độ dài chỉ định từ vị trí truyền vào. Dùng để lấy một chuỗi trích xuất. |
| 10 | REPLACE() | Là hàm dung để thay thế tất cả các lần xuất hiện của chuỗi con thành chuỗi con mới xuất hiện trong 1 chuỗi. |
| 11 | REVERSE() | Dùng để đảo ngược các kí tự trong một biểu thức. Nếu là null thì trả về null. |
| 12 | DECODE() | Dùng để thực hiện cấu trúc rẽ nhánh(if else và case when) |
| 13 | TRUNC() | Dùng để thu gọn một số đến vị trí nào đó sau dấu, vị trí được truyền vào bởi tham số truyền vào. |
| 14 | LENGTH() | Dùng để trả về độ dài của một chuỗi. Sử dụng khi ta cần biết độ dài của một chuỗi. |
| 15 | lPAD() | Là hàm đệm phía bên trái của một chuỗi cụ thể. (value chính, độ dài, value truyền vào). |
| 16 | RPAD() | Là hàm đệm phía bên phải của một chuỗi cụ thể. (value chính, độ dài, value truyền vào). |
| 17 | TRIM() | Sử dụng để xóa khoảng trắng đầu, đuôi hoặc cả 2 của một chuỗi. |
| 18 | LTRIM() | Sử dụng xóa bỏ khoảng trắng các vị trí bên trái của một chuỗi. |
| 19 | RTRIM() | Sử dụng xóa bỏ khoảng trắng các vị trí bên phải của một chuỗi. |
| 20 | ROUND() | Sử dụng để làm tròn số đến một vị trí chỉ định. |
| 21 | ADD\_MONTHS() | Được sử dụng để tang hoặc giảm tháng trong date value. Ví dụ 1 thì cộng thêm 1 tháng. Còn -1 thì trừ đi 1 tháng |

**Câu 2**:

SELECT \*

FROM MDM\_CUSTOMER A

WHERE A.CUST\_LGL\_ENG\_NM LIKE ‘%\\_LOGISTICS%’ ESCAPE ‘\’

ORDER BY A.CUST\_LOCL\_LANG\_NM NULLS FIRST

A) Vui lòng giải thích ý nghĩa của câu SQL trên

B) ý nghĩa của việc dùng ESCAPSE

C) Ý nghĩa của việc dung Nulls First.

D) Ý nghĩa của việc dung alias, có nên dung alias trong mọi trường hợp không?

**Trả lời:**

A):

Lấy tất cả dữ liệu từ bảng MDM\_CUSTOMER với điều kiện CUST\_LGL\_ENG\_NM là chuỗi có chứa “\_LOGISTICS” và sắp xếp theo thứ tự tăng dần và record có giá trị CUST\_LOCL\_LANG\_NM = null thì được đưa sắp xếp lên đầu tiên.

B):

ESCAPSE: Thể hiện cái kí tự sau kí tự chỉ định là một kí tự thông thường ví dụ sau ‘\’ là một dấu ‘\_’ nếu không có ESCAPSE thì nó sẽ hiểu dấu đó là 1 kí tự bất kì.

C): Nulls First nó sắp xếp các record có dữ liệu của các thuộc tính được chọn trong order by là null thì sẽ được sắp xếp lên đầu.

D): Alias là đặt tên. Không vì nó chỉ lấy dữ liệu từ 1 bảng.

**Câu 3**:

SELECT \*

FROM MDM\_CUSTOMER

WHERE 1 = 1

AND CUST\_CNT\_CD = 'JP'

AND TO\_CHAR(CUST\_SEQ) = '201791'

Theo bạn câu trên cách dung TO\_CHAR(CUST\_SEQ) = '201791' có hợp lý không, tại sao?

**Trả lời:**

Không hợp lý. Tại vì TO\_CHAR chuyển dữ liệu thành 1 chuỗi mà ‘201791’ đã là chuỗi thì không cần thêm TO\_CHAR.

