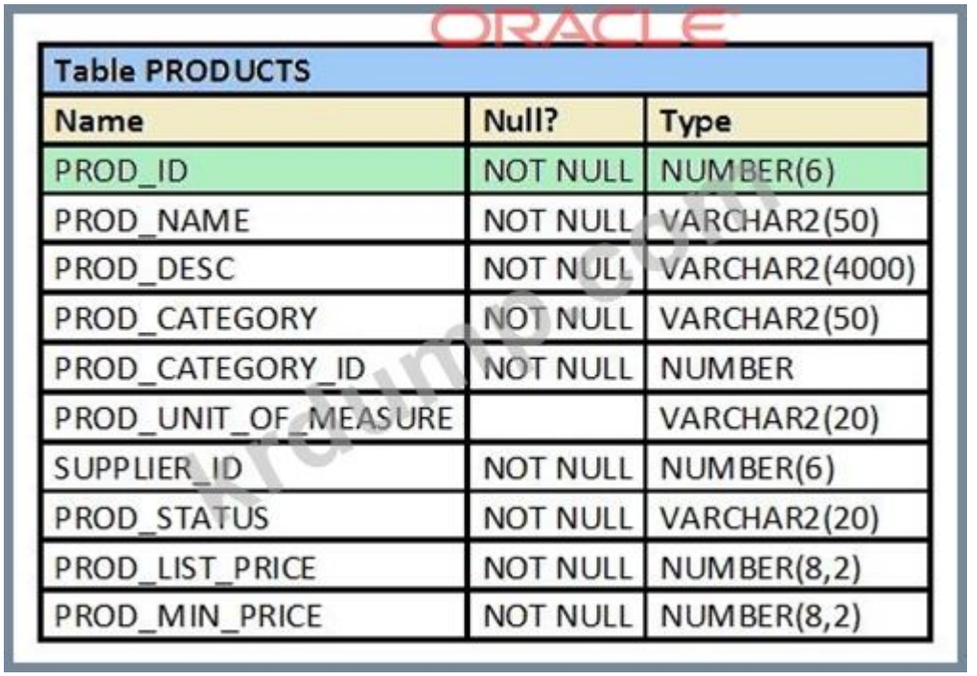
Oracle.1z0-071.v2022-03-01.q157

□□□□:	1z0-071
□□□□:	Oracle Database SQL
□□□:	Oracle
□□ □□ □□□:	157
□□:	v2022-03-01
# □□ □:	2711
# □□ □□□:	1570
https://www.krdump.com/Oracle.1z0-071.v2022-03-01.q157.html	

NEW QUESTION: 1
□□□□ □□ PRODUCT □□□□ □□□ □□□□□□.



□□ □□□ □□□ □ □□ □□□ □□□□□? (2□□ □□□□□.)
A. PROD_MIN_PRICE□ □□ □□□ □□ PROD_LIST_PRICE□□ □□ □□□ □□ □□□ □□ □□ □ □□□□□.
B. □ □□ □□□ □□ □□ PROD_LIST_PRICE □□
C. PROD_LIST_PRICE□ 1000 □□□ □□□ □□
D. □□□(102)□ □□□ □ □□ □□ □□□□ □□ □□□ 'OBSOLETE'□ □□
E. PROD_LIST_PRICE□ □□ PROD_LIST_PRICE□□ □ □□□ □□ □□□□□.
Answer: A,E (LEAVE A REPLY)

NEW QUESTION: 2

□□ Oracle □□□□ □□□□ □□ □□□□ □□□□□□.

```
CREATE TABLE product (pcode NUMBER(2), pname VARCHAR2(20))

INSERT INTO product VALUES (1, 'pen');

INSERT INTO product VALUES (2, 'pencil');

INSERT INTO product VALUES (3, 'fountain pen');

SAVEPOINT a;

UPDATE product SET pcode = 10 WHERE pcode = 1;

COMMIT;

DELETE FROM product WHERE pcode = 2;

SAVEPOINT b;

UPDATE product SET pcode = 30 WHERE pcode = 3;

SAVEPOINT c;

DELETE FROM product WHERE pcode = 10;

ROLLBACK TO SAVEPOINT b;

COMMIT;
```

□ □□ □□ □ □□ □□?

A. □□□□ □□□ □□ □□□□.
B. □□ □□□ 1□□□.
C. □□ □□□ 10□□□.
D. □□ □□□ □□ □□□□.
E. □□□ □□□ □□ □□□□.
F. □□□□ □□□ 3□□□.
Answer: B,E,F (LEAVE A REPLY)

NEW QUESTION: 3
□□ □□□ □□□□□ □□□□□.
CREATE GLOBAL TEMPORARY TABLE □□ _ gtt
(□□ ID INTEGER,
□□□□_ □ NUMBER(10, 2)
) □□ □□ □□□;

INSERT INTO □□□□_ gtt VALUES (1, 100);
□ □□□;
□□ □□ □□□□□?
A. □ □□□□ □□□□ □□□□□ □□ □□□□ □□□ □□□.
B. □ □□□□ □□□□ □□ □□□ □ □□□□.
C. □□□ □□□□ □□ □□ □□□□□.
D. □□ □□□□ □□□ □□ □ □ □□□□.
E. □□□□ □□ □□ □□□ □ □□□□.
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 4
□□ □□□ □□□□□.
□□ d1 CROSS JOIN □□ d2 CROSS JOIN □□ d3□□ 2□ □□□□□□.
□□ □ □□□ □□□□□?
A. 8□
B. 3□
C. 0□
D. □□
E. 1□
F. 6□
Answer: E (LEAVE A REPLY)

NEW QUESTION: 5
□□ □□□ □□□□ □□□ □ □□ □□□ □□□□.
□□ □□ □□□ □□ □□□□ □□□□ □ □□ HCM.EMPLOYEE_RECORDS □□□□ □□ EMP□
□ □□□□ □□□□□?
A. hcm.employee_records□ □□ □□□ □□ emp;
B. hcm.employee_records□ □□ □□□ □□□ emp □□;
C. hcm□ □□ □□ □□□ emp□ □□□□□. □□ □□;
D. hcm.employee_records□ □□ SYNONYM SYS.emp □□;
E. hcm.employee_records□ □□ □□□ PUBLIC.emp □□;
Answer: (SHOW ANSWER)

NEW QUESTION: 6
non-equijoin □ □□□ □□ □□□□ □□ □□ □□?
A. BETWEEN □□□ □□ >= □ <= □□□ □□□□ □□□ □□□ □□□□□.
B. □□□ □□ □□□ □□□ □□□ □□□□.
C. □□□ □□□ □□□ □□□□ □ □□□□.
D. BETWEEN □□□ >= □ <= □□□ □□□□ □□□ □□ □ □ □□□□□.
E. Oracle □□ □□□ SQL:1999 □□ ANSI □□ □□□□ □□□ □□□□.

Answer: C (LEAVE A REPLY)
□□/□□: https://www.academia.edu/17342225/SQL_notes

NEW QUESTION: 7
ORDERStable□□ ORDER_ID □□ □□ □□ □ □□ □□□ □□□□.
ORDER_ITEMStable□□ ORDER_ID □□ □□ □□ □ □□ □□□ □□□, □□ ORDERStable□ □ □ □□ □□□□□.
□□ □□□ ON DELETE CASCADE□ □□□□□.
ORDER_TOTAL□ 1000 □□□ ORDERS □□□□ □□ □□□□.
□□ □ □□ DELETE□□ □□□□□ □□□□□?
A. DELETE FROM □□ WHERE order_total < 1000;
B. □□□□ □□;
C. DELETE * FROM □□ WHERE order_total < 1000;
D. DELETE □□ WHERE order_total < 1000;
E. DELETE order_id FROM □□ WHERE order_total < 1000;
Answer: B,C,D (LEAVE A REPLY)

NEW QUESTION: 8
PRODUCTStable□ □□ □□□ □□□□□.

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(2)
PRODUCT_NAME		VARCHAR2(10)
UNIT_PRICE		NUMBER(3)
SURCHARGE		VARCHAR2(2)
EXPIRY_DATE		DATE
DELIVERY_DATE		DATE

□□ □ □□□ □□□□ □□□□ □ □□ □□□ □□□□□? (3□□ □□□□□.)
A. SELECT product_id, (expiry_date - delivery_date) * 2 FROM □□;
B. □□□□;
SELECT product_id, unit_price, unit_price + □□ □□ FROM □□;
C. SELECT product_id, expiry_date * 2 FROM □□;
D. □□□□ □□;
SELECT product_id, unit_price, 5 "□□", unit_price + □□ □□ - □□
E. □□ ID, □□ □□ | | 5 "□□", □□ + □□ □□ -
F. □□ ID □□, (□□ * 0.15 / (4.75 + 552.25)) FROM □□;
Answer: B,C,D (LEAVE A REPLY)

NEW QUESTION: 9
□□ SQL □□ □□□□□.

```
SQL> SELECT promo_id, promo_category
FROM promotionsd
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';
```

□ □□ □□□ □□ □□□□ □□ □□?

A. SET □□□□ □□ ORDER BY □□□ □□ □□□□ □□□ □ □□ □□□ □□□ □□□□□.

B. □□□□□ □□□□ PROMO_CATEGORY□ □□□□□□ □□ □□□□□.

C. ORDER BY □□ □□ □□□ □, □ □□□ SELECT □□ □□ □□□□ □□□ □□□ □□□□□.

D. □□□□□ □□□□□ □□□□ □□ □□□□ □□ □□□ ORDER BY □□ □□□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 10

□□□□ □□□□ PROMOTIONS □□□□ □□□ □□□□□□.

Table PROMOTIONS

Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(6)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY_ID	NOT NULL	NUMBER
PROMO_CATEGORY	NOT NULL	VARCHAR2(30)
PROMO_CATEGORY_ID	NOT NULL	NUMBER
PROMO_COST	NOT NULL	NUMBER(10,2)
PROMO_BEGIN_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

PROMOTIONS □□□□ □□□□

□□ A□□ $0-2000 □ $2000-5000 □□□ □□ □□□□□ □□ □□ □□□ □□□ □□□.

□□ SQL □□ □□□□□.

```
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;
```

□□□ □□□ □□□□□?
A. WHEN □□ □□ □□□ □□□ □ □□□□ □□□ □□□□□.
B. □□□□□ □□□□ □□□ □□□ □□□□□.
C. CASE□ □□ □□□ □□ □□□ □ □□□□ □□□ □□□□□.
D. □□ □□□ NULL□ □□□ □ □□ □□□ □□□ □□□□□.
Answer: B (LEAVE A REPLY)
□□/□□:
□□:
CASE □□□
IF-THEN-ELSE □□ □□□ □□□□ □□□ □□□ □□□□ □□□.
CASE expr WHEN comparison_expr1 THEN return_expr1
[WHEN comparison_expr2 THEN return_expr2
WHEN comparison_exprn □ □□ return_exprn
ELSE else_expr]
□

NEW QUESTION: 11
□□□□ □□ ORDERS_MASTER □ MONTHLY_ORDERS □□□□ □□□□ □□□□□.
ORDERS_MASTER

ORDER_ID	ORDER_TOTAL
1	1000
2	2000
3	3000
4	

MONTHLY_ORDERS

ORDER_ID	ORDER_TOTAL
2	2500
3	

□□ MERGE □□ □□□□□.
MERGE_INTO □□_□□□ o
USINGmonths_orders m
ON (o.order_id = m.order_id)
□□□ □
□□□□ SET o.order_total = m.order_total
DELETE WHERE(m.order_total IS NULL)
□□□□ □□ □
INSERT VALUES (m.order_id, m.order_total);

□ □□□ □□□ □□□ □□□?
A. ORDERS_MASTER □□□□□ ORDER_ID 1, 2, 3 □ 4□ □□□□□.
B. ORDERS_MASTER □□□□□ ORDER_ID 1, 2 □ 4□ □□□□□.
C. ORDERS_MASTER □□□□□ ORDER_ID 1, 2 □ 3□ □□□□□.
D. ORDERS_MASTER □□□□□ ORDER_ID 1□ 2□ □□□□□.
Answer: B (LEAVE A REPLY)
https://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9016.htm

NEW QUESTION: 12
□□ □□□ □□ □□□□ □□ □□?
A. □□ □□□ □□ □□□ □□□ □ □□□□.
B. AVG □□□ □□□□□ NULLS□ 0□□ □□□□□.
C. MAX □ MIN □□□ □□ □□□ □□□ □□ □□ □□□ □ □□□□.
D. □□ □□□ SELECT □□ □□ □□□ □□□ □ □□□□.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 13
□□□□ □□ ORDERS □ CUSTOMERS □□□□ □□□ □□□□□□.
(□□ □□ □□ □□□□□.)
□□ UPDATE □□ □□□□□□.
□□□□
(SELECT order_date, order_total, customer_id FROM □□)
order_date = '2007□ 3□ 22□' □□
WHERE □□ ID IN
(□□□□ □□ ID □□
WHERE cust_last_name = '□□□' AND credit_limit = 600);
□□□ □□ □□□□ □□ □□?
A. □□□ □□ □□ SELECT □□ □□□ □ □□ □□□ □□□□ □□□□.
B. UPDATE □□ WHERE □□□ □□□□ □□□ □ □□ □□□ □□□□ □□□.
C. □□□ UPDATE □□□ □ □□ □□□□ □□□ □ □□ □□□ □□□□ □□□□.
D. SELECT □□ □□□ □□ □□ □□□ □□□□ □□□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 14
□□□□ □□ PRODUCT □□□□ □□□ □□□□□□.

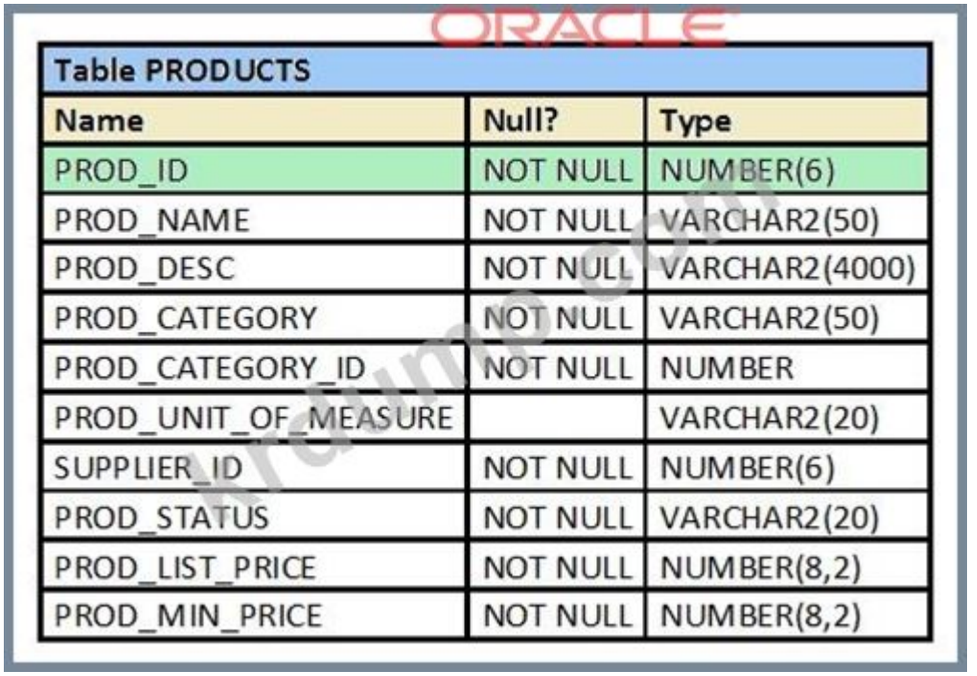


Table PRODUCTS		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(6)
PROD_NAME	NOT NULL	VARCHAR2(50)
PROD_DESC	NOT NULL	VARCHAR2(4000)
PROD_CATEGORY	NOT NULL	VARCHAR2(50)
PROD_CATEGORY_ID	NOT NULL	NUMBER
PROD_UNIT_OF_MEASURE		VARCHAR2(20)
SUPPLIER_ID	NOT NULL	NUMBER(6)
PROD_STATUS	NOT NULL	VARCHAR2(20)
PROD_LIST_PRICE	NOT NULL	NUMBER(8,2)
PROD_MIN_PRICE	NOT NULL	NUMBER(8,2)

□□ □□□ □□□ □ □□ □□□ □□□□□? (2□□ □□□□□.)

