

Name: Nguyen Trung Hieu

Database test

New / Select Database Connection

Connection Name	Connection Details
NYK Test	opusdev@//10.82.175...
ONE	C#ONE@//10.0.0.1...
THL	ICDADM@//10.0.0.10...

Connection Name: ONE

Username: C#ONE

Password:

☒ Save Password ☐ Connection Color

Oracle

Connection Type: Basic Role: default

Hostname: 10.0.0.14

Port: 1521

☒ SID: ORCL

☐ Service name

☐ OS Authentication ☐ Kerberos Authentication ☐ Proxy Connection

Status :

Help Save Clear Test Connect Cancel

Câu 1: Giải thích các hàm thông dụng sau

Stt	Tên Hàm	Mục đích sử dụng & nên sử dụng khi nào
1	Count()	Hàm COUNT() trả về số lượng dòng phù hợp với một tiêu chí được chỉ định.
2	Sum()	Hàm SUM() trả về tổng của một cột số.
3	MAX()	Hàm MAX() trả về giá trị lớn nhất trong một cột số
4	MIN()	Hàm MIN() trả về giá trị nhỏ nhất trong một cột số.
5	NVL()	Hàm NVL() cho phép thay thế một giá trị null bằng một giá trị khác.
6	TO_CHAR()	Hàm TO_CHAR() dùng để chuyển đổi một số hoặc ngày tháng sang kiểu chuỗi.
7	TO_DATE()	Hàm TO_DATE() có tác dụng chuyển định dạng của một chuỗi hoặc một dãy số sang định dạng ngày tháng
8	TO_NUMBER()	Hàm TO_NUMBER() dùng để chuyển đổi một chuỗi sang kiểu số
9	SUBSTR()	Hàm SUBSTR() dùng để trích xuất một chuỗi con từ một chuỗi
10	REPLACE()	Hàm REPLACE() dùng để thay thế chuỗi ký tự này thành chuỗi ký tự khác trong một chuỗi cho trước
11	REVERSE()	Hàm REVERSE() dùng để đảo ngược một chuỗi
12	DECODE()	Hàm DECODE() dùng để hỗ trợ tính năng câu lệnh IF-THEN-ELSE hay CASE
13	TRUNC()	Hàm TRUNC() dùng để thu gọn một số đến một vị trí nào đó sau dấu phẩy
14	LENGTH()	Hàm LENGTH() dùng để đếm tổng số ký tự có trong chuỗi, kể cả ký tự khoảng trắng.
15	LPAD()	Hàm LPAD() trả về một chuỗi được đệm vào bên trái với một chuỗi được chỉ định theo một độ dài nhất định.
16	RPAD()	Hàm RPAD() trả về một chuỗi được đệm vào bên phải với một chuỗi được chỉ định theo một độ dài nhất định.

17	TRIM()	Hàm TRIM() dùng để loại bỏ các khoảng trắng 2 bên của một string. Ta có thể chỉ định các ký tự để loại bỏ.
18	LTRIM()	Hàm LTRIM() dùng để loại bỏ các khoảng trắng bên trái của một string. Ta có thể chỉ định các ký tự để loại bỏ.
19	RTRIM()	Hàm RTRIM() dùng để loại bỏ các khoảng trắng bên phải của một string. Ta có thể chỉ định các ký tự để loại bỏ.
20	ROUND()	Hàm ROUND() dùng để làm tròn số đến một vị trí phần nguyên hoặc vị trí phần thập phân được chỉ định.
21	ADD_MONTHS()	Hàm ADD_MONTHS() có tác dụng thêm hoặc bớt đi một hoặc nhiều tháng từ date truyền vào.