**Câu 4**: cho câu SQL và kết quả như hình bên dưới



A) Giải thích ý nghĩa COUNT(\*), COUNT(1), COUNT(CUST\_LOCL\_LANG\_NM)

B) Tại sao COUNT(CUST\_LOCL\_LANG\_NM) lại bằng 0

**Trả lời:**

1. : COUNT(\*) và COUNT(1) là giống nhau đều đếm số lượng các record. Còn COUNT(CUST\_LOCL\_LANG\_NM) cũng giống như vậy nhưng nó không đếm record có giá trị null.
2. : Tại vì giá trị mà hàm COUNT (CUST\_LOCL\_LANG\_NM) có giá trị null.

**Câu 5**: có 2 cách như bên dưới, cách nào tốt tại sao

|  |  |  |
| --- | --- | --- |
|  | **Cách 1** | **Cách 2** |
|  | NVL(SUM(COL1),0) | SUM(NVL(COL1,0)) |

**Trả lời:**

Cách 1 tốt hơn. Vì cách 1 SUM tính tổng COL1 và bỏ qua các giá trị null. Sau đó NVL tìm giá trị NULL và đổi về 0 nếu có thì chỉ NVL 1 lần. Còn cách 2 NVL đang đổi NULL = 0 nhiều lần rồi mới SUM.

**Câu 6**: có 2 cách như bên dưới, cách nào tốt tại sao

|  |  |  |
| --- | --- | --- |
|  | **Cách 1** | **Cách 2** |
|  | NVL(SUM(COL1),0) + NVL(SUM(COL2),0) | Ex.1] SUM(NVL(COL1 + COL2,0))  Ex.2] NVL(SUM(COL1 + COL2),0) |

**Trả lời:**

Cách 2 nhanh hơn vì ta chỉ duyệt một lần khi SUM. Còn cách 1 Duyệt 2 lần rồi mới SUM.

**Câu 7**: có 2 cách như bên dưới, cách nào tốt tại sao

|  |  |  |
| --- | --- | --- |
|  | **Cách 1** | **Cách 2** |
|  | SELECT A.CUST\_NO, A.ORD\_NO, A.PRO\_CD, B.PROD\_NM  FROM TB\_ORD A,  TB\_PROD B  WHERE 1 = 1  AND A.PRO\_CD = B.PROD\_CD  AND B.PROD\_CD IN (SELECT PROD\_CD FROM TB\_PROD D WHERE D.PROD\_CD = A.PRO\_CD AND PROD\_UNIT\_AMT < 800); | SELECT A.CUST\_NO, A.ORD\_NO, A.PRO\_CD, B.PROD\_NM  FROM TB\_ORD A,  TB\_PROD B  WHERE 1 = 1  AND A.PRO\_CD = B.PROD\_CD  AND EXISTS (SELECT D.PROD\_CD FROM TB\_PROD D WHERE D.PROD\_CD = A.PRO\_CD AND D.PROD\_UNIT\_AMT < 800); |

**IN** đc dung khi nào và EXISTS đc dung khi nào.

**Trả lời:**

In() được sử dụng như một toán tử multiple OR hoặc truy vấn subquery phụ single. Còn EXISTS() được sử dụng để xác định xem có giá trị nào được trả về hay không (ít nhất là 1 hàng) hoặc truy vấn subquery phụ lớn. Nếu truy vấn phụ lớn thì EXISTS nhanh hơn IN. Nhưng trong tường hợp này sẽ sử dụng IN() vì nó chỉ là single subquery.

**Câu 8**: có 2 cách như bên dưới, cách nào tốt tại sao

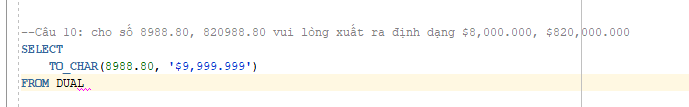
|  |  |  |
| --- | --- | --- |
|  | **Cách 1** | **Cách 2** |
|  | SELECT A.CUST\_NO, A.ORD\_NO, A.PRO\_CD, B.PROD\_NM  FROM TB\_ORD A,  TB\_PROD B  WHERE 1 = 1  AND A.PRO\_CD = B.PROD\_CD  AND B.PROD\_CD IN ('00001','00002'); | SELECT A.CUST\_NO, A.ORD\_NO, A.PRO\_CD  , (SELECT B.PROD\_NM FROM TB\_PROD B WHERE B.PROD\_CD = A.PRO\_CD) AS PROD\_NM  FROM TB\_ORD A  WHERE 1 = 1  AND A.PRO\_CD IN ('00001','00002'); |