A. □□ □□□ □□ □□ □□

B. □□ □□□ '□□ □□' □□□ □□□ □□ □□□□ □□ □□ □□□ □□□□□.

C. □□□ 1000 □□□ □□ □□□□□ □□□□□.

D. □□□ □□ □□□□ □□ □□□ □□□ □□□□□.

E. □□□(102)□ □□□ □ □□ □□ □□□□ □□ □□□ 'OBSOLETE'□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 15

□□ □□□□ □□ □□□□ □□ □□?

A. CREATE TABLE AS SELECT □□ □□ □□□□□ □□□□□□□ □□ □□□□ □□□□ □□□ □□ □ □□□ □ □□□□.

B. □□ □□□□ □□ □□□ □ □□□□□□ □□□□□□ □□□ □□□□□.

C. □□ □□□□ □□ REJECT LIMIT□ UNLIMITED□□□.

D. ORACLE_LOADER □ ORACLE_DATAPUMP□ □□ □□□□ □□ □□□ □ □□□ □□□ □□□ □□□□.

Answer: A (LEAVE A REPLY)

https://docs.oracle.com/cd/B28359_01/server.111/b28310/tables013.htm

NEW QUESTION: 16

Exhibit□ □□ CUSTOMERStable□ □□□ □□□□□□.

Table CUSTOMERS		
Name	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_FIRST_NAME	NOT NULL	VARCHAR2 (20)
CUST_LAST_NAME	NOT NULL	VARCHAR2 (40)
CUST_GENDER	NOT NULL	CHAR (1)
CUST_YEAR_OF_BIRTH	NOT NULL	NUMBER (4)
CUST_MARITAL_STATUS		VARCHAR2 (20)
CUST_STREET_ADDRESS	NOT NULL	VARCHAR2 (40)
CUST_POSTAL_CODE	NOT NULL	VARCHAR2 (10)
CUST_CITY	NOT NULL	VARCHAR2 (30)
CUST_STATE_PROVINCE	NOT NULL	VARCHAR2 (40)
COUNTRY_ID	NOT NULL	NUMBER
CUST_INCOME_LEVEL		VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2 (30)

CUSTOMERS □□□□ □□□□ □□ □□□ □□ 15%□ □□ □□ □□□ □□□□ □□□□ □□□ □ □□□.

□□ □□□ □□ □□□ "□□□ □ □□"□ □□□□□ □□□.

□□ SQL □□ □□□ □□□ □□□□□?

SELECT NVL(TO_CHAR(cust_credit_limit*.15),'□□□ □ □□') "NEW CREDIT"

A. □□□□□□;

SELECT NVL(cust_credit_limit*.15,'□□□ □ □□') "NEW CREDIT"

B. □□□□□□;

C. □□□□□□;

SELECT NVL(cust_credit_limit,'□□□ □ □□')*.15 "NEW CREDIT"

D. □□□□□□;

SELECT TO_CHAR(NVL(cust_credit_limit*.15,'□□□ □ □□')) "NEW CREDIT"

Answer: D (LEAVE A REPLY)

NEW QUESTION: 17

□□□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. && □□□□ □□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□ □□ □ □□□ □□ □□ □ □□ □□□□□ □□□□□.
B. □□ □□□ SELECT □□□□ □□□ □ □□□□.
C. & □□□□ □□ □□ □□□ □□ □□□ □□ □□ □ □□ □□□□□□□.
D. □ □□□ □□ □ □□□□ □□ □□□ □□□□□□ □□□ □□□.
E. SELECT □□ □□ □□ □□ □□ □□□ □□□ □ □□□□.
F. □ □□□ □□ □ □□□□ □□ □□□ □□□□□ □□□ □□□.
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 18
□□ □ □□□ □□□ □□□□ □□□□ □□□□ □□□□□?
A. GRANT CREATE TABLETO user1, user2;
B. GRANT ALTER TABLETO PUBLIC;
C. □□□□ □□ □□ □□ □□
D. GRANT CREATE VIEWON table1 TOuser1;
Answer: (SHOW ANSWER)

NEW QUESTION: 19
□□ □□□ □□□□□.

```
SELECT TO_CHAR(NEXT_DAY(LAST_DAY(SYSDATE), 'MON'), 'dd "Monday for" fmMonth rrrr')
  FROM DUAL;
```

□□□ □□□□□?
A. □□ □□ □□□ □□□ □□□ □□□□□.
B. □□□ □□□□□.
C. □□□□□ □□□□□ □□□ □□□□ □□□□.
D. □□ □ □□ □□□□ □□□ □□□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 20
□□□ □□ □□ □□□□ WITH □□ □□□□ □□ □□□□ □□ □ □□ □□ □ □□ □□ □□□□ □? (3□□ □□□□□□.)
A. □□ □□ □□□ □□□ □□□ □□□ □□ □□□ □□□ □□□□□.
B. WITH □□ □ □□□ □□□ □□□ □ □□□□.
C. WITH □□ □□ □□□ WITH □□ □□ □□ □□□ □□ □□ □□□□ □ □ □□□□.
D. SELECT □□□ □□□ □ □□□□.
Answer: B,C,D (LEAVE A REPLY)

NEW QUESTION: 21
□□□ ANDREW□ □□□ □□□□□ □□□□ □□□.
1. CUSTOMERS □□□□ TITLE □ ADDRESS □□ □□□□□.

2. □□ □□□□□ □□□ □□□□□.
□□ □□□ □□□ □ □□□□?
A. □□□□ □□□□□ □□□□□. □□, □□.□□ □□□□□ GRANT OPTION;
B. □□□ □□□ □□ □□□□□ □□□□ □□□□(□□, □□) □□;
C. □□□□ □□□□□ □□□□□. □□, □□ □□. □□□□□;
D. □□□ □□□ □□ □□□□□ □□.□□, □□.□□□ □□ □□□□ □□;
E. □□□□ □□□□(□□, □□)□ □□□□□.
F. andrew with GRANT OPTION □□□ □□ GRANT □□□□(□□, □□)
Answer: (SHOW ANSWER)

NEW QUESTION: 22
□□□□ □□ DEPT □ LOCATIONS □□□□ □□ □□□ □□□□□□.

DEPT

Name	Null?	Type
DEPARTMENT_ID		NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)
CITY		VARCHAR2(30)

LOCATIONS

Name	Null?	Type
LOCATION_ID	NOT NULL	NUMBER(4)
STREET_ADDRESS		VARCHAR2(40)
POSTAL_CODE		VARCHAR2(12)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_ID		CHAR(2)

□ □□□ □□ LOCATIONS □□□□ CITY □□ □□ □□ □□□ □□ □□ □□ DEPT □□□□ CITY □□ □□□□□□□ □□□.
□□□ □□□□ □□ □□ SQL □□ □□□□□□□□?
A. UPDATE □□ dSET □□ = ANY(SELECT cityFROM □□ l)
B. □□□□ □□ dSET □□ = (SELECT cityFROM □□ l)WHERE d.location_id = l.location_id;
C. □□□□ □□ dSET □□ = ALL (SELECT cityFROM □□ lWHERE d.location_id = l.location_id);
D. □□□□ □□ dSET □□ = (SELECT cityFROM □□ lWHERE d.location_id = l.location_id);
Answer: D (LEAVE A REPLY)

NEW QUESTION: 23
□□ □ □□ □□□ □□ □ □□ □□ □ □□ □□?
A. □□ □□□ □□ □□□ □□□ □□□□ □□□.
B. □□ □□□ □□ □□□ □□□□ □□□□ □□□.
C. SQL □□ □□□ □□□□ □□ □□ □□□□ □□□.
D. SQL □□□ □□ □ □□ □□ □□□ □□ □ □□ □ □□□□.
E. HAVING □□□ □□□ □ □□□□.
F. □□□ □□□ □ □□□□.
Answer: D,E,F (LEAVE A REPLY)

NEW QUESTION: 24

MERGE □□ □□ □□□□ □□ □□?

A. □□ □□□ □□□□ □□ □□ □□□ □ □□□□.

B. □□ □□□□□ □□□□ □□ □□□□, □□ □□ □□□ □ □□□□.

C. □□ □□□□ □□□ □□ □□ □ □□□□□ □ □□□□.

D. □□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□ □ □□□□.

E. □□ □□□□ □□ □□ □□□ □ □□□□.

F. □□□□ □□ □□□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 25

□□ □□□ □□□□□? (□□□□ □□ □□ □□□□□□.)

A. □□□ □□□ □□□□□□ □□□□ □□□ □□ □□□□□.

B. □□□ □□□□ □□ □□□□ □□ □□□□ □□□ □□ □□□□ □□□□ □□□□□.

C. □□□□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□□ □□□ □□□□□.

D. □□ □□□ □□□□ □□ □□□ □□□□ USER_CONS_COLUMNS □□ □□□□ □□□.

E. USER_OBJECTS □ CAT □□ □□ □□□□ □□□ □□ □□□ □□ □□□ □□□ □□□□□.

F. DBA, ALL □ USER□ □□ □□□ □□□ □□□□ □□ □□□ □□□ □□□ □□□ □□ □□□□ □□□□□.

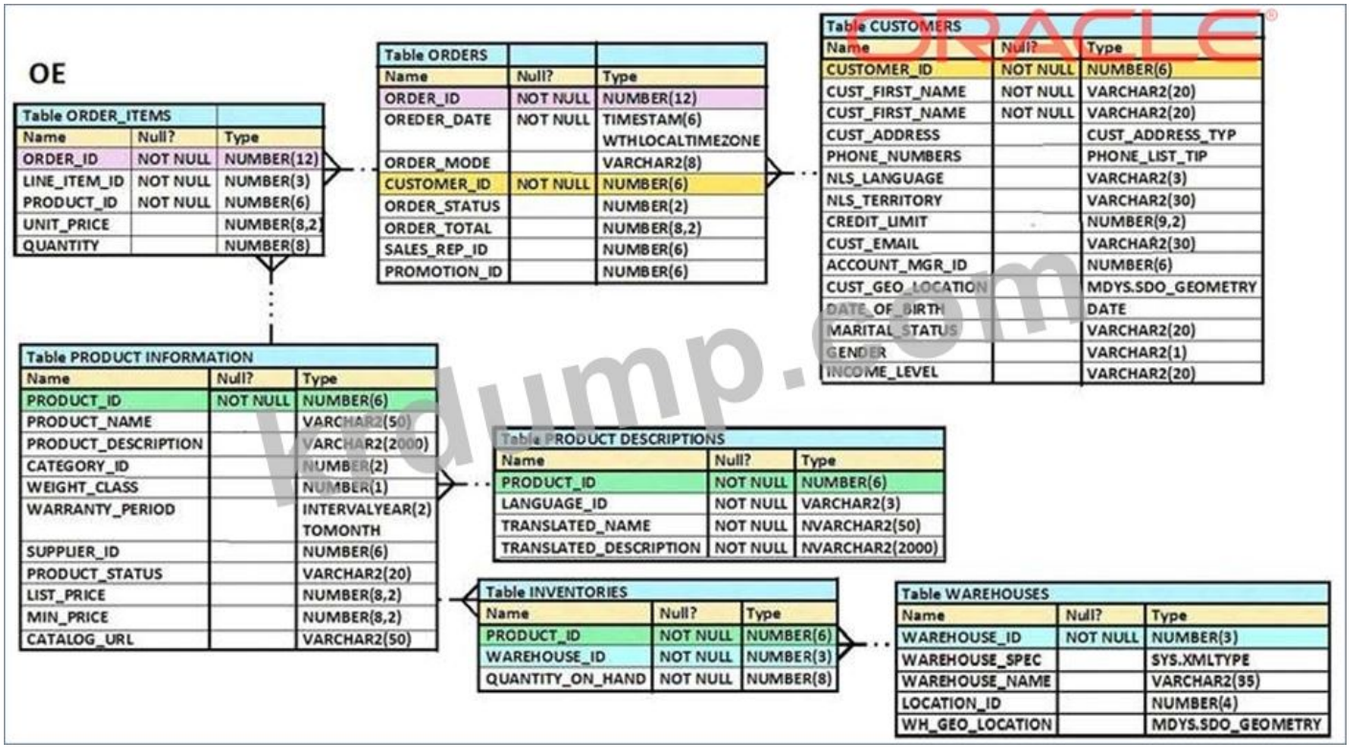
Answer: (SHOW ANSWER)

□□:

https://docs.oracle.com/cd/B10501_01/server.920/a96524/c05dicti.htm

NEW QUESTION: 26

□□□□ □□ ORDERS □ CUSTOMERS □□□□ □□□ □□□□□□.



□□ UPDATE □□ □□□□□□.

```
UPDATE
      ( SELECT order_date, order_total, customer_id FROM orders)
Set order_date = '22-mar-2007'
WHERE customer_id IN
     (SELECT customer_id FROM customers
     WHERE cust_last_name = 'Roberts' AND credit_limit = 600);
```

□□□ □□ □□□□ □□ □□? (□□ □□ □□ □□□□□.)

A. SELECT □□ □□□ □□ □□ □□□ □□□□ □□□□□.

B. □□□ □□ □□ SELECT □□ □□□ □ □□ □□□ □□□□ □□□□.

C. UPDATE □□ WHERE □□□ □□□□ □□□ □ □□ □□□ □□□□ □□□.

D. □□□ UPDATE □□□ □ □□ □□□□ □□□ □ □□ □□□ □□□□ □□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 27

□□□ □□ □□□□ □□ □□?

A. ROLLBACK □□ □□□ □□□□□ □□ □□□ □□□□ □□□□.

B. □□ □ □□□ □□□□ □□□□ □□□□.

C. GRANT □ REVOKE□ □□ DCL(□□□ □□ □□) □□ □□□ □ □□□□.

D. □□□ □□□ □□ □□□ □□□□□ □□□□ □□□□□.

E. TO SAVEPOINT □□ ROLLBACK □□ □□□□ □□□□□ □□ □□□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 28

MEMBERStable□ □□□ □□□□□.

Name	Null?	Type
MEMBER_ID	NOT NULL	VARCHAR2 (6)
FIRST_NAME		VARCHAR2 (50)
LAST_NAME	NOT NULL	VARCHAR2 (50)
ADDRESS		VARCHAR2 (50)
CITY		VARCHAR2 (25)
STATE		VARCHAR2 (3)

A □□□ □□□□ □ □□ □□□ □ □□□ □□ □□ □□□□ □□ □□□□ □□ □□□ □□□□□ □□□.