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
- Lấy ra các giá trị của bảng MDM_CUSTOMER theo điều kiện giá trị cột CUST_LGL_ENG_NM phải chứa chuỗi “_LOGISTICS” , sau đó sắp xếp theo cột CUST_LOCL_LANG theo thứ tự null được xếp đầu.
- B) Ý nghĩa của việc dùng ESCAPE
- Để xóa ký tự ‘\’ , với mục đích biến ký tự ‘_’ trở về 1 ký tự bình thường
- C) Ý nghĩa của việc dung Nulls First.
- Dùng để đưa những thuộc tính có giá trị null lên đầu.
- D) Ý nghĩa của việc dung alias, có nên dung alias trong mọi trường hợp không?
- Dùng ALIAS giúp cho câu SQL ngắn gọn, dễ nhìn hơn, đỡ nhầm lẫn khi các cột cùng tên, tiết kiệm được thời gian với những thuộc tính có tên dài hoặc được gọi nhiều. ều
 - Không nên, chỉ nên dùng trong các trường hợp cụ thể.

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 hợp lý, bởi vì oracle có hỗ trợ so sánh số với một chuỗi số, vậy nên việc dùng TO_CHAR sẽ dư thừa và làm giảm tốc độ xử lý.

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

Worksheet

Query Builder

- A) Giải thích ý nghĩa COUNT(*), COUNT(1), COUNT(CUST_LOCL_LANG_NM)
- COUNT(*) : đếm số lượng dòng trả về, kể cả giá trị null
 - COUNT(1) : đếm số lượng dòng trả về, kể cả giá trị null
 - COUNT(CUST_LOCL_LANG_NM): đếm số lượng dòng trả về, chỉ tính những dòng có giá trị khác null ở thuộc tính CUST_LOCL_LANG_NM
- B) Tại sao COUNT(CUST_LOCL_LANG_NM) lại bằng 0
- Tại vì các dòng trả về, giá trị của thuộc tính CUST_LOCL_LANG_NM đều bằng 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 sẽ tối ưu hơn, vì hàm NVL chỉ chạy 1 lần. trong khi cách 2 hàm NVL phải chạy n lần.

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 sẽ tốt hơn, vì cách 2 nếu như kết quả 1 trong 2 cột bị null thì kết quả tổng sẽ null theo.

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.

Cách 1 tốt hơn vì kết quả truy vấn con ít, vì vậy hiệu năng của IN sẽ nhanh hơn EXISTS

- IN được dùng trong trường hợp truy vấn con có kết quả truy vấn nhỏ, không cần dữ liệu từ bảng khác. IN còn có thể được dùng truy vấn giá trị
- EXISTS được dùng trong trường hợp truy vấn con có kết quả truy vấn lớn, cần dữ liệu từ bảng khác. EXISTS chỉ có thể dùng trong truy vấn con

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 số 1 sẽ tốt hơn, bởi vì khi xử lý data với số lượng ít, sử dụng phép join sẽ hoạt động tốt hơn subquery.

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

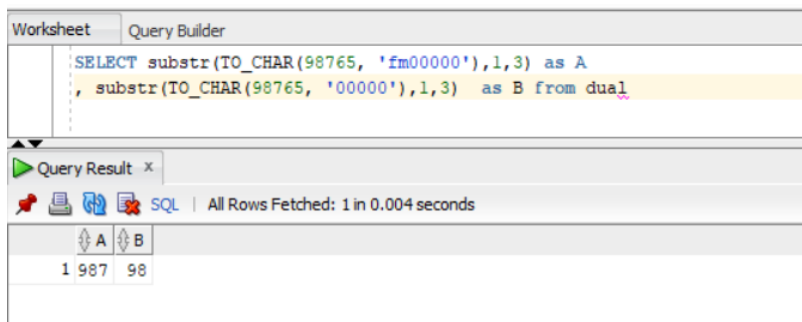
```
SELECT TO_CHAR(8988.80, '$9,999.999') FROM DUAL
```

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),'$9,999.99')FROM DUAL
```

```
SELECT TO_CHAR(TRUNC(820988.80 ,-3),'$999,999.99')FROM DUAL
```

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



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

- Khi dùng TO_CHAR để chuyển sang kiểu string, ta sẽ có thêm 1 ký tự ở đầu đại diện cho dấu, nhưng vì số ta truyền vào là dương, nên ký tự đầu sẽ là khoảng trắng.
- Mà A có fm ở tham số format_mask nên sẽ bị xóa đi khoảng trắng, còn B thì không.
- Vậy nên substring của B lấy 3 ký tự đầu, sẽ bao gồm có khoảng trắng ở ký tự đầu

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