**Trả lời:**

Cách 2 sẽ tốt hơn do truy vấn phụ thứ 2 đã giúp ta loại bỏ được các phần tử không cần bởi where và lúc truy vấn thứ 2 sẽ giảm bớt chi phí gọi các hàm đã được lược bỏ. Dữ liệu lớn mà sử dụng cách 1 sẽ rất nặng do đều gọi hết 2 bảng rồi mới lược bỏ bởi điều kiện where.

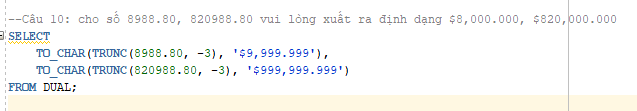
**Câu 9**: cho số 8988.80 vui lòng xuất ra định dạng $8,988.800

**Trả lời:**

****

**Câu 10**: cho số 8988.80, 820988.80 vui lòng xuất ra định dạng $8,000.000, $820,000.000

**Trả lời:**



**Câu 11**: Cho cấu SQL và kết quả như sau:



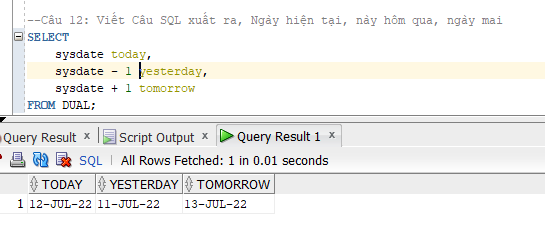
Như hình trên cả 2 A và B điêu substr từ 1, đến 3 tại sao kết quả lại khác nhau.

**Trả lời:**

FM ở đây nó đã thể hiện cho việc loại bỏ 1 kí tự thể hiện dấu + or - ở trước số đó nên khi substr ở chỗ có FM sẽ có độ dài nhỏ hơn substr không có FM. Nên khi substr trả về dữ liệu khác nhau.

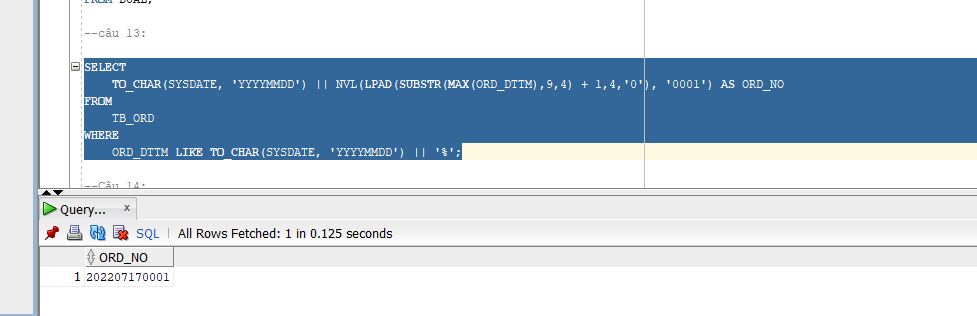
**Câu 12**: Viết Câu SQL xuất ra, Ngày hiện tại, này hôm qua, ngày mai

**Trả lời**:



**Câu 13**: ta có table (**TB\_ORD**), yêu cầu viết câu SQL để generate ORD\_NO có đô dài 10 tự với format sau: yyyymmdd000Seq, ví dụ hnay là 20191028 và chưa có seq nào thì ORD\_NO sẽ là 201910280001, và nếu đã tồn tại ORD\_NO 201910280001 thì nó sẽ là 201910280002