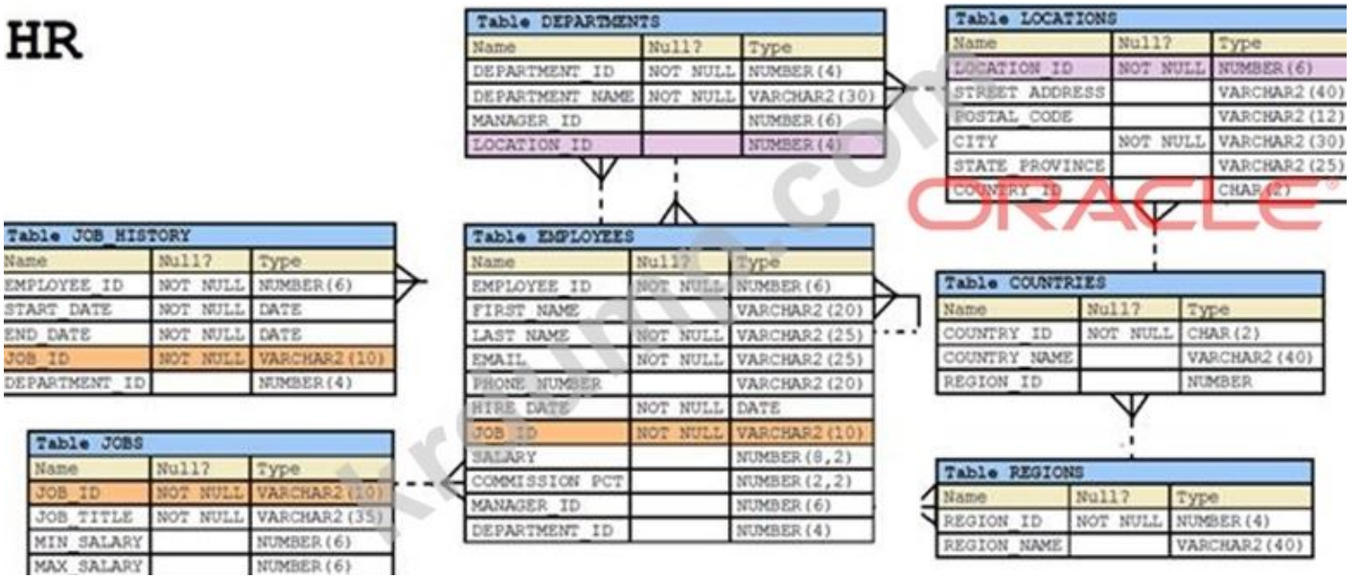
□□ SQL □□ □□□□ □□□?

A. SELECT * FROM MEMBERS WHERE state LIKE '%A_';

B. SELECT * FROM MEMBERS WHERE state LIKE 'A_';

C. SELECT * FROM MEMBERS WHERE state LIKE 'A%';
D. SELECT * FROM MEMBERS WHERE □□ LIKE 'A_%';
Answer: B (LEAVE A REPLY)

NEW QUESTION: 29
□□□□ □□ EMPLOYEES □□□□ □□□ □□□□□□.



□□ SQL □□ □□□□□.
SELECT □□ ID, □□ ID
□□□□□□
WHERE Department_id= 50 order BY department_id
□□ □□
SELECT □□ ID, □□ ID
□□□□□□
□□□ □□ ID = 90
□□ □□
SELECT □□ ID, □□ ID
□□□□□□
□□□ □□ ID = 10;
□□ SQL □□ □□□ □□□□□?
A. □□□□ □□□□□ □□□□□ ORDER BY □□ □□□□ □□ □□□ □□□ □□□□□.
B. ORDER BY □□ □□ □□□ □□ □□ □□□□ □□□□ □□□ □□ □□□□ □□□□.
C. ORDER BY □□ SQL □□ □, □ □□□ SELECT □□□ □□□□ □□□ □□ □□□□ □□□□.
D. □□□□ □□□□□ □□□□ □□ □□ DEPARTMENT_ID□ □□□□□□ □□□□□.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 30
□□ SQL □□ □□□□□.

```
DELETE FROM employees e
 WHERE EXISTS
    (SELECT 'dummy'
       FROM emp_history
      WHERE employee_id = e.employee_id);
```

□□ □□ □□□□□?

A. □□ □□□ □□ □□ □□□□□□ DELETE □□ □□□□□ □□□□□.

B. EMPLOYEEStable□ □□ □□ □□ □□□□□.

C. DELETE□□ □□□□ □□ □□ □□□ □□□□□.

D. □□ □□□ □□ □□ □□□ □□□□.

E. EMPLOYEEStable□ □□ □□ □□ □□ □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 31

GLOBAL TEMPORARY TABLES□ □□ □ □□ □□ □ □□ □□?

A. GLOBAL TEMPORARY TABLES□ DML(□□□ □□ □□)□ REDO□ □□□□ □□□□.

B. GLOBAL TEMPORARY TABLE□ □□□ □□□□ □□ □ □□□□.

C. GLOBAL TEMPORARY TABLE□ PUBLIC SYNONYM□ □□ □ □□□□.

D. GLOBAL TEMPORARY TABLE□ □□ □□□□ □□ □ □□□□.

E. □□ □□ □□□□ GLOBAL TEMPORARY TABLE□ □□□ □ □□□□.

F. □□□□ GLOBAL TEMPORARY TABLE□ □□□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 32

EMPLOYEES □□□□□ 1000□□ □□ □□□ □□□ 10□ □□ □□□□ □□□□ □□□□.

□□ SQL □□ □□□□□.

□□□ □□□ □□□□□?

A. NVL □□□ UPDATE□ □□ □□□ □ □□ □□□ □□□ □□□□□.

B. □□□ □□□□ □□ NVL □□□ □□□□ □□□ □□□ □□□□□.

C. □□□□ 600□ □□ □□□ □□□ □□□ □□□□□ □□□□ □□□□□□□.

D. □□□□□ □□□□□ □□□□□ □□ □□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 33

□□□ □□ □□□□ SALES1 □ SALES2□ □ □□□□ □□□□ □□ □□□□ □□□ □□ □□□□ □. SALES2 □□□□ □□ SALES1 □□□□ □□□ □□□□□ □□□.

SALES1 □□□

□□ □ □□

SALES_IDNUMBER

STORE_IDNUMBER

ITEMS_IDNUMBER

QUANTITYNUMBER

SALES_DATEDATE

SALES2 □□□

□□ □ □□

SALES_IDNUMBER

STORE_IDNUMBER

ITEMS_IDNUMBER

QUANTITYNUMBER

SALES_DATEDATE

□□ □□ □□□□ □□□ □□□ □□□□□?

A. □□

B. □□□

C. □□□

D. □□□□

E. □□

Answer: D (LEAVE A REPLY)

□□:

https://docs.oracle.com/cd/B19306_01/server.102/b14200/queries004.htm

NEW QUESTION: 34

Exhibit□ □□ CUSTOMERStable□ □□□ □□□□□□.

CUSTOMERS

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(6)
CUST_NAME		VARCHAR2(20)
CUST_EMAIL		VARCHAR2(30)
INCOME_LEVEL		VARCHAR2(20)

CUSTOMER_VU□ CUSTOMERStable□ □□□ □□□ □□ CUSTOMERS_BR1 □□□□ □□□□ □□ □□□□.

CUSTOMERS□□□ □□ □□ □□□ □□□□□ □□□□□□ □□□.

□□ MERGE□□ □□□ □□□□□?

```
MERGE INTO customers c
    USING customer_vu     cv
    ON (c.customer_id = cv.customer_id)
WHEN MATCHED THEN
    UPDATE SET
   c.customer_id = cv.customer_id,
   c.cust_name = cv.cust_name,
   c.cust_email = cv.cust_email,
   c.income_level = cv.income_level
WHEN NOT MATCHED THEN
    INSERT VALUES(cv.customer_id,cv.cust_name,cv.cust_email,cv.income_level)
      WHERE cv.income_level >100000;
```

A. INTO□□ □□□□ □□ □□□□□□□.

B. WHERE □□ INSERT□ □□ □□□ □ □□□□.

C. CUSTOMER_VU□ □□□ □□□ □□□ □ □□□□.

D. CUSTOMER_ID □□ □□□□□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 35

EMPLOYEES □□□□□ NUMBER □□□ □□□ EMP_ID □□ DATE □□□ □□□ HIRE_DATE □ □ □□□□.

□□ □ 6□□□ □□ □ □ □□ □□□□ □□□ □□□□□ □□□.

NLS_TERRITORY □□□□□ □□□□ AMERICA□ □□□□□ □□□□ □□ □ □□ □□□□□.

□□ □□□ □□□ □ □□□□?

A. SELECT emp_id, ADD_MONTHS(hire_date, 6), NEXT_DAY('MONDAY') FROM □□;

B. SELECT emp_id, NEXT_DAY (MONTHS_BETWEEN (hire_date, SYSDATE), 6) FROM □□;

C. SELECT emp_id, NEXT_DAY(ADD_MONTHS(hire_date, 6), 1) FROM □□;

D. SELECT emp_id, NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY') FROM □□;
Answer: D (LEAVE A REPLY)

NEW QUESTION: 36
□□ □ □□□ □□□□□.

```
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;
```

□□ □ □□□ □□ □□□□ □□ □□?
A. □□ 2□□ □□□ □□□□□.
B. □□ 2□□ □□□ □□□□□.
C. □□□□ □□□ □□ □□□□.
D. CUST__CREDIT__LIMIT □□ null □□ □□ □□□□ □□ 2□ □□□ □□□□□.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 37
□□□□ □□ PRODUCTS □ SALES □□□□ □□□□□□.
□□ 1

Table PRODUCTS

Name	Null?	Type
PROD_ID	**NOT NULL**	**NUMBER (6)**
PROD_NAME	NOT NULL	VARCHAR2 (50)
PROD_DESC	NOT NULL	VARCHAR2 (4000)
PROD_CATEGORY	NOT NULL	VARCHAR2 (50)
PROD_CATEGORY_ID	NOT NULL	NUMBER
PROD_UNIT_OF_MEASURE		VARCHAR2 (20)
SUPPLIER_ID	NOT NULL	NUMBER (6)
PROD_STATUS	NOT NULL	VARCHAR2 (20)
PROD_LIST_PRICE	NOT NULL	NUMBER (8, 2)
PROD_MIN_PRICE	NOT NULL	NUMBER (8, 2)

□□ 2

Table 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)

□□□ □□□ □□□□ □□ □□□ □□□□□ □□ □□□ □□□□□.

```
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;
```

□□ □□□ □□□□ □□□ □□□?

A. □□□□ □□□□□ □□□□ □□□ □□□ □□□□□.

B. FROM □□ □□ □□□□ GROUP BY □□ □□□ □ □□ □□□ □□□ □□□ □□□□□.

C. □□ □□□ ITEM_CNT□ □□□ □ □□ □□□ □□□ □□□ □□□□□.

D. FROM □□ □□ □□□ □□ □□□ □□ □□□ □ □□ □□□ □□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 38

□□□□ □□ PROMOTIONS □□□□ □□□□ □□□□□□.

PROMO_NAME	PROMO_CATEGORY	PROMO_COST	PROMO_BEGIN_DATE
NO PROMOTION #	NO PROMOTION	0	01-JAN-99
newspaper promotion #16-108	newspaper	200	23-DEC-00
post promotion #20-232	post	300	25-SEP-98
newspaper promotion #16-349	newspaper	400	10-JUL-98
internet promotion #14-471	internet	600	26-FEB-00
TV promotion #13-448	TV	1100	06-AUG-00
internet promotion #25-86	internet	1400	20-SEP-98
TV promotion #12-49	TV	1500	10-AUG-00
post promotion #21-166	post	2000	25-SEP-98
newspaper promotion #19-210	newspaper	2100	19-MAR-99
post promotion #20-282	post	2300	06-DEC-00
newspaper promotion #16-327	newspaper	2800	09-APR-99
internet promotion #29-289	internet	3000	01-NOV-98
TV promotion #12-252	TV	3100	20-JUN-98
magazine promotion #26-258	magazine	3200	04-MAY-00

PROMO_BEGIN_DATE□ □□ □□ □□□ dd-mon-rr□ □□□□□.

2000□ 1□ 1□ □□□ □□□ POST □□□ □□ □□□□□ □□, □□ □ □□ □□□ □□□□ □□ □□ □□□□ □□□.

□□ SQL □□ □□□□□□□□?

A. SELECT promo_name, promo_cost, promo_begin_dateFROM PromotionsWHERE promo_category LIKE '%post%' AND promo_begin_date < '1-JAN-00';

B. SELECT promo_name, promo_cost, promo_begin_dateFROM PromotionsWHERE promo_cost LIKE 'post%' AND promo_begin_date < '01-01-2000';
C. SELECT promo_name, promo_cost, promo_begin_dateFROM □□□□WHERE promo_category = '□□□' AND promo_begin_date < '01-01-00';
D. SELECT promo_name, promo_cost, promo_begin_dateFROM PromotionsWHERE promo_category LIKE 'P%' AND promo_begin_date < '1-JANUARY-00';
Answer: (SHOW ANSWER)

NEW QUESTION: 39
□ □□□□□ □□ □□□□ □ □□ □□□ □□□□□? (3□□ □□□□□.)
A. □□□ □□□□ SELECT□□ □□□ □ TRUNCATE□□ □□□ □
B. □□□ □□□□ CREATE TABLE□□ □□□□□ □□□ □ CREATE INDEX□□ □□□□ □□
C. □□□ □□□□ SELECT□□ □□□ □ CREATE TABLE□□ □□□ □
D. □□□ □□□□ COMMIT □□ ROLLBACK □□ □□□ □ □ □□ DML(□□□ □□ □□) □□ □ □□ □
E. □□□ □□□□ DML □□ □□□ □ DML □□ □□□□ □□
F. CREATE TABLE AS SELECT□□ □□□ □□□□ □□□ □ SELECT FOR UPDATE□□ □□□ □
Answer: D,E,F (LEAVE A REPLY)
□□/□□: https://docs.oracle.com/cd/B19306_01/server.102/b14220/transact.htm

NEW QUESTION: 40
□□ □□□ □□□□□.

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

□ □□□ □□□ □□□ □□□□□?
A. +25-00,+00-650,+00 11:12:10.123457
B. +00-300,+00-650,+00 11:12:10.123457
C. +00-300, +54-02,+00 11:12:10.123457
D. +25-00, +54-02, +00 11:12:10.123457
Answer: (SHOW ANSWER)

NEW QUESTION: 41
□□□□ □□ PRODUCTS □ ORDER_ITEMS □□□□ □□□□□□.

PRODUCTS

PRODUCT ID	PRODUCT NAME
1	Inkjet C/8/HQ
2	CPU D300
3	HD 8GB /I
4	HD 12GB /R

ORDER_ITEMS

ORDER ID	PRODUCT ID	QTY	UNIT PRICE
11	1	10	100
22	2	15	120
33	3	10	50
44	1	5	10
66	2	20	125

PRODUCT_NAME □ □□ □□ □□□ □□□□ □□ □□ □□□ □□□□□□.
SQL>SELECT p.product_name, i.item_cnt
FROM(□□ ID □□, COUNT(*) item_cnt
□□_□□□□
GROUP BY product_id) i RIGHT OUTER JOIN □□ p
ON i.product_id = p.product_id;
□□ □□□ □□□□ □□□ □□□?
A. □□□ □□ □□ □□□ □□ □□□ □ □□ □□□ □□ □□□□ □□□□.
B. □□ □□□ ITEM_CNT □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□.
C. □□□□□ GROUP BY □□ □□□ □ □□ □□□ □□□□ □□□□ □□□□.
D. □□□□ □□□□□ □□□□ □□□ □□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 42
□□ □ □□□ □□ □□ □□□□ □□ □□ □□□□ □□ □□ □□□□□?
A. □2□□□
B. □ □□ □□□
C. □3□□□
D. □4□□□
Answer: (SHOW ANSWER)

https://blog.udemy.com/database-normal-forms/

NEW QUESTION: 43

EMPLOYEES □□□□□ □□ □□□□ □□□□□.

