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() | Đếm số lượng record |
| 2 | Sum() | Tính tổng của column chỉ định |
| 3 | MAX() | Return giá trị max của column được chỉ định |
| 4 | MIN() | Return giá trị min của column được chỉ định |
| 5 | NVL() | Thay thế giá trị null bằng một giá trị chỉ định |
| 6 | TO\_CHAR() | Chuyển đổi giá trị sang kiểu chuỗi với format cho trước |
| 7 | T0\_DATE() | Chuyển đổi giá trị sang kiểu date với format cho trước |
| 8 | TO\_NUMBER() | Chuyển đổi giá trị sang kiểu kiểu số |
| 9 | SUBSTR() | Cắt một chuỗi con từ một chuỗi cho trước |
| 10 | REPLACE() | Thay thế chuỗi con thanh một chuỗi con khác trong một chuỗi lớn |
| 11 | REVERSE() | Đảo ngược kí tự trong một chuỗi |
| 12 | DECODE() | Sử dụng giống như case when trong java |
| 13 | TRUNC() | Dùng để rút gọn số đến một vị trí chỉ định sau dấu “,” |
| 14 | LENGTH() | Return độ dài của chuỗi chỉ định |
| 15 | lPAD() | Chèn vào bên trái của chuỗi với một tập giá trị cho trước |
| 16 | RPAD() | Chèn vào bên phải của chuỗi với một tập giá trị cho trước |
| 17 | TRIM() | Xóa khoảng trắng 2 bên của chuỗi |
| 18 | LTRIM() | Xóa khoảng trắng bên trái của chuỗi |
| 19 | RTRIM() | Xóa khoảng trắng bên phải của chuỗi |
| 20 | ROUND() | Làm tròn số thập phân |
| 21 | ADD\_MONTHS() | Thêm vào số tháng chỉ định, có thể giảm khi set giá trị âm |

**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

1. Vui lòng giải thích ý nghĩa của câu SQL trên

-> Get toàn bộ column của table MDM\_CUSTOMER có CUST\_LGL\_ENG\_NM giống như

B) ý nghĩa của việc dùng ESCAPSE

-> Báo hiệu sau dấu ‘\’ thì các kí tự đều trở thành kí tự bình thường

C) Ý nghĩa của việc dung Nulls First.

-> Sắp xếp giá trị Null lên đầu

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

-> alias dung để định danh thay thế column hoặc table, giúp cho việc gọi lại đối tượng dễ dàng hơn. Trong một số trường hợp có thể dùng để định danh thay thế cho một câu select lớn

**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?

* Không cần thiết vì oracle có thể tự hiểu để convert giá trị '201791' về kiểu dữ liệu của column so sánh

**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)

-> COUNT(\*), COUNT(1) có cùng ý nghĩa là đếm số lượng record trong bảng, bao gồm cả giá trị NULL

-> COUNT(CUST\_LOCL\_LANG\_NM) có ý nghĩa là đếm số lượng record của cột CUST\_LOCL\_LANG\_NM nhưng không bao gồm giá trị NULL

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

-> Bơi vì column CUST\_LOCL\_LANG\_NM đang có giá trị là 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)) |

* Cách 1 tốt hơn vì sau khi sum toàn bộ col1 thì mới kiểm tra null để xem có return về 0 hay không, trong khi đó cách 2 thì sẽ kiểm tra null của từng record để trả về 0 sau đó 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) |

* Cách 1 tốt hơn vì khi để 2 column cộng giá trị trực tiếp sẽ có thể bị trường hợp value + null = null, điều này sẽ dẫn đến kết quả sai dự kiến

**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.

* Trường hợp có thể xác định số record được select trong IN là số cố định và nhỏ thì có thể sử dụng IN, trong trường hợp số record được select trong IN quá lớn hoặc không thể xác định thì nên dung exists

**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'); |

* Cách 1 tốt hơn thì đối với cách 2 sau khi select được record của TB\_ORD thì với từng record sẽ đối chiếu với TB\_PROD để get name, việc này sẽ lâu hơn so với việc kết 2 bảng từ trước

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

* SELECT TO\_CHAR (8988.80,'L9G999G999D000')

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

* SELECT TO\_CHAR(TRUNC(8988.80,-3), 'L9G999G999D000') A, TO\_CHAR(TRUNC(820988.80,-3), 'L9G999G999D000')

**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.

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

* SELECT SYSDATE - 1 YESTEDAY, SYSDATE TODAY, SYSDATE + 1 TOMORROW FROM DUAL

**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

* SELECT NVL(MAX(ORD\_NO)+1, TO\_CHAR(SYSDATE, 'YYYYMMDD')|| '0001') AS ORD\_NO

FROM TB\_ORD

WHERE ORD\_NO LIKE TO\_CHAR(SYSDATE,'YYYYMMDD') || '%';

**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)

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

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

**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

* SELECT

MAX(PROD\_UNIT\_AMT) AS MAX \_AMT,

MIN(PROD\_UNIT\_AMT) AS MIN\_ AMT,

AVG(PROD\_UNIT\_AMT) AS AVG\_ AMT,

MAX(PROD\_NM) KEEP (DENSE\_RANK LAST ORDER BY PROD\_UNIT\_AMT NULLS FIRST) AS MAX\_ NAME

FROM TB\_PROD

Kết quả phải ra đc như sau:



**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**.**

SELECT

PRO\_CD

FROM

(SELECT

PRO\_CD,

COUNT(PRO\_CD) AS CNT

FROM TB\_ORD

GROUP BY PRO\_CD

ORDER BY CNT DESC)

WHERE ROWNUM <= 5

1. Viết cấu SQL lấy ra cái ORD\_DT, ORD\_TM, PROD\_CD gần nhất theo CUST\_NO

* SELECT CUST\_NO,ORD\_DTTM, PRO\_CD

FROM (

SELECT CUST\_NO,ORD\_DTTM, PRO\_CD, ROW\_NUMBER () OVER (PARTITION BY CUST\_NO ORDER BY ORD\_DTTM DESC) TOP

FROM TB\_ORD

) A

WHERE A.TOP = 1;

Kết quả mong đợi như sau:



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



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