**Trả lời:**



**Câu 14**: ta có table (**MDM\_CUSTOMER**) và dữ liệu như bên dưới

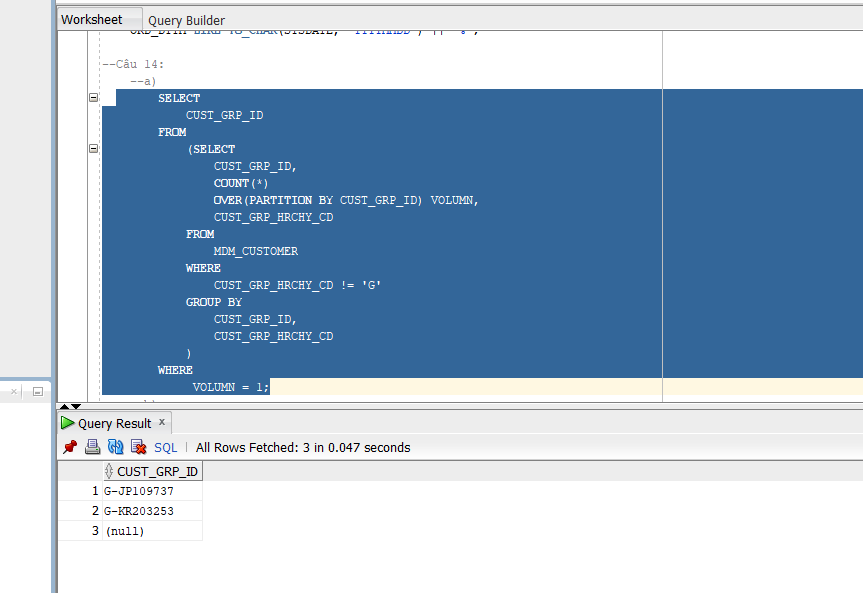


Các field liên quan: CUST\_CNT\_CD, CUST\_SEQ, CUST\_GRP\_HRCHY\_CD, CUST\_GRP\_ID

Dữ liệu cột CUST\_GRP\_HRCHY\_CD có thể có(**I**: individual, **C**: Country, **G**: Global)

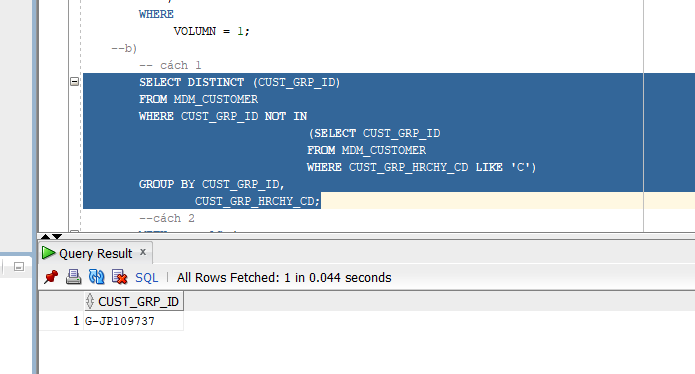
1. Viết câu SQL tìm CUST\_GRP\_ID sao cho: CUST\_GRP\_HRCHY\_CD có **I** hoặc **C** nhưng không có **G**