ID	LAST_NAME	SALARY	DEPT_ID
1	Smith	1000	10
2	Jones	2000	10
3	Markham	1500	20
4	Black	1300	20

□□ □□□ □□□□□ □□□□□?

SELECT □□ ID, INSTR(last_name, 'A'), SUM(□□) FROM □□ GROUP BY

A. SELECT □□ ID, LENGTH(last_name), SUM(□□) FROM □□ GROUP BY

B. □□ ID;

SELECT dept_id, MAX(last_name), SUM(□□) FROM □□ GROUP BY dept_id;

C. □□ ID;

D. □□ ID;

SELECT dept_id, STDDEV(last_name), SUM(□□) FROM □□ GROUP BY

Answer: A (LEAVE A REPLY)

NEW QUESTION: 44

□□□□ □□ COSTS □ PROMOTIONS □□□□ □□□ □□□□□□.

□□□□ □□□ □□□□ □□ □□ □□□ PROD ID□□ □□ PROD IDS□ □□□□□ □□□.

□□ □□ □□.

□□ SQL □□ □□□□□.

□□ ID □□

□□□□

WHERE □□□□ ID IN

(□□□□ ID □□

□□□□□□

WHERE promo_cost < □□

(SELECT MAX(□□□□ □□)

□□□□□□

□□ BY (□□□□ □□ □□-□□□□_□□_□□)));

□□□ □□□ □□□?

A. □□□□□ □□□□ □□□ □□□ □□□□□.

B. □□□□□ □□□□□ □□□ □□□ □□□□ □□□□.

C. GROUP BY □□ □□□□ □□ □□□ □□□□□.

D. ALL □□□□ □□□□ □□ □□□ □□□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 45
□□□□ □□ INVOICE □□□□ □□□ □□□□ □□□□□□.

INVOICE

Name	Null?	Type
INV_NO	NOT NULL	NUMBER (3)
INV_DATE		DATE
CUST_ID		VARCHAR2 (4)
INV_AMT		NUMBER (8, 2)

INV_NO	INV_DATE	CUST_ID	INV_AMT
1	01-APR-07	A10	1000
2	01-OCT-07	B1R	2000
3	01-FEB-07		3000

□□ □ □□ SQL □□ □□□□□ □□□□□? (2□□ □□□□□.)
A. SELECT AVG(inv_date - SYSDATE), AVG(inv_amt)FROM □□;
B. SELECT AVG(inv_date)FROM □□;
C. SELECT MAX(inv_date),MIN(cust_id)FROM □□□□;
D. SELECT MAX(AVG(SYSDATE -inv_date))FROM □□;
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 46
SQL □□ WHERE □ HAVING □□ □□ □□□□ □□ □□? (3□□ □□□□□.)
A. HAVING □ □□□ □□ □□□ □□ □ □□□□.
B. HAVING □ □□□ □□ □□ □□□ □□□ □ □□□□.
C. HAVING □□ □□□□ □□□□ □ □□ □□□ □□ □□□ □□□□ □ □□□□□.
D. WHERE □□ □□□□ □□□□□ □□ □□ □□□□ □ □□□□□.
E. WHERE □□ HAVING □□ SQL □□□ □□ □□□ □ □□□□.
Answer: A,C,D (LEAVE A REPLY)

NEW QUESTION: 47

□□□□ □□ CUSTOMERS □□□□ □□□ □□□□□□.

Table CUSTOMERS

Name	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_FIRST_NAME	NOT NULL	VARCHAR2 (20)
CUST_LAST_NAME	NOT NULL	VARCHAR2 (40)
CUST_GENDER	NOT NULL	CHAR (1)
CUST_YEAR_OF_BIRTH	NOT NULL	NUMBER (4)
CUST_MARITAL_STATUS		VARCHAR2 (20)
CUST_STREET_ADDRESS	NOT NULL	VARCHAR2 (40)
CUST_POSTAL_CODE	NOT NULL	VARCHAR2 (10)
CUST_CITY	NOT NULL	VARCHAR2 (30)
CUST_STATE_PROVINCE	NOT NULL	VARCHAR2 (40)
COUNTRY_ID	NOT NULL	NUMBER
CUST_INCOME_LEVEL		VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2 (30)

□□ □□□□□ □□ □□ □□ □□□ □□□□ □□ □ □□ □□□ □□□□□? (2□□ □□□□□.)

A. '□□'□ □□□□ □□□ □□□□□ □□□□□ □□□ □□□ □□□□□.

B. □□□□□ □□ 1980□ □□ □□□ □□

C. '□□'□ '□□□'□ □□□□ □□ □□□ □□ □□ □□ □□□

D. □ □□□□ □□ □□□ '□□'□ □□ □□ □□□□.

E. □ □□□□ □□ □□□ □□ □□ □□□□ □□ □□□ □□ □□ □□ □□□ □

Answer: A,E (LEAVE A REPLY)

NEW QUESTION: 48

NLS_DATE_FORMAT□ DD-MON-YYYY HH24:MI:SS□ □□□ □□□ □□/□□ □□□ □□ □□□□ □□ □□? (2□□ □□□□□.)

A. CURRENT_TIMESTAMP□ CURRENT_DATE□ □□□ □□□ □□□□□.

B. CURRENT_TIMESTAMP□ SYSDATE□ □□□ □□ □ □□□ □□□ □□ □□ □□ □□ □□ □ □□ □□ □□□□□.

C. SYSDATE □ CURRENT_DATE□ □□□□□□ □□□ □□ □□□ □□□ □□ □□ □ □□□ □ □□□□.

D. SYSDATE□ □□ □□ □□□ DD-MON-RR□ □□□□ □□□□ □□□ □ □□□□.

E. SYSDATE□ DUAL □□□□□□ □□□ □ □□□□.

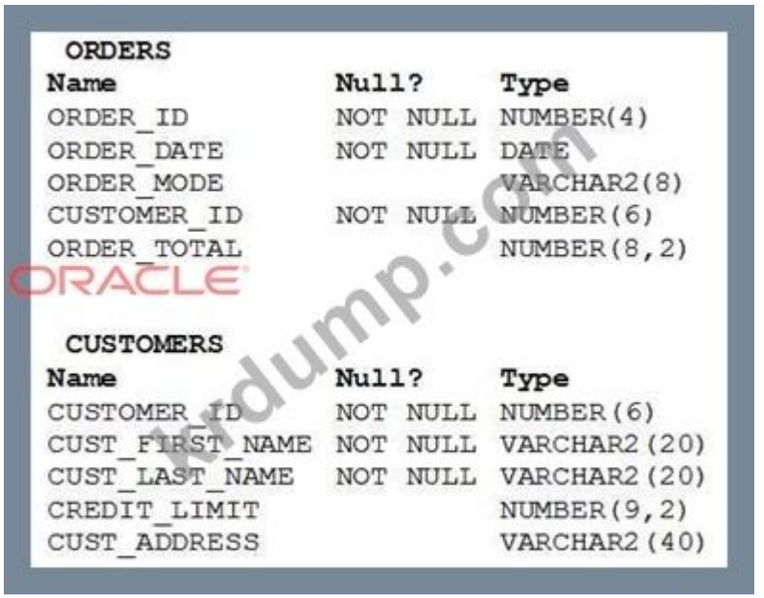
F. CURRENT_DATE□ □□ □□□□ □□ □□ □□□ □□□ □□□□□.

Answer: E,F (LEAVE A REPLY)

□□/□□:

NEW QUESTION: 49

□□□□ □□ ORDERS □ CUSTOMERS□□□□ □□□ □□□□□□.



ORDERS

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER(4)
ORDER_DATE	NOT NULL	DATE
ORDER_MODE		VARCHAR2(8)
CUSTOMER_ID	NOT NULL	NUMBER(6)
ORDER_TOTAL		NUMBER(8,2)

CUSTOMERS

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(6)
CUST_FIRST_NAME	NOT NULL	VARCHAR2(20)
CUST_LAST_NAME	NOT NULL	VARCHAR2(20)
CREDIT_LIMIT		NUMBER(9,2)
CUST_ADDRESS		VARCHAR2(40)

Roberts □□ □□ cust_last_name □□ □□ □□□ □ □ □□□□□. CUST_LAST_NAME□ Roberts □□ CREDIT_LIMIT□ 600□ □□□ ORDERS □□□□ □□ □□□□□ □□ INSERT □□ □□□□ □□□?

A. INSERT INTO □□(order_id, order_data, order_mode,
(□□ ID □□
□□□□□□
WHERE cust_last_name='□□□' AND
credit_limit=600), order_total)
□(1, '2007□ 3□ 10□', '□□', &customer_id, 1000).

B. INSERT INTO □□(order_id, order_data, order_mode,
(□□ ID □□
□□□□□□
WHERE cust_last_name='□□□' AND
credit_limit=600), order_total)
VALUES(1, '2007□ 3□ 10□', '□□', &&customer_id, 1000);

C. INSERT INTO(SELECT o.order_id, o.order_date, o.order_mode, c.customer_id, o.order_total FROM □□ o, □□ c WHERE o.customer_id = c.customer_id AND c.cust_last_name='□□□' AND c.credit_limit=600) VALUES (1, '2007□ 3□ 10□', '□□', (SELECT customer_id FROM □□ WHERE cust_last_name='Roberts' AND credit_limit=600), 1000);

D. INSERT INTO □□
VALUES(1, '2007□ 3□ 10□', '□□',
(□□ ID □□
□□□□□□
WHERE cust_last_name='□□□' AND
Credit_limit=600), 1000);

Answer: D (LEAVE A REPLY)

NEW QUESTION: 50

PROMOTIONS □□□□ □□□ □□□□□□. (□□ □□ □□ □□□□□□.)

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)

□□□□ □ □□ □□□□ □□□ □□ □□□ □□ □□□□ □□□□□.

□ □□ □□□ □□□□ □□□ □□□□□?

A. SELECT promo_category, DISTINCT promo_cost FROM □□□□

B. SELECT DISTINCT promo_cost, DISTINCT promo_category FROM □□□□;

C. SELECT DISTINCT promo_category, promo_cost FROM □□□□ ORDER BY 1

D. SELECT DISTINCT promo_cost, promo_category FROM □□□□

Answer: C (LEAVE A REPLY)

NEW QUESTION: 51

□□ □□ □□

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

□ □□□ □□□ □□□ □□□□□?

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

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

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

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

Answer: D (LEAVE A REPLY)

NEW QUESTION: 52

□□□□ □□ SALES □□□□ □□□ □□□□□□.

Table 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)

□□ □□□ SALES □□□□□ □□□□ □□ □□ ID□ □□□□□ □□□□□□□.

55000□□ □□□ 10□ □□□ □□□ □□□□□□.

```
SQL> SELECT prod_id
FROM sales
WHERE quantity_sold > 55000 AND COUNT(*)>10
GROUP BY prod_id
HAVING COUNT(*)>10;
```

□ SQL □□ □□ □□□□ □□ □□?

A. □□□□□ □□□□ □□□ □□□ □□□□□.

B. SELECT □□□ COUNT(*)□ □□□□ □□□ □□□ □□□□□.

C. COUNT(*)□ WHERE □□ □□□ HAVING □□□ □□□ □□□ □□□ □□□□□.

D. □□□□□ □□□□□ COUNT(*) □□ COUNT(prod_id)□ □□□□ □□□ □□□ □□□□ □□□ □.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 53

ORDERStable□ □□ □□□ □□□□□.

ORDER_ID	ORDER_DATE
1	<null>
2	<null>
3	01-JAN-2019
4	01-FEB-2019
5	01-MAR-2019

INVOICEStable□ □□ □□□ □□□□□.

INVOICE_ID	ORDER_ID	ORDER_DATE
1	1	<null>
2	2	01-JAN-2019
3	3	<null>
4	4	01-FEB-2019
5	5	<null>

□□ □□□ □□□□□.

```
SELECT order_id, order_date FROM orders
 MINUS
SELECT order_id, order_date FROM invoices
```

□□ □ □□ □□□□□? (3□□ □□□□□.)

A. 5 <□>

B. 4 2019□ 2□ 1□

C. 3 <□>

D. 3 2019□ 1□ 1□

E. 2 <□>

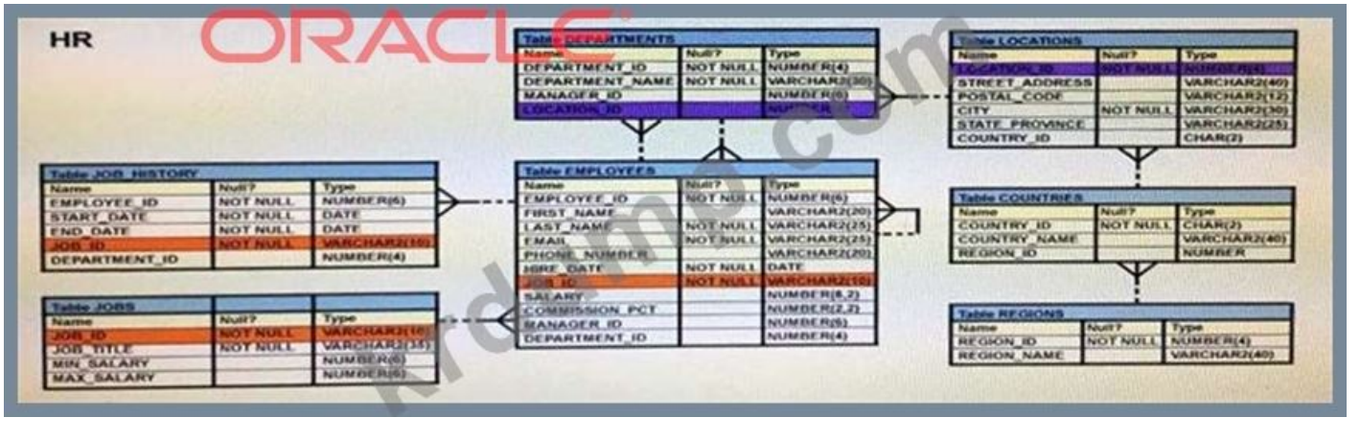
F. 2019□ 3□ 5□

G. 1 <□>

Answer: (SHOW ANSWER)

NEW QUESTION: 54

□□□□ □□ EMPLOYEES □□□□ □□ □□□ □□□□□. (2□□ □□□□□.)



□□ SQL □□ □□□□□□.

SELECT □□, □□ ID, □□

□□□□□□

ORDER BY □□ ID, □□, □□ □□;

□□□ □□ □□□□ □□ □□?

A. □□ □□ □□ □□□□□□ □□□□□.

B. SALARY □□ □□ DEPARTMENT_ID □□ □□ □□□ □□ □□□ □□ □□□□□□ □□□□□.

C. FIRST_NAME □□ □□ DEPARTMENT_ID □□ □□ □□□ □□ □□□ □□ □□□□□□ □□□ □□.

D. SALARY □□ □□ DEPARTMENT_ID □ FIRST_NAME □□ □□ □□□ □□ □□□ □□ □□□ □□□ □□□□□.

E. FIRST_NAME □□ □□ DEPARTMENT_ID □□ □□ □□□ □□ □□□ □□ □□□□□□ □□□ □□.

Answer: (SHOW ANSWER)

NEW QUESTION: 55

Exhibit□ □□ □□□□ □□ □□□□ □□□ □□□□ □□ EMP □□□□ □□□ □□□□□□. (2□ □ □□□□□□.) □□ SQL □□ □□□□□□.

ALTER TABLE EMP

DROP COLUMN □□ □□;

□□ □ □□ □□□ □□□□□?