```
SELECT sysdate-1 as Yesterday, sysdate as Today ,sysdate+1 as Tomorrow from dual
```

Câu 13: ta có table (**TB_ORD**), yêu cầu viết câu SQL để generate ORD_NO có độ 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

```
INSERT INTO TB_ORD(CUST_NO, ORD_NO, PRO_CD, ORD_DTTM)
```

```
SELECT :CUST_NO,:ORD_NO,:PRO_CD ,
```

```
DECODE(COUNT(*), 0, RPAD(TO_CHAR(SYSDATE, 'YYYYMMDD'), 12, '0') + 1, MAX(ORD_DTTM)+1) AS  
DTM
```

```
FROM TB_ORD
```

WHERE ORD_DTTM LIKE TO_CHAR(SYSDATE, 'YYYYMMDD')||'%'

Câu 14: ta có table (**MDM_CUSTOMER**) và dữ liệu như bên dưới

CUST_CNT_CD	CUST_SEQ	CNTR_DIV_FLG	BLK_DIV_FLG	CUST_GRP_ID	CUST_LGL_ENG_NM	CUST_LOCL_LANG_NM	CUST_ABBR_NM	CNTR_CUST_TP_CD	BLK_CUS
1 JP	210679 Y	N	(null)		YAMATO TRADING CORP.,	(null)	YAMATO TRADING	N	(null)
2 JP	203222 Y	N	(null)		YAMATO TRADING CO.,LTD.	(null)	YAMAT.TRDG	N	(null)
3 JP	204547 Y	N	(null)		YAMATO INTERNATIONAL CO. LTD.	(null)	YAMATO INTERNAT B		(null)
4 JP	201791 Y	N	(null)		YAMATO LOGISTICS CO., LTD.	(null)	YAMATO LOGISTIC N		(null)
5 JP	205323 Y	N	(null)		YAMATO LOGISTICS CO. LTD	(null)	YAMATO LOGISTIC N		(null)
6 JP	204170 Y	N	(null)		YAMATO LOGISTICS CO.,LTD.	(null)	YAMATO GLOBAL F	N	(null)
7 JP	205138 Y	N	(null)		YAMATO LOGISTICS CO., LTD.	(null)	YAMATO LOGISTIC N		(null)
8 JP	200779 Y	N	G-JP109737		YAMATO LOGISTICS CO., LTD	(null)	YAMATO LOGISTIC N		(null)
9 JP	208036 Y	N	(null)		HOFU EXPRESS CO., LTD.	(null)	HOFU EXPRESS CO N		(null)
10 KH	200784 Y	N	(null)		WEN YUN CO., LTD	(null)	(null)	N	(null)
11 KH	200709 Y	N	(null)		WEY SAN GARMENT CO., LTD	(null)	WEY SAN GARMENT N		(null)
12 KH	200901 Y	N	(null)		WFP CAMBODIA	(null)	(null)	N	(null)
13 KH	200235 Y	N	(null)		WHITE GOLD IMPORT EXPORT CO., LTD.,	(null)	WHITE GOLD IMPO N		(null)
14 KH	200878 Y	N	(null)		WIDE GATE TRANS CO., LTD.	(null)	(null)	N	(null)
15 KH	200024 Y	N	(null)		THE WILLBES CAMBODIA AND CO.,LTD.	(null)	THE WILLBES CAM N		(null)
16 KR	201402 Y	N	(null)		BOOIL SAFES CO.,LTD.	(null)	BOOIL SAFES CO. N		(null)
17 KR	206581 Y	N	(null)		BOOKANG C TECH CO.,LTD.	(null)	(null)	N	(null)
18 KR	203156 Y	N	(null)		BOOKYUNG MEAT CO., LTD	(null)	BOOKYUNG MEAT C N		(null)
19 KR	204131 Y	N	G-KR204131		BOOMIN ENTERPRISE.	(null)	BOOMIN ENTERPRI B		(null)
20 KR	202725 Y	N	(null)		BOOXEN CO., LTD.	(null)	BOOXEN CO.,LTD. N		(null)

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**