**Trả lời:**

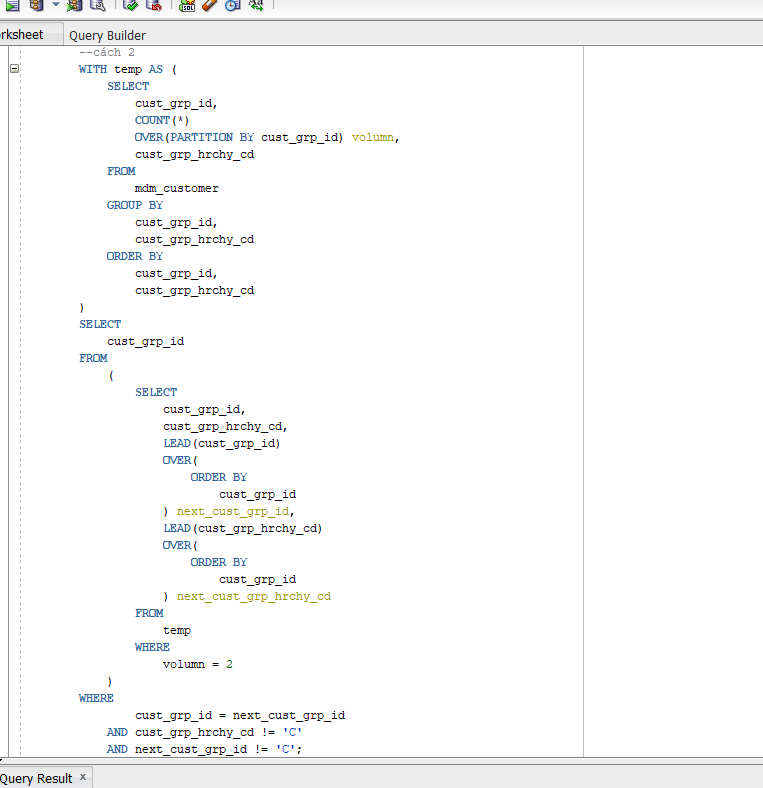
****

B) Viết câu SQL tìm CUST\_GRP\_ID sao cho: CUST\_GRP\_HRCHY\_CD có **G** và có **I** nhưng không có **C**

**Trả lời:**

****

**Cách 2:**

****

**Câu 15**: ta có table (**TB\_PROD**) và dữ liệu như bên dưới



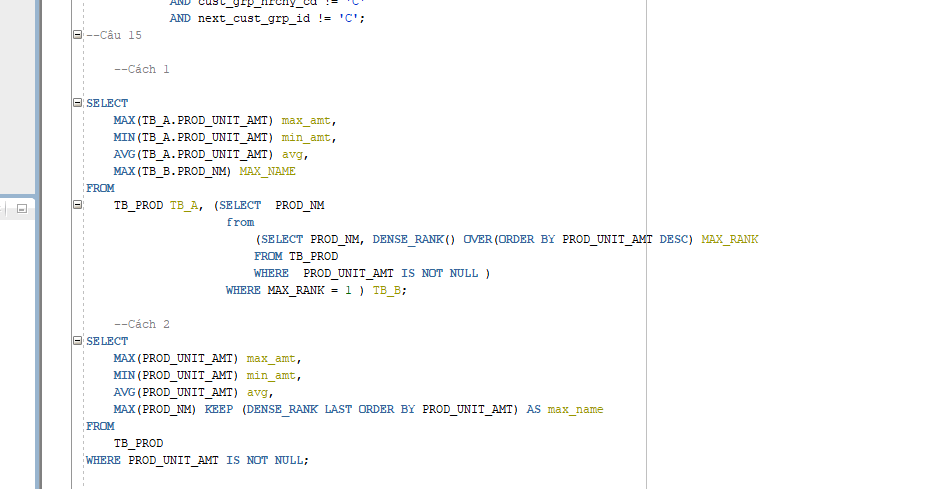
Viets cấu SQL để suất ra kêt quả như sau:

1. Lấy max(PROD\_UNIT\_AMT)
2. Lấy giá trị min(PROD\_UNIT\_AMT)
3. Lấy giá trị trung bình PROD\_UNIT\_AMT
4. Lấy tên của sản phẩm có PROD\_UNIT\_AMT lớn nhất

Kết quả phải ra đc như sau:



**Trả lời:**

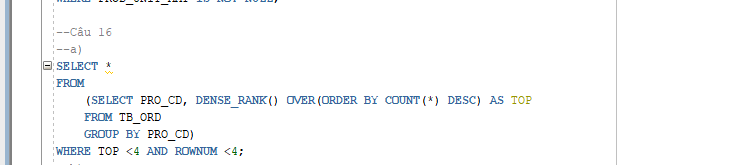
****

**Câu 16**: ta có table (**TB\_ORD**) và dữ liệu như bên dưới



A) viết cấu SQL lấy ra top3 sản phẩm đc bán nhiều nhất**.**

**Trả lời:**

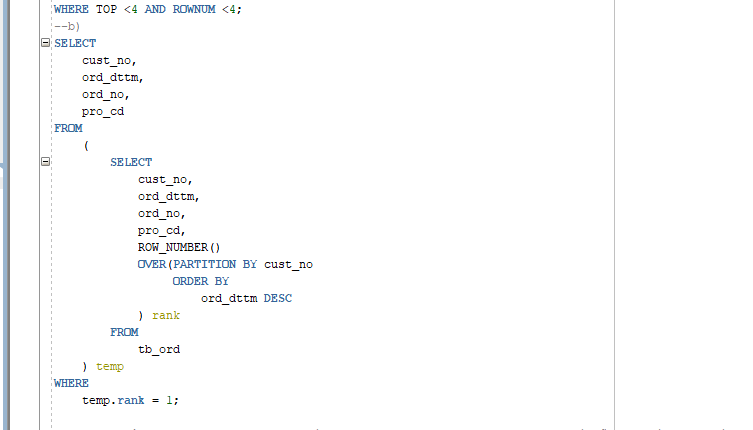
****

B) Viết cấu SQL lấy ra cái ORD\_DT, ORD\_TM, PROD\_CD gần nhất theo CUST\_NO **ROWNUM**

Kết quả mong đợi như sau:



**Trả lời:**

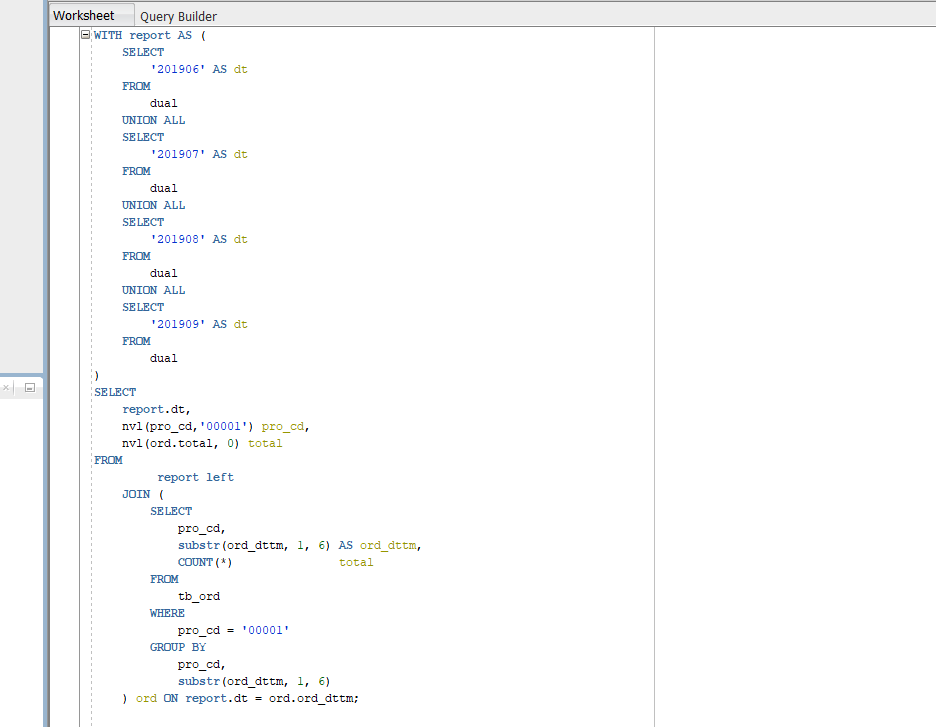
****

C) viết cấu SQL report xem trong tháng 06, 07, 08, 09 cảu 2019 sản phẩm có mã code là 00001bán đc bao nhiêu cái.

Kết quả mong đợi nhưu sau: left outer join partition - 130



**Trả lời:**

****

D) giả sư lúc đầu sản phẩn 00001 có 100 cái, viết report để tính số lương remain theo tháng 06, 07, 08, 09



**Trả lời:**