A. □□□□ □□□□ □□ FIRST_NAME □□ □□□□□.

B. FIRST_NAME □□ CASCADE □□□ SQL □□ □□□ □□ □□ PRIMARY KEY□ □□□ □□□ □ □□□ □ □□□□.

C. □□□□ □□ □□□ □□ □□ □□□ FIRST_NAME □□ □□□□□.

D. SQL □□ SET UNUSED □□□ □□□□ FIRST_NAME □□□ □□□ □□□ □ □□.

Answer: (SHOW ANSWER)

NEW QUESTION: 56

□□□□ □□ INVOICE □□□□ □□□ □□□□ □□□□□□.

INVOICE

Name	Null?	Type
INV_NO	NOT NULL	NUMBER (3)
INV_DATE		DATE
CUST_ID		VARCHAR2 (4)
INV_AMT		NUMBER (8, 2)

INV_NO	INV_DATE	CUST_ID	INV_AMT
1	01-APR-07	A10	1000
2	01-OCT-07	B1R	2000
3	01-FEB-07		3000

□□ □ □□ SQL □□ □□□□□ □□□□□? (2□□ □□□□□.)

A. SELECT AVG(inv_date - SYSDATE), AVG(inv_amt)FROM □□;

B. SELECT AVG(inv_date)FROM □□;

C. SELECT MAX(AVG(SYSDATE -inv_date))FROM □□;

D. SELECT MAX(inv_date),MIN(cust_id)FROM □□□□;

Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 57

Books_TRANSACTIONS □□□□ □□□ □□□□□.

Name	Null?	Type
TRANSACTION_ID	NOT NULL	VARCHAR2 (6)
TRANSACTION_TYPE		VARCHAR2 (3)
BORROWED_DATE		DATE
DUE_DATE		DATE
BOOK_ID		VARCHAR2 (6)
MEMBER_ID		VARCHAR2 (6)

SQl □□ □□□□□.

SQL> SELECT * FROM books_transactions WHERE brrowed_data<SYSDATE AND transaction_type='RM' □□ MEMBER_ID IN

('A101;' A102 ');

□□□ □□ □□□□ □□ □□?

A. RM□ TRANSACTION-TYPE□□ □□ □□ □□□ □□ □□ A101, A102□ □□ □□□ □□□□□.

B. RM□ TRANSACTION_TYPE□□ □□ □□ □□ □□□ □□□ A101 □□ A102 □□□ □□□ □□ □□□.

C. RM□ TRANSACTION_TYPE□□ □□ □□ □□□ □□ □□□ □□□□ □□□ □□□□□.

D. RM□ TRANSACTION_TYPE□□, MEMBER_ID□ A101, A102□ □□ □□ □□□ □□ □□□ □□ □ □□□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 58
□□□□□ □□□□ □□□□ □□□□□□.
CREATE USER □□ IDENTIFIED BY pwfin;
CREATE USER □ □□□ IDENTIETED BY pwmgr;
CREATE USER □. □□□ pwclerk□ □□ □□□;
GRANT CREATE SESSON 20 □□, □□□□ □□;
GRANT SELECT ON □□. Emp□ GRANT OPTION□□ □□□ □□□□□.
CONNECT □□□□/pwfin
GRANT SELECT ON □□. emp fin_ _clerk□□;
□□ □□ □□□□□?
A. □□□ FIN CLERK□ □□□ FIN MANAGER□□ SCORT, ENP□ □□ SELECT □□□ □□□ □ □ □□□.
B. □□□ FINANCE□ □□□ FIN MANAGER□□ CREATE SESSION□ □□□ □ □□□□.
C. □□□ FINANCE□ □□□□ SCOTT□□ SELECT□ □□□□ □□□□□. □□□ FIN _ CLERK□ EMP
D. □□□ FINANCE□ SCOTT.ENP□ ALL□ FIN MANAGER□ □□□ □ □□□□.
E. SCOTT□□ SELECT□ □□□□□. □□□ FINANCE□ EMP□ □□□ FIN_ CLERK□ □□□ □□ □□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 59
□□□□ □□ EMPLOYEES □ DEPARTMENT□□□□ □□□ □□□□□□.

EMPLOYEES

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(10,2)
COMMISSION		NUMBER(6,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

DEPARTMENTS

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

□□□ □□ EMPLOYEEStable□ □□□□□□□ □□□.
□□□ □□ □□□(□□ 2900 □ 2700)□□ □□□ □□□ □□□□□□□.

- □ □□□ Department_id□ London□ □□□□ department_id□ □□□□□.
- (□□_ID 2100).

location_id2100□ □□ □□□ □□ □□ □□□ 1.1□□ □□□□□.

- location_id2100□ □□ □□□□ □□□ □□ □□□□ 1.5□□ □□□□□.
- □□.

□□ □□□ □□□□□.

```
QL> 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;
```

□□□ □□□□□?

A. UPDATE □□□ □□ □(SALARY, COMMISSION)□ □□ □□□ □ □□□□ □□□ □□□□□.

B. □□□□ UPDATE □□ □□ □□□ □□ □ □□ □□□ □□□ □□□□□.

C. □□□□□ □□□□ □□□ □□□□□ □□□□□.

D. □□□□□ □□□□□ □□□ □□□□□ □□□□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 60

SAVEPOINT□ □□ □□□□ □□ □□?

A. □□□ □□□□□□ □□□ SAVEPOINT□ □□□ □ □□□□.

B. SAVEPOINT□ □□□□ CREATE INDEX □□ □□ □□□ □ □□□□.

C. SAVEPOINT□ COMMIT□ □□□□ □□□□.

D. SAVEPOINT□ □□□□ DELETE □□ □□ □□□ □ □□□□.

E. SAVEPOINT□ □□□□ TRUNCATE □□ □□ □□□ □ □□□□.

Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 61

SALES □□□□ □□ □□□ □□□□□.

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(10)
CUSTOMER_ID	NOT NULL	NUMBER(10)
TIME_ID	NOT NULL	DATE
CHANNEL_ID	NOT NULL	NUMBER(5)
PROMO_ID	NOT NULL	NUMBER(5)
QUANTITY_SOLD	NOT NULL	NUMBER(10,2)
PRICE		NUMBER(10,2)
AMOUNT_SOLD	NOT NULL	NUMBER(10,2)

SALES □□□□□ 55,000□□ □□ □□□□.

□□ □□□□ □□□□□□.

```
CREATE TABLE sales1 (prod_id, cust_id, quantity_sold, price)
AS
SELECT product_id, customer_id, quantity_sold, price
  FROM sales
 WHERE 1 = 1;
```

□□ □ □□ □□□ □□□□□?

A. SALES1□ □ □□ □□□□□.

B. SALES1□□ SALES □□□□ □□□ □□ □□□ □□ □□□ □□ □□ PRIMARY KEY □ UNIQUE □□ □□□ □□□□.

C. SALES1□□ SALES □□□□ □□□ □□ □□□ □□ □□□ □□ □□ NOT NULL □□ □□□ □ □□□.

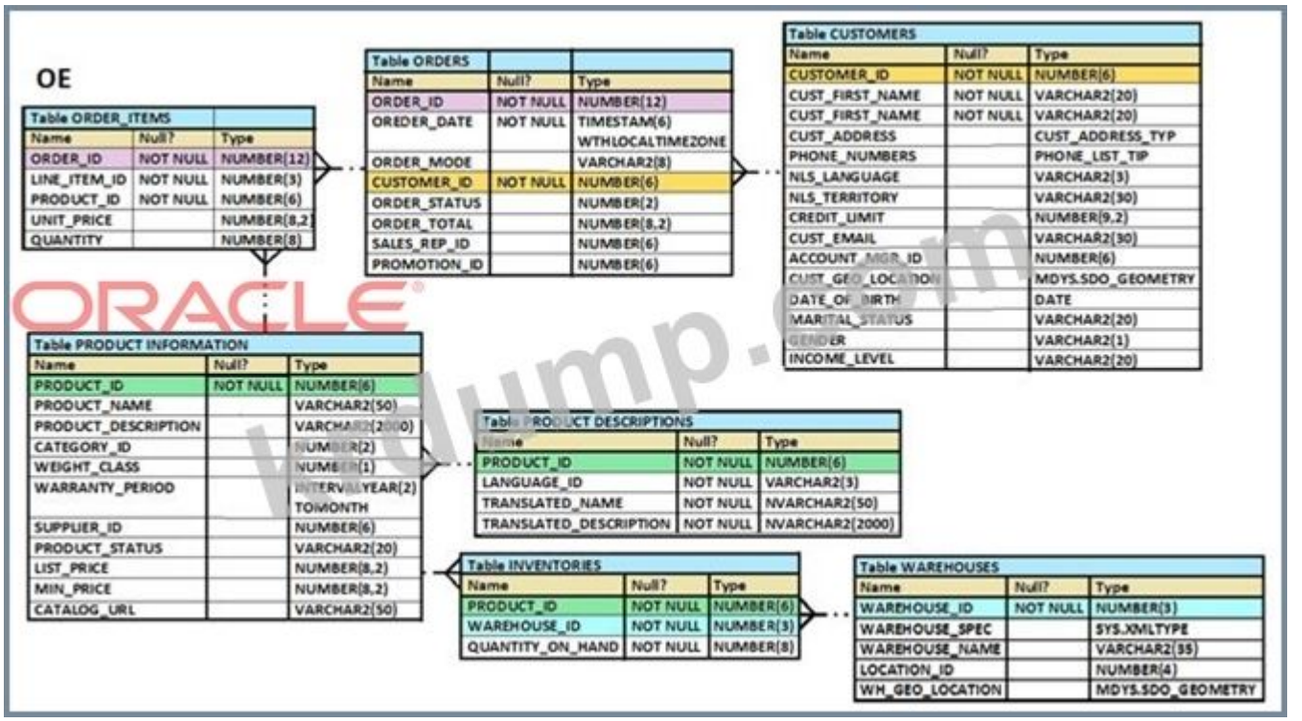
D. SALES1□ 55,000□□ □□□ □□□□□.

E. SALES1□ 1□□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 62

□□□□ □□ PRODUCT_INFORMATION □□□□ □□□□□ □□□□□□. (2□□ □□□□□.)



□ SQL □□ □□□□□□.

SELECT TO_CHAR(□□_□□,'$9,999')

□□ □□□□;

□□□ □□ □ □□ □□ □ □□ □□? (2□□ □□□□□.)

A. LIST_PRICE □□ □ 1123.90□ □□□ □□ $1,124□ □□□□□.

B. LIST_PRICE □□ □ 1123.90□ □□□ □□ $1,123□ □□□□□.
C. LIST_PRICE □□ □ 11235.90□ □□□ □□ $1,123□ □□□□□.
D. LIST_PRICE □□ □ 11235.90□ □□□ □□ #######□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 63
□□□ □□□ □□ □□□□ □□ □□?
A. □□□ □□ □□□ □□□ □□□ □□□ □ □□□□.
B. PUBLIC□ □□□□□□ □□□ □ □□□□.
C. PUBLIC□ □□□□□□□ □□ □□□□□ □□□ □□ □□ □□□ □□□.
D. □□□ □□ □□□□ □□□ □□□□□.
E. □□□ □□□ □□ □□ □□□□□□□ □□ □□□ □□□□□.
F. □□ □□□ SYS □□□□ □□□□□.
G. □□□□ □□□□□ □□□□ □□ □□□ □□ □□ □□ □□□ □□□□.
Answer: A,C,G (LEAVE A REPLY)

NEW QUESTION: 64
□□ □□□ □□ □□□□ □□ □□? (3□□ □□□□□.)
A. □□□ □□□□ □□□ □ □□□□.
B. □□ □□ □□□ □□□ □□□ □ □□□□.
C. GROUPBY□□ □□ SQL□□□□ □□□ □ □□.
D. SQL □□ SELECT □□□ □□□ □□□ □□□ □ □□□□.
E. SQL□□ SELECT□□□ □□ □ □□□ □□ □□□ □ □□□□.
Answer: A,B,E (LEAVE A REPLY)
□□/□□:
□□:
https://www.safaribooksonline.com/library/view/mastering-oracle-sql/0596006322/ch04.html

NEW QUESTION: 65
□□□□ □□ ORDERStable□ □□□□□□.

ORDERS

Name	Null?	Typ
ORDER ID	NOT NULL	NUMBER(4)
ORDER DATE		DATE
CUSTOMER ID		NUMBER(3)
ORDER TOTAL		NUMBER(7,2)

ORDERStable□□ □□□□ □□□□ □□□ □□ □□□□ □□ ID□ □□□□ □□□□.
CUSTOMER_ID □□ NOTNULL □□ □□□ □□□□ □□□□ □□□□□?
A. ALTER TABLE □□
MODIFY CONSTRAINT orders_cust_id_nn NOT NULL(customer_id);

B. ALTER TABLE □□
ADD CONSTRAINT orders_cust_id_nn NOT NULL(customer_id);
C. ALTER TABLE □□
MODIFY customer_id CONSTRAINT orders_cust_nn NOT NULL(customer_id);
D. ALTER TABLE □□
ADD customer_id NUMBER(6)CONSTRAINT orders_cust_id_nn NOT NULL;
Answer: (SHOW ANSWER)

NEW QUESTION: 66
ORDER by □□ □□ □□□ □□ □□□□ □□ □□?
A. ORDER by □□□□ SELECT □□□ □□□ □□ □□□ □ □□□□.
B. □□ □□□□ □□ □□□□□ □□□□□.
C. □□ □□ □□□□ □□ □□ □□□□□□ □□□□□.
D. NULL□ □□ □□□ □□□□ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 67
□□ □□□□□ CUST_CITY □□□ CUST_FIRST_NAME 'Abigail'□ □□ 'Paris' □□ □□□□□.
□□ □□□ □□□□□.

```
SQL> SELECT INITCAP(cust_first_name || ' ' ||
           UPPER(SUBSTR(cust_city,-LENGTH(cust_city),2)))
     FROM customers
     WHERE cust_first_name = 'Abigail';
```

□□□ □□□ □□□□□?
A. □□□□ IS
B. □□□□ PA
C. □□ □□□
D. □□□□ □
Answer: D (LEAVE A REPLY)

NEW QUESTION: 68
□□□□ □□ CUSTOMERStable□ □□□ □□□□□□.
CUSTOMERS □□□□ □□□□ □□ □□ □□□ □□□□ □□□□ □□□□ □□□.
□□ □□□□ 15%. □□ □□□ □□□□ □□ □□□ "□□□ □ □□" □□□□ □□□□□ □□□.
□□ SQL □□ □□□ □□□ □□□□□?

Table CUSTOMERS

Name	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_FIRST_NAME	NOT NULL	VARCHAR2 (20)
CUST_LAST_NAME	NOT NULL	VARCHAR2 (40)
CUST_GENDER	NOT NULL	CHAR (1)
CUST_YEAR_OF_BIRTH	NOT NULL	NUMBER (4)
CUST_MARITAL_STATUS		VARCHAR2 (20)
CUST_STREET_ADDRESS	NOT NULL	VARCHAR2 (40)
CUST_POSTAL_CODE	NOT NULL	VARCHAR2 (10)
CUST_CITY	NOT NULL	VARCHAR2 (30)
CUST_STATE_PROVINCE	NOT NULL	VARCHAR2 (40)
COUNTRY_ID	NOT NULL	NUMBER
CUST_INCOME_LEVEL		VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2 (30)

SELECT NVL (TO CHAR(cust_credit_limit * .15), '□□□ □ □□') "NEW CREDIT"

A. □□□□□□;

SELECT NVL(cust_credit_limit * .15), '□□□ □ □□') "NEW CREDIT"

B. □□□□□□;

SELECT NVL(cust_credit_limit), '□□□ □ □□') "NEW CREDIT"

C. □□□□□□;

D. □□□□□□;

SELECT TO_CHAR(NVL(cust_credit_limit * .15), '□□□ □ □□') "NEW CREDIT"

Answer: D (LEAVE A REPLY)

NEW QUESTION: 69

Oracle □□□□□□□ □□□ SQL □□□ □□□□ □□ □□ □□□ □ □□ □ □□ □□ □□□ □□ □□□? (3□□ □□□□□.)