```
SELECT DISTINCT(CUST_GRP_ID)
FROM MDM_CUSTOMER
WHERE CUST_GRP_ID NOT IN (SELECT CUST_GRP_ID
FROM MDM_CUSTOMER
WHERE CUST_GRP_HRCHY_CD = 'G')
AND CUST_GRP_HRCHY_CD IN('I', 'C');
```

B) Viết câu SQL tìm CUST_GRP_ID sao cho: CUST_GRP_HRCHY_CD có **G** và có **I** nhưng không có **C**

```
SELECT DISTINCT(CUST_GRP_ID)
FROM MDM_CUSTOMER A
WHERE CUST_GRP_ID NOT IN (SELECT CUST_GRP_ID
FROM MDM_CUSTOMER
```

```

WHERE CUST_GRP_HRCHY_CD = 'C')

AND EXISTS(SELECT CUST_GRP_ID FROM MDM_CUSTOMER B

WHERE

A.CUST_GRP_HRCHY_CD like 'G'

AND B.CUST_GRP_HRCHY_CD like 'I'

AND A.CUST_GRP_ID = B.CUST_GRP_ID)

```

Câu 15: ta có table (**TB_PROD**) và dữ liệu như bên dưới

	PROD_CD	PROD_NM	PROD_UNIT_AMT
1	00001	Prod 01	100
2	00002	Prod 02	300
3	00003	Prod 03	500
4	00004	Prod 04	300
5	00005	Prod 05	500
6	00006	Prod 06	700
7	00007	Prod 07	800
8	00008	Prod 08	100
9	00009	Prod 09	100
10	00010	Prod 10	100

Viets cấu SQL để suất ra kết 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:

	MAX_AMT	MAX_NAME	MIN_AMT	AVG
1	800	Prod 07	100	350

```

SELECT A.MAX_AMT , A.MIN_AMT , A.AVG, B.PROD_NM AS MAX_NAME
FROM (SELECT MAX(PROD_UNIT_AMT) AS MAX_AMT,

MIN(PROD_UNIT_AMT) AS MIN_AMT,

AVG(PROD_UNIT_AMT) AS AVG

```

FROM TB_PROD) A, TB_PROD B

WHERE B.PROD_UNIT_AMT = A.MAX_AMT

Câu 16: ta có table (**TB_ORD**) và dữ liệu như bên dưới

	CUST_NO	ORD_NO	ORD_DTTM	PRO_CD
1	CUS01	ORD01	201910041500	00001
2	CUS01	ORD02	201909041230	00002
3	CUS01	ORD03	201909041540	00004
4	CUS01	ORD04	201910041520	00001
5	CUS01	ORD05	201908041500	00001
6	CUS02	ORD01	201910041500	00001
7	CUS02	ORD02	201909041230	00003
8	CUS02	ORD03	201909041540	00002
9	CUS02	ORD04	201910041520	00005
10	CUS02	ORD05	201908041500	00006
11	CUS03	ORD01	201910041500	00001
12	CUS03	ORD02	201909041230	00001
13	CUS03	ORD03	201909041540	00001
14	CUS03	ORD04	201910041520	00001
15	CUS03	ORD05	201908041500	00001
16	CUS04	ORD01	201910041500	00003
17	CUS04	ORD02	201909041230	00002
18	CUS04	ORD03	201909041540	00004
19	CUS04	ORD04	201910041520	00002
20	CUS04	ORD05	201908041500	00002

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

SELECT CUST_NO,SUM(PRO_CD) AS TOTAL

FROM TB_ORD

GROUP BY CUST_NO

ORDER BY(TOTAL) DESC

FETCH FIRST 3 ROWS ONLY;

B) Viết cấu SQL lấy ra cái ORD_DT, ORD_TM, PROD_CD gần nhất theo CUST_NO

Kết quả mong đợi như sau:

CUS01	ORD02	201911130002	2
CUS02	ORD05	201908041500	6
CUS03	ORD05	201908041500	1
CUS04	ORD06	201911190001	4