A. □□ □□ □ □□

B. □□ □□□

C. □□

D. □ □□□

E. □□

Answer: A,B,E (LEAVE A REPLY)

NEW QUESTION: 70

Oracle □□□□ □□ □□□□ □□ □□?

A. □ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□ □ □□□□.
B. □□□□ □□ □□□□ □□□ □ □□□□.
C. SEQUENCE□ □□□□ □□ □ □□□□.
D. PL /SQL □□□□ □□ □□□□ □□ □ □□□□.
E. □□ □□□□ PUBLIC □□□□ □□□ □ □□□□.
Answer: A,B,D (LEAVE A REPLY)

NEW QUESTION: 71
PROMOTIONS □□□□ □□□ □□□□□□. (□□ □□ □□ □□□□□□.)

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)

□□□□ □ □□ □□□□ □□□ □□ □□□ □□ □□□□ □□□□□.
□ □□ □□□ □□□□ □□□ □□□□□?
A. SELECT DISTINCT promo_cost, DISTINCT promo_category FROM □□□□;
B. SELECT promo_category, DISTINCT promo_cost FROM □□□□
C. SELECT DISTINCT promo_category, promo_cost FROM □□□□ ORDER BY 1
D. SELECT DISTINCT promo_cost, promo_category FROM □□□□
Answer: C (LEAVE A REPLY)

NEW QUESTION: 72
□□ □□□ □□□□□.

```
SELECT employee_id, first_name, salary
     FROM employees
WHERE hire_date > '&1';
```

□ □□□ □□□ □ □□ □□ □□ □□ □□□□ □□□□ □□□ □□□ □□ □ □□ □□□ □□□ □ □□□?
A. □□□ □□□□ □□ UNDEFINE □□□ □□□□□.
B. □□□ □□□□□ □□□□ □□ □ □□□□□ □□ □□ □□□□□.
C. □□□ □□□□ □□ SET VERIFY OFF □□□ □□□□□.
D. □□□ □□□□ □□ SET VERIFY ON □□□ □□□□□.
E. □□□ □□□□ □□ DEFINE □□□ □□□□□□.
F. □□□□ '&1'□ '&&1'□ □□□□.
Answer: E,F (LEAVE A REPLY)

NEW QUESTION: 73
PROMOTIONS □□□□ □□□ □□□□□□. (□□ □□ □□ □□□□□□.)

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)

□□□□ □ □□ □□□□ □□□ □□ □□□ □□ □□□□ □□□□□.
□ □□ □□□ □□□□ □□□ □□□□□?

A. SELECT promo_category, DISTINCT promo_cost FROM □□□□
B. SELECT DISTINCT promo_cost, DISTINCT promo_category FROM □□□□;
C. SELECT DISTINCT promo_category, promo_cost FROM □□□□ ORDER BY 1
D. SELECT DISTINCT promo_cost, promo_category FROM □□□□
Answer: (SHOW ANSWER)

NEW QUESTION: 74
ORDER by □□ □□ □□□ □□ □□□□ □□ □□?
A. ORDER by □□□□ SELECT □□□ □□□ □□ □□□ □ □□□□.
B. □□ □□□□ □□ □□□□□ □□□□□.
C. NULL□ □□ □□□ □□□□ □□□□.
D. □□ □□ □□□□ □□ □□ □□□□□□ □□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 75
EMPLOYEES □□□□ □□ □□□ □□□□□.

Name	NULL?	Type
EMP_NO	NOT NULL	NUMBER(5)
LAST_NAME		VARCHAR2(10)
DEPT_NO	NOT NULL	NUMBER(5)
SALARY		NUMBER(6,2)

□□□ □□ □□□ □□□ □□□□□.
SELECT dept_no AS department_id, MAX(□□) max_sal□
□□□□□□
□□ >10000
GROUP BY □□ ID
ORDER BY max_sal;
□□ □□ □□□ □□□□□?

A. □□
B. □□
C. □□
D. □□□ □□

Answer: D (LEAVE A REPLY)

NEW QUESTION: 76

□□□□ □□ EMPLOYEES □ DEPARTMENT□□□□ □□□ □□□□□□.

```
MPLOYEES
lame                 Null?          Type
-----------------    -----          ------
MPLOYEE_ID           NOT NULL       NUMBER(6)
IRST_NAME                           VARCHAR2(20)
AST_NAME             NOT NULL       VARCHAR2(25)
IRE_DATE             NOT NULL       DATE
OB_ID                NOT NULL       VARCHAR2(10)
ALARY                               NUMBER(10,2)
OMMISSION                           NUMBER(6,2)
ANAGER_ID                           NUMBER(6)
EPARTMENT_ID                        NUMBER(4)

EPARTMENTS
lame                 Null?          Type
-----------------    -----          ------
EPARTMENT_ID         NOT NULL       NUMBER(4)
EPARTMENT_NAME       NOT NULL       VARCHAR2(30)
ANAGER_ID                           NUMBER(6)
OCATION_ID                          NUMBER(4)
```

□□□ □□ EMPLOYEEStable□ □□□□□□□ □□□.

* □□□ □□ □□□(□□ 2900 □ 2700)□□ □□□ □□□ □□□□□□□.

* □ □□□ Department_id□ London(location_id)□ □□□□ Department_id□ □□□□□.

2100).

* location_id2100□ □□ □□□ □□ □□ □□□ 1.1□□ □□□□□.

* location_id2100□ □□ □□□□ □□ □□ □□□□ 1.5□□ □□□□□.

□□ □□□ □□□□□.

```
    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;
```

□□□ □□□□□?

A. □□□□□ □□□□ □□□ □□□□□ □□□□□.

B. □□□□□ □□□□□ □□□ □□□□□ □□□□ □□□□.

C. UPDATE □□□ □□ □(SALARY, COMMISSION)□ □□ □□□ □ □□□□ □□□ □□□□□.

D. □□□□ UPDATE □□ □□ □□□ □□ □ □□ □□□ □□□ □□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 77
ORD_ITEMS □□□□ □□□□ □□□□□.

ORD_ID	ITEN_NO	QTY
1	111	10
1	222	20
1	333	30
2	333	30
2	444	40
3	111	40

□□ □□□ □□□□□.

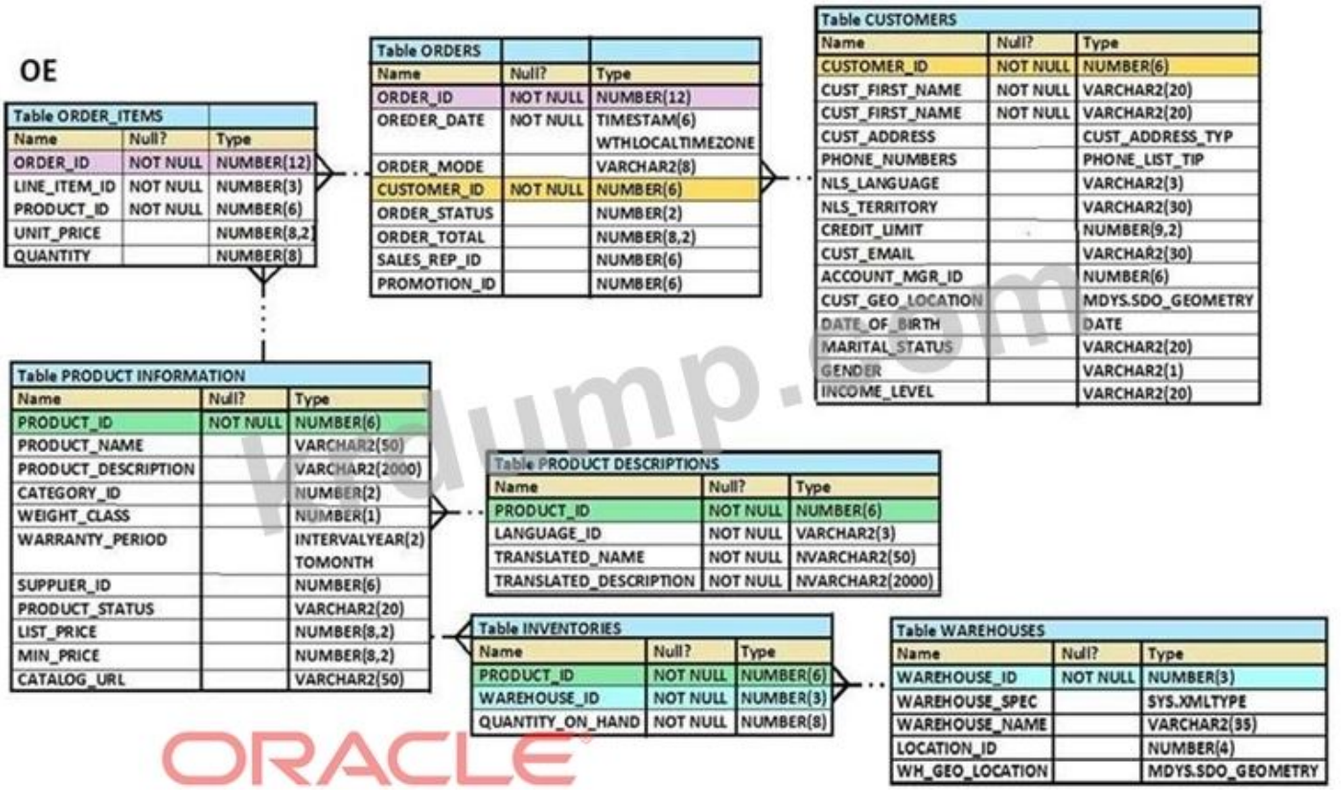
```
SQL>SELECT item_no, AVG(qty)
FROM ord_items
HAVING AVG(qty) > MIN(qty) * 2
GROUP BY item_no;
```

□□□ □□ □□□□ □□ □□?
A. □□ □□□ □□ □□□ □□ □□□ 2□ □□□ □□ □□ □□□ □□ □□□ □□ □□ □□□ □□ □□ □□□□□.
B. HAVING □□□ □□□□ □□ □□ □□□ SELECT □□□ □□□□ □□□ □□□ □□□□□.
C. □□ □□□ □□□□ □□ □□ □□□ □□ □□ □□□ □ □ □□□ □□ □□ □□□ □□ □□□ □□ □□□□□.
D. GROUP BY□ □□ HAVING□□ □□□□ □□□ □□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 78
SELECT □□ WHERE □ HAVING □□ □□ □□□□ □□ □□?
(□□□□ □□ □□ □□□□□□.)
A. WHERE □□ □□□□ □□ □□□□ □□ □ □□□ □ □□□□.

B. HAVING □□ □□□□ □□ □□ □ □□ □□□ SELECT □□□ □□□□□ □□□.
C. WHERE □ HAVING □□ □□□□ □□ □□ □□□□ □□□□ □□□ □□□ □□□ □ □□□□.
D. WHERE □□ □□□□ □□ □□□□ □□□ □□ □□□ □ □□□□.
E. HAVING □□ □□ □□□□ □□ □□□ □□ □□□ □ □□□□.
Answer: D,E (LEAVE A REPLY)

NEW QUESTION: 79
□□□□ □□ PRODUCT_INFORMATION □□□□ □□□ □□□□□□. (□□ □□ □□ □□□□□.)



PRODUCT_ID □□ □□ □□□□.
□□ □□□ □□□□ □□□□ □□□□□.
SQL > □□□ □□ upper_name_idx
ON product_information(UPPER(product_name));
PRODUCT_INFORMATION □□□□ □□ □□□□ □□□□.
□□ □□□ UPPER_NAME_IDX □□□□ □□□□□?
A. SELECT product_id, UPPER(product_name)
FROM □□ □□
WHERE UPPER(product_name) = 'LASERPRO' □□ list_price > 1000;
B. SELECT UPPER(□□□)
□□ □□□□;
C. □□ ID □□
FROM □□ □□
WHERE UPPER(□□□) IN ('LASERPRO', 'CABLE');
D. SELECT UPPER(□□□)

FROM □□ □□
□□ □□ ID = 2254;
Answer: (SHOW ANSWER)

NEW QUESTION: 80
DEPARTMENT_DETAILS □
□□□ □□□ □□:

```
SQL>CREATE TABLE DEPARTMENT_DETAILS
(DEPARTMENT_ID NUMBER PRIMARY KEY,
DEPARTMENT_NAME       VARCHAR2(50),
HOD                   VARCHAR2(50));
SQL>CREATE TABLE COURSE_DETAILS
(COURSE_ID            NUMBER PRIMARY KEY,
COURSE_NAME           VARCHAR2(50),
DEPARTMENT_ID     NUMBER REFERENCES DEPARTMENT_DETAILS
                                    (DEPARTMENT_ID));
```


□□ □□ ID□ □□□ □□□ □□□□ □□ □□ ID□ □□□□□ □□□ □□ □□ □□ ID□ □□ □ □□□ □□□□□ □□□.
□□ SQL □□ □□□□ □□□?
A. SELECT c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id=d. Department_id)
B. SELECT c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id<>d. Department_id)
C. SELECT c.course_id, d.department_id FROM Course_details c RIGHT OUTER JOIN .department_details d ON (c.depatrment_id=d.department_id)
D. SELECT Course_id, Department_id, FROM Department_details d RIGHT OUTER JOIN Course_details c USING (department_id)
Answer: A (LEAVE A REPLY)

NEW QUESTION: 81
□□ □□□ □□□□ □□□ □ □□ □□□ □□□□.
□□ □ □□ □□□□ □□□□ □ □□ HCM.EMPLOYEE_RECORDS □□□□ □□ EMP□□ □□□ □ □□□□ □□□□ □□□□□?
A. hcm.employee_records□ □□ □□ SYNONIM emp □□;
B. hcm.employee_records□ □□ □□□ SYNONIM emp □□;
C. hcm.employee_records□ □□ SYNONIM emp □□;
D. hcm.employee_records□ □□ SYNONIM PUBLIC.emp □□;
E. hcm.employee_records□ □□ SYNONIM SYS.emp □□;
Answer: (SHOW ANSWER)
□□ SYNONYM emp_table □□

□□: https://docs.oracle.com/database/121/SQLRF/statements_7001.htm#SQLRF01401

NEW QUESTION: 82
non-equijoin □ □□□ □□ □□□□ □□ □□ □□?
A. Oracle □□ □□□ SQL:1999 □□ ANSI □□ □□□□ □□□ □□□□.
B. BETWEEN □□□ >= □ <= □□□ □□□□ □□□ □□ □ □ □□□□□.
C. □□□ □□ □□□ □□□ □□□ □□□□.
D. □□□ □□□ □□□ □□□□ □ □□□□.
E. BETWEEN □□□ □□ >= □ <= □□□ □□□□ □□□ □□□ □□□□□.
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 83
DEPARTMENT_DETAILS □ COURSE_DETAILS□ □□□□ □ □□□ □□□ □□□□□.

```
SQL>CREATE TABLE DEPARTMENT_DETAILS
(DEPARTMENT_ID NUMBER PRIMARY KEY,
DEPARTMENT_NAME       VARCHAR2(50),
HOD                   VARCHAR2(50));
SQL>CREATE TABLE COURSE_DETAILS
(COURSE_ID            NUMBER PRIMARY KEY,
COURSE_NAME           VARCHAR2(50),
DEPARTMENT_ID      NUMBER REFERENCES DEPARTMENT_DETAILS
                                 (DEPARTMENT_ID));
```


□□ □□ ID□ □□□ □□□ □□□□ □□ □□ ID□ □□□□□ □□□ □□ □□ □□ ID□ □□ □ □□□ □□□□□ □□□.
□□ SQL □□ □□□□ □□□?
A. SELECT c.course_id, d.department_id FROM Course_details c RIGHT OUTER JOIN .department_details d ON (c.depatrment_id=d.department_id)
B. SELECT c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id<>d. Department_id)
C. SELECT c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id=d. Department_id)
D. SELECT Course_id, Department_id, FROM Department_details d RIGHT OUTER JOIN Course_details c USING (department_id)
Answer: (SHOW ANSWER)

NEW QUESTION: 84
□□ □ □□□ □□□□□ □□□□□?
A. SELECT SYSDATE "INTERRVAL '1' DAY FROM DUAL;
B. DUAL□□ INTERVAL '1' DAY + INTERVAL '1' MONTH □□;
C. SELECT INTERVAL '1' DAY - DUAL□□ SYSDATE;
D. SELECT SYSTIMESTAMP + INTERVAL '1' DAY FROM DUAL;
E. □□□□ □□ '1'□ - □□ '1'□□ □□□□□□.