```
SELECT *
```

```
FROM (
```

```
    SELECT CUST_NO,ORD_DTTM, ROW_NUMBER () OVER (PARTITION BY CUST_NO ORDER BY  
ORD_DTTM DESC) rn
```

```
    FROM TB_ORD
```

```
    GROUP BY CUST_NO, ORD_DTTM
```

```
) A
```

```
WHERE A.RN = 1;
```

```
SELECT A.DT, B.PRO_CD, NVL(B.TOTAL,0) AS TOTAL
```

```
FROM
```

```
(
```

```
    SELECT '201906' AS DT FROM DUAL
```

```
    UNION ALL
```

```
    SELECT '201907' AS DT FROM DUAL
```

```
    UNION ALL
```

```
    SELECT '201908' AS DT FROM DUAL
```

```
    UNION ALL
```

```
    SELECT '201909' AS DT FROM DUAL
```

```
) A
```

```
LEFT OUTER JOIN
```

```
(SELECT B.PRO_CD, SUBSTR(B.ORD_DTTM, 1, 6) AS ORD_DTTM, COUNT(*)
```

AS TOTAL

FROM TB_ORD B

GROUP BY B.PRO_CD, SUBSTR(B.ORD_DTTM, 1, 6)

) B PARTITION BY (B.PRO_CD)

ON A.DT = B.ORD_DTT;

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

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

	MON	PRO_CD	NVL(B.TOTAL,0)
1	201906	00001	0
2	201907	00001	0
3	201908	00001	2
4	201909	00001	2
5	201906	00002	0
6	201907	00002	0
7	201908	00002	1
8	201909	00002	2
9	201906	00003	0
10	201907	00003	0
11	201908	00003	0
12	201909	00003	1
13	201906	00004	0
14	201907	00004	0
15	201908	00004	0
16	201909	00004	2
17	201906	00005	0
18	201907	00005	0
19	201908	00005	0
20	201909	00005	0
21	201906	00006	0

SELECT *

```

FROM (
    SELECT CUST_NO,ORD_DTTM, ROW_NUMBER () OVER (PARTITION BY CUST_NO ORDER BY
ORD_DTTM DESC) rn
    FROM TB_ORD
    GROUP BY CUST_NO, ORD_DTTM
) A
WHERE A.RN = 1;

```

D) giả sử lúc đầu sản phẩm 00001 có 100 cái, viết report để tính số lượng remain theo tháng 06, 07, 08, 09

	PRO_CD	MONTH	TOTAL	REMAIN
1	00001	201906	0	100
2	00001	201907	0	100
3	00001	201908	2	98
4	00001	201909	2	96

```

SELECT A.DT, B.PRO_CD, NVL(B.TOTAL,0) AS TOTAL, 100 - NVL(SUM(B.TOTAL) OVER (PARTITION BY
B.PRO_CD ORDER BY A.DT),0) AS REMAIN
FROM
(
    SELECT '201906' AS DT FROM DUAL
    UNION ALL
    SELECT '201907' AS DT FROM DUAL
    UNION ALL
    SELECT '201908' AS DT FROM DUAL
    UNION ALL
    SELECT '201909' AS DT FROM DUAL
) A
LEFT OUTER JOIN
(SELECT B.PRO_CD, SUBSTR(B.ORD_DTTM, 1, 6) AS ORD_DTTM, COUNT(*)
AS TOTAL

```

```
FROM TB_ORD B
WHERE B.PRO_CD = '00001'
GROUP BY B.PRO_CD, SUBSTR(B.ORD_DTTM, 1, 6)
) B PARTITION BY (B.PRO_CD)
ON A.DT = B.ORD_DTTM;
```