Answer: D,E (LEAVE A REPLY)

NEW QUESTION: 85
□□□□ □□ INVOICE □□□□ □□□ □□□□ □□□□□□.

INVOICE

Name	Null?	Type
INV_NO	NOT NULL	NUMBER (3)
INV_DATE		DATE
CUST_ID		VARCHAR2 (4)
INV_AMT		NUMBER (8, 2)

INV_NO	INV_DATE	CUST_ID	INV_AMT
1	01-APR-07	A10	1000
2	01-OCT-07	B1R	2000
3	01-FEB-07		3000

□□ □ □□ SQL □□ □□□□□ □□□□□? (2□□ □□□□□.)
A. AVG □□(inv_date -SYSDATE), AVG(inv_amt)
□□□□
B. SELECT MAX(AVG(SYSDATE -inv_date))
□□□□
C. SELECT MAX(inv_date), MIN(cust_id)
□□□□
D. AVG □□(inv_date)
□□□□
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 86
□□□□ □ □□ □□□□ □□□ □ □□ □ □□ □□□ □□□□□? (3□□ □□□□□.)
A. □□
B. □□□
C. □□
D. □□□□
E. □□
F. □□

G. □□
Answer: D,E,G (LEAVE A REPLY)

NEW QUESTION: 87
□□ □□□ □□□□□.
```
TRUNCATE TABLE depts;
```
□□ □□ □□□□□?
A. □□□□ □□□ □□□□ □□□□□.
B. □□□□ □□□ □□□ □□ □□□ □□□□□.
C. □□□ □□ □□□□ □□□ □□ □□□□□.
D. □□□□ □□□ □□ □□□□ □□□□□.
E. ROLLBACK □□ □□□□ □□□ □□□□ □□□ □ □□□□.
F. FLASHBACK TABLE □□ □□□□ □□□ □□□□ □□□ □ □□□□.
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 88
□□ □ □□ □□□□ □□ □□□ □□ □□□□ □□□□□□.
SQL> CREATE TABLE store(store_id NUMBER(4) CONSTRAINT store_id_pk PRIMARY KEY, store_name VARCHAR2(12), store_address VARCHAR2(20), start_date DATE); SQL> CREATE TABLE sales(sales_id NUMBER(4) CONSTRAINT sales_id_pk PRIMARY KEY, item_id NUMBER(4), □ □ NUMBER(10), sales_date DATE, store_id NUMBER(4), CONSTRAINT store_id_fk FOREIGN KEY(store_id) □□(store_id) REFER ; □□ □□□□ □□□□□□.
SQL> □□□□□ DELETE
WHERE store_id=900;
□□□ □□ □□ □□□ □□ □□□□ □□□□□.
ORA-02292: □□□ □□ □□(HR.STORE_ID_FK) □□
□□□□ □□□□□ □□□□□ □□ □ □□ □□□ □□□□□?
A. SALES □□□□□ STORE_ID = 900□ □□ □□□ □□ STORES □□□□□ □□ □□□□□.
B. on DELETE CASCADE □□□ □□□□ SALES_ID □□ SALES □□□□ □□ □□ □□□□□.
C. STORES □□□□ □□ □□ □□□□□□□.
D. SALES □□□□□ FOREIGN KEY□ □□□□□ □□ □□ □□□□□.
E. DELETE □□ □□ CASCADE □□□□ □□□□□.
Answer: A,B,D (LEAVE A REPLY)

NEW QUESTION: 89
Oracle SQL □□ □□□□ □□□ □ □□ □ □□ □□□ □□□□□? (2□□ □□□□□.)
A. □□ □□□□□□ □□□□ □□□□ □□
B. □□□□□□ □□□□□ □□
C. □□ □□□□□□□ □□□□□ □□□ □□
D. □□□□□□ □□□□ □□
E. □□□□ □□ □□(OS) □□ □□

Answer: A,C (LEAVE A REPLY)

http://www.techonthenet.com/oracle/password.php

https://docs.oracle.com/cd/B28359_01/server.111/b28324/tdpii_distdbs.htm

NEW QUESTION: 90

□□□ □□ □ □□ □□ □ □□ □□? (3□□ □□□□□.)

A. □□□ ALTER ROLE □□ □□□□ □□□ □□□□□.

B. □□□ □□□ □□ □□□□□ □□□ □ □□□□.

C. □□ □□□□□ □□ □□□ □□□ □ □□□□.

D. □□□ ALTER ROLE □□ □□□□ □□□ □□□□□.

E. □□□ □□□□□□ □□□ □ □□ □□ □□□ □□□ □□□□□.

F. □□□ GRANT□□ □□□□ □□□ □□□□□.

G. ALTER USER□□ □□□□ □□□□□ □□□ □□□□□.

Answer: B,C,F (LEAVE A REPLY)

GRANT □□ □□□□ □□□ □□ □ □□□ □□□□□.

□□:

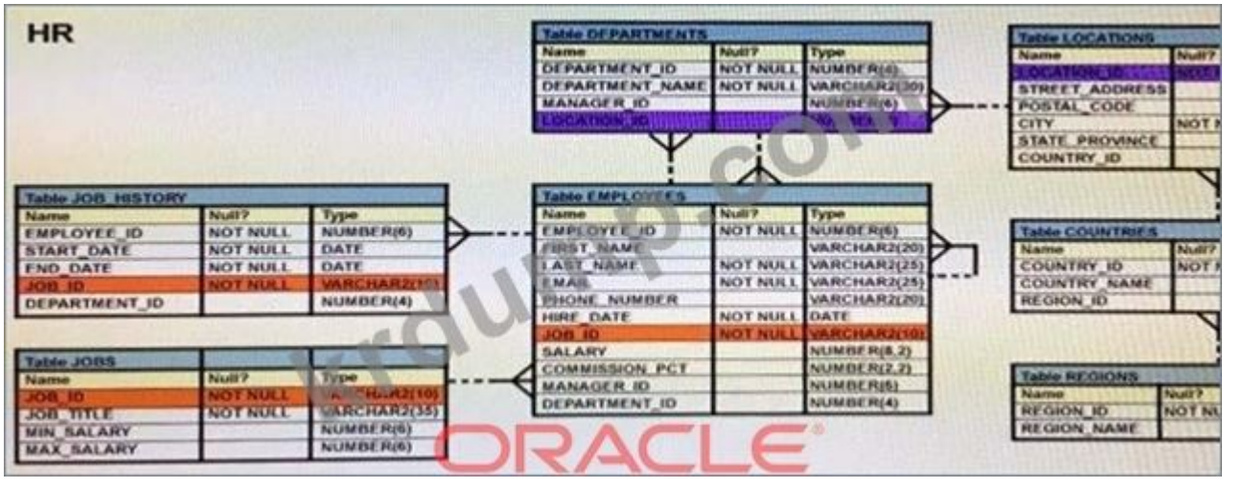
http://archive.dnnsoftware.com/docs/85/administrators/security/roles/assign-multiple-users-to-role.html

https://www.dnnsoftware.com/docs/administrators/user-accounts/assign-user-to-multiple-roles.html

https://www.ibm.com/support/knowledgecenter/en/SSGU8G_11.70.0/com.ibm.sqls.doc/ids_sqs_0828.htm

NEW QUESTION: 91

□□□□ □□ DEPARTMENTSand EMPLOYEES□□□□ □□ □□□ □□□□□.



EMPLOYEE_ID, FIRST_NAME □ DEPARTMENT NAME□ □□ □□ □□□ □□□ □□, □□ SQL □ □ □□□□□□□.

SELECT □□ ID, □□, □□ □□

□□□□□□

NATURAL JOIN □□;

□□ SQL□□ □□□ □ □□□ □□□ □□ □□□. □ □□□ □□□□□?

A. SELECT □□ □ □□□ □□ □□□ □□□□ □□□□.

B. NATURAL JOIN□□ USING□□ □□□□.

C. DEPARTMENTStable□ FROM□□ EMPLOYEEStable□□ □□ □□□□ □□□□.
D. EMPLOYEES □ DEPARTMENTStables□□ □□□ □ □□ □ □□□ □□□ □□ □ □□□ □□ □ □□□.
Answer: (SHOW ANSWER)
□□/□□:
□□:
□□ □□□ □ □□□□□ □□□ □□ □□□ □□□□□. EMPLOYEES □ DEPARTMENTS □□□□ □ □□□ □ □□ □□ □□□□(Department_ID □ Manager_ID).

NEW QUESTION: 92
□□□□ □□ ORDERStable□ □□□□□□.
□□
□□? □□
□□
□□ ID□ NULL□ □□□□(4)
□□ □□ □□
□□ ID □□(3)
□ □□ □(7,2)
ORDERStable□□ □□□□ □□□□ □□□ □□ □□□□ □□ ID□ □□□□ □□□□.
CUSTOMER_ID □□ NOTNULL □□ □□□ □□□□ □□□□ □□□□□?
A. ALTER TABLE □□
MODIFY CONSTRAINT orders_cust_id_nn NOT NULL(customer_id);
B. ALTER TABLE □□
ADD CONSTRAINT orders_cust_id_nn NOT NULL(customer_id);
C. ALTER TABLE □□
ADD customer_id NUMBER(6)CONSTRAINT orders_cust_id_nn NOT NULL;
D. ALTER TABLE □□
MODIFY customer_id CONSTRAINT orders_cust_nn NOT NULL(customer_id);
Answer: (SHOW ANSWER)

NEW QUESTION: 93
□□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. □□ □□□ □□ □□ □□□ □□□□ □□ □□ □□□□ □□□□□ □□□.

B. SELECT □□ WHERE □□ □□ □□□□ 3□□□□□ □□□ □ □□.
C. □ □□□ □ □□ □□ □□□ □□□ □ □□□□.
D. □□ □□□ 0□□ □□ □□□□ □□ □□ □□ □□□□□ □□□ □□ NULL□□□.
E. □□ □□□ □□□□ □□ □□□ □□□ □□ □□□ □□□□ □□□□ □ □□□□.
Answer: D,E (LEAVE A REPLY)

NEW QUESTION: 94
DEPARTMENT_DETAILS □ COURSE_DETAILS□ □□□□ □ □□□ □□□ □□□□□.

```
SQL>CREATE TABLE DEPARTMENT_DETAILS
(DEPARTMENT_ID NUMBER PRIMARY KEY,
DEPARTMENT_NAME       VARCHAR2(50),
HOD                   VARCHAR2(50));
SQL>CREATE TABLE COURSE_DETAILS
(COURSE_ID            NUMBER PRIMARY KEY,
COURSE_NAME           VARCHAR2(50),
DEPARTMENT_ID     NUMBER REFERENCES DEPARTMENT_DETAILS
                                    (DEPARTMENT_ID));
```

□□ □□ ID□ □□□ □□□ □□□□ □□ □□ ID□ □□□□□ □□□ □□ □□ □□ ID□ □□ □ □□□ □□□□□ □□□.
□□ SQL □□ □□□□ □□□?
A. SELECT c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id<>d. Department_id)
B. SELECT c.course_id, d.department_id FROM Course_details c RIGHT OUTER JOIN .department_details d ON (c.depatrment_id=d.department_id)
C. SELECT c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id=d. Department_id)
D. SELECT Course_id, Department_id, FROM Department_details d RIGHT OUTER JOIN Course_details c USING (department_id)
Answer: (SHOW ANSWER)

NEW QUESTION: 95
□□ □□□□ □□□□□□.
SELECT 1 AS id, ' John' AS □□
□□□□
□□ □□
SELECT 1 , 'John' AS □□
□□□□
1□ □□
□□ □ □□□ □□□□□?
A. 2□

B. 0□
C. □□
D. 1□
Answer: (SHOW ANSWER)

NEW QUESTION: 96
□□□□ □□ PODUCT_INFORMATION □□□□ □□□ □□□□□□.
(□□ □□ □□ □□□□□.)
PRODUCT_ID □□ □□ □□□□.
□□ □□□ □□□□ □□□□ □□□□□.
SQL > □□□ □□ upper_name_idx
ON product_information(UPPER(product_name));
PRODUCT_INFORMATION □□□□ □□ □□□□ □□□□.
□□ □□□ UPPER_NAME_IDX □□□□ □□□□□?
A. SELECT product_idFROM product_informationWHERE UPPER(product_name) IN ('LASERPRO', 'CABLE');
B. SELECT UPPER(product_name)FROM product_information;
C. SELECT UPPER(product_name)FROM product_informationWHERE product_id = 2254;
D. SELECT product_id, UPPER(product_name)FROM product_informationWHERE UPPER(product_name) = 'LASERPRO' □□ list_price > 1000;
Answer: A (LEAVE A REPLY)

NEW QUESTION: 97
□□ □□ □□□ □□□□ EXISTS □□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. □□ □□□ □ □□ □□ □□□□ □□ □□□ □□ □□ □□□ □□□□□.
B. □□ □□□□ □□□ □□ □□ □□ □□□ □□□□□ □□□ □□□□□ □ □□□□□.
C. □□ □□□□ □□□ □□ □□ □□□ □□ □□□ □□□□□ □□□ □□□□□ □ □□□□□.
D. □□ □□□ □□ □□□ □□ □□ □□□ □□□ □□ □□□ □□ □□□ □□ □□□□□.
Answer: A,C (LEAVE A REPLY)
□□
□□:
http://www.techonthenet.com/oracle/exists.php

NEW QUESTION: 98
□□ CREATE TABLE □□□ □□□□□.

```
CREATE TABLE order_item
(order_id NUMBER (3),
item-id NUMBER (2),
qty NUMBER (4),
CONSTRAINT ord_itm_id_pk
        PRIMARY KEY (order_id, item_id)
      USING INDEX
       (CREATE INDEX ord_itm_idx
        ON order_item (order_id, item_id)));
```

□□ SQL□□ □□ □□□□ □□ □□?

A. □□□□□ □□□□ ORD_ITM_IDX □□□□ □□□□□.

B. CREATE TABLE □□□□ USING INDEX □□ □□□□ □□ □□□ □□□ □□□□□.

C. □□□□□ □□□□ □ □□ □□□ ORD_ITM_IDX □ ORD_ITM_ID_PK□ □□□□□.

D. USING INDEX □□ □□ □□□□ □□□ □ □□ □□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 99

□□□□ □□ SALES □ PRODUCTS □□□□ □□□ □□□□□□. (2□□ □□□□□.)

SALES

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(3)
CUST_ID	NOT NULL	NUMBER(4)
TIME_ID		DATE
QTY_SOLD		NUMBER(10,2)

PRODUCTS

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(3)
PROD_NAME		VARCHAR2(30)
PROD_LIST_PRICE		NUMBER(8,2)

SALES □□□□□ PROD_ID□ PRODUCTS □□□□ PROD_ID□ □□□□ □□ □□□□. □ □□ ID □ □□ □□□ □□□□ □□□.

JOIN □□□□ □□□ □□ □□□ □□□□□.

```
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;
```

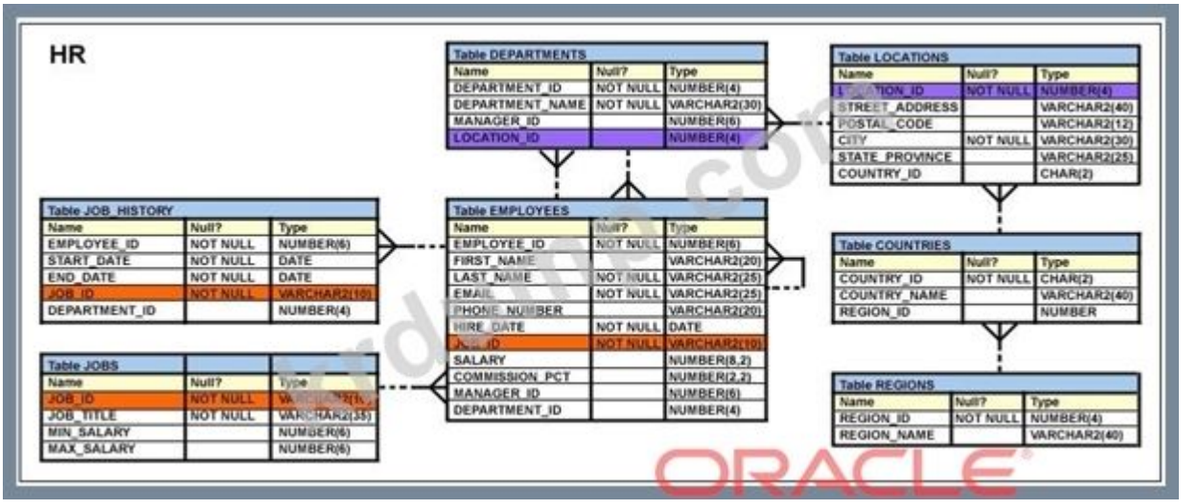
□□□ □□□ □□ □□ □□ □ □□ JOIN □□□ □□□ □ □□□□? (2□□ □□□□□.)

A. □□ □□ □□

B. □□

C. □□□ □□ □□
D. □□ □□ □□
Answer: (SHOW ANSWER)

NEW QUESTION: 100
Exhibit□ □□ EMPLOYEES □□□□ □□□ □□□□□□.
MANAGER_ID□ 100□ □□ □□□ □□ □□□□ □□□□□ □□□.
□ □□ □□ □□□ □□□□. □ □□ □□□ □□□□ LAST_NAME□ □□ □ □□ □□□ □□□ LAST_NAME□ □□□□.



□□ SQL □□ □□□□□□□□?
A. m.last_name "□□□", e.last_name "□□" □□
FROM □□ m JOIN □□ e
ON m.employee_id = e.manager_id
WHERE m.manager_id=100;
B. SELECT m.last_name "□□□", e.last_name "□□"
FROM □□ m JOIN □□ e
ON e.employee_id = m.manager_id
WHERE m.manager_id=100;
C. SELECT m.last_name "□□□", e.last_name "□□"
FROM □□ m JOIN □□ e
WHERE m.employee_id = e.manager_id AND e.managerjd=100;
D. m.last_name "□□□", e.last_name "□□" □□
FROM □□ m JOIN □□ e
ON m.employee_id = e.manager_id
WHERE e.managerjd=100;
Answer: D (LEAVE A REPLY)

NEW QUESTION: 101
□□ □□□ □□ □□□□ □□ □□?
A. □□ □□ □□□(+)□ □□□□ □□ □□ □□□□ □ □□ □□□□□.

B. IN □□□□ □□ □□□ □□□□ □□□□ □□□ □ □□□□.
C. □□ □□□ □□□□ □□□ □□ □□ □□□□ □□□□ □□ □□□ □□□ □ □□□□.
D. □□ □□ □□□(+)□ □□ □□□□ □□ □□□ □□□ □□ □□□ □ □□□□.
E. □□ □□□ □□ □□□ □□ □□ □□ □□□□ □ □□□□□.
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 102
SELECT □□ WHERE □ HAVING □□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. HAVING □□ □□ □□□□ □□ □□□ □□ □□□ □ □□□□.
B. HAVING □□□ □□□□ □□ □□ □ □□ □□□ SELECT □□□ □□□□□ □□□.
C. WHERE □ HAVING □□ □□□□ □□ □□ □□□□ □□□□ □□□ □□□ □□□ □ □□□□.
D. WHERE □□ □□□□ □□ □□□□ □□□ □□ □□□ □ □□□□.
E. WHERE □□ □□□□ □□ □□□□ □□ □ □□□ □ □□□□.
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 103
□□□□ □□ ORDER_ITEMS □□□□ □□□ □□□□□□.

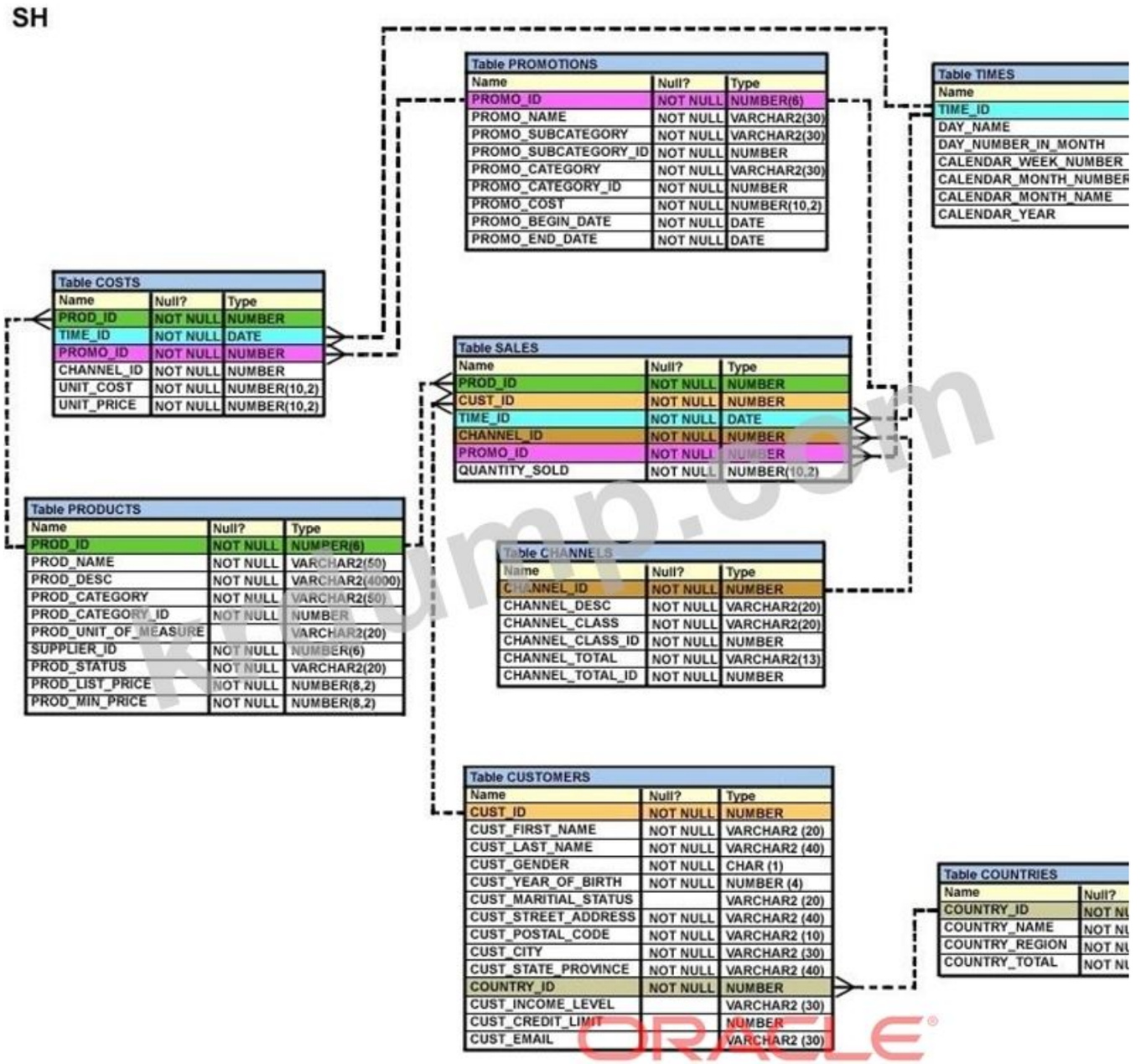
ORDER_ITEMS

ORDER_ID	LINE_ITEM_ID	PRODUCT_ID	UNIT_PRICE	QUANTITY
2355	4	2322	19	188
2355	5	2323	17	190
2355	9	2359	226.6	204
2355	1	2289	46	200
2356	5	2308	58	47
2356	6	2311	95	51
2356	1	2264	199.1	38
2356	2	2274	148.5	34
2356	3	2293	98	40
2356	4	2299	72	44
2357	2	2245	462	26
2357	3	2252	788.7	26
2357	4	2257	371.8	29
2357	5	2262	95	29

ORDER_ITEMS □□□□ □□ □□ □ □□ □□□ □□ □□ □□□ ORDER_ID□ □□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?
□□ ID □□
A. order_items□□

WHERE(□□*□□) = (SELECT MAX(□□*□□)
□□_□□□□)
GROUP BY order_id;
□□ ID □□
B. □□_□□□□
WHERE(□□*□□) = (SELECT MAX(□□*□□)
□□_□□□□
GROUP BY order_id)
C. □□ □□□□
WHERE(□□*□□) = MAX(□□*□□)
GROUP BY order_id;
□□ ID □□
D. order_items□□
GROUP BY order_id
HAVING SUM(□□*□□) = (SELECT MAX(SUM(□□*□□)) FROM order_items GROUP BY order_id);
□□ ID □□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 104
□□□□ □□ SALES, CUSTOMERS, PRODUCTS □ TIMEStables□ □□□ □□□□□□.



PROD_ID □□ PRODUCTStable□ □□□□ SALEStable□ □□ □□□□.

□□□□□ CUST_ID □ TIME_ID □□ □□ CUSTOMERS □ TIMEStable□ □□□□ SALEStable□ □ □ □□□□.

□□ CREATE TABLE □□□ □□□□□.

CREATE TABLE new_sales(prod_id, cust_id, order_date DEFAULT SYSDATE)

□□

SELECT prod_id, cust_id, time_id

□□□□;

□ □□□ □□ □□□□ □□ □□?

A. NEW_SALEStable□ □□□□ □□□ □□ □□□ □□ NOTNULL □□ □□□ □ □□□□ □□□□ □.

B. □ □□□ DEFAULT □□ □□□ □ □□ □□□ NEW_SALEStable□ □□□□ □□□□.

C. NEW_SALEStable□ □□□□ □□□ □□ □□□ □□ FOREIGNKEY □□ □□□ □ □□□□ □□ □□□.

D. CREATETABLE □□□ □ □□□ SELECT □□ □□□□ □□ □□□ NEW_SALEStable□ □□□ □ □□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 105

□□□ □□□ □□ □ □□ □□ □ □□ □□? (3□□ □□□□□.)

A. BLOBdata □□ □□ □□ □□ □□□ □□ □□□□ □□□□ □ □□□□□.

B. □□□□ □□□ LONG □□ □□□ □ □□□□.

C. VARCHAR2 □□□ □□ □□□ □□ □□□ □ □□ □□ □□ □□□ 1□□.

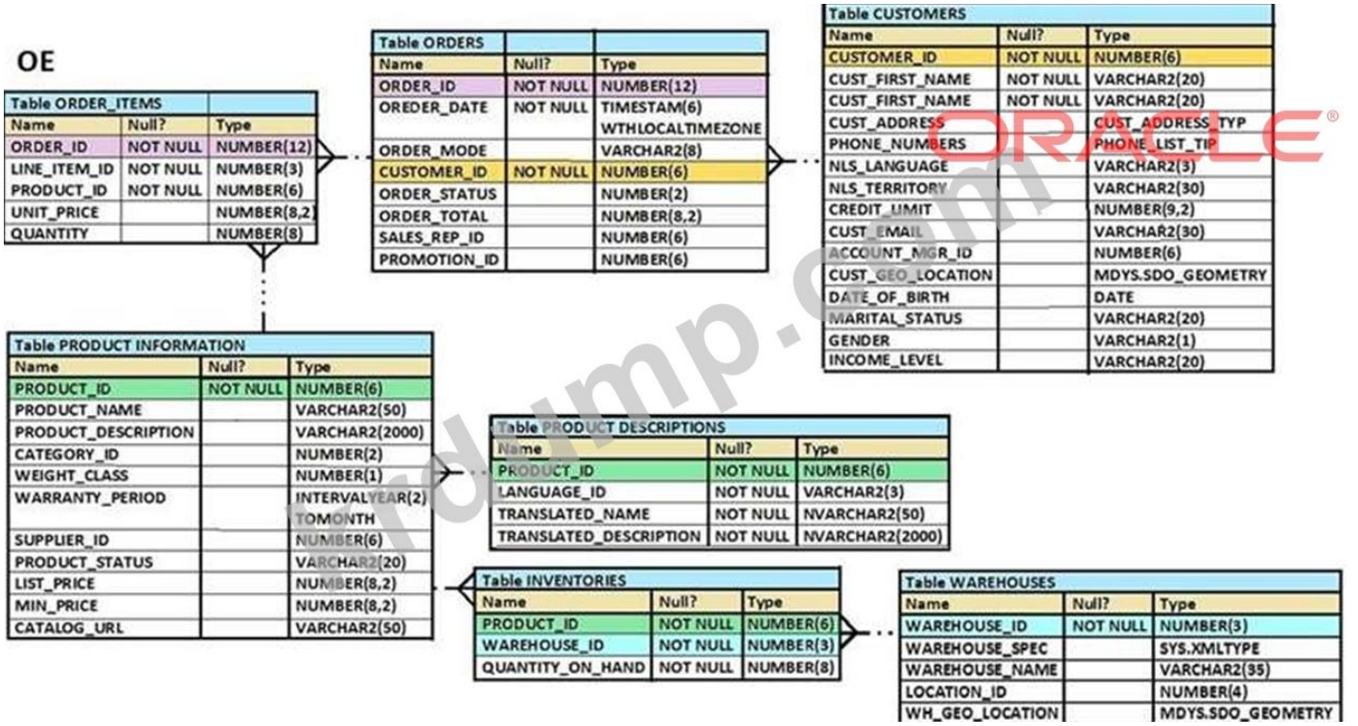
D. CHARdata □□ □□ □□ □□□ □□ □ □□□□ □□□□ □□□□□.

E. TIMESTAMP □□□ □□ □□ □□ □□□ □□ □□ □□ □□ □□□□□.

Answer: B,C,D (LEAVE A REPLY)

NEW QUESTION: 106

□□□□ □□ ORDER_ITEMS □ ORDERS□□□□ □□□ □□□□□□.



□ □□□ 50,000□□ □ ORDER_ID, □□_ID □ □ □□(UNIT_PRICE □□□ QUANTITY)□ □□□□ □ □□□□ □□□□□.

□□ SQL □□ □□□□□□.

SELECTprder_id, product_id, unit_price*quantity "□ □□"

□□_□□□□

WHERE □□*□□ > 50000

NATURAL JOIN □□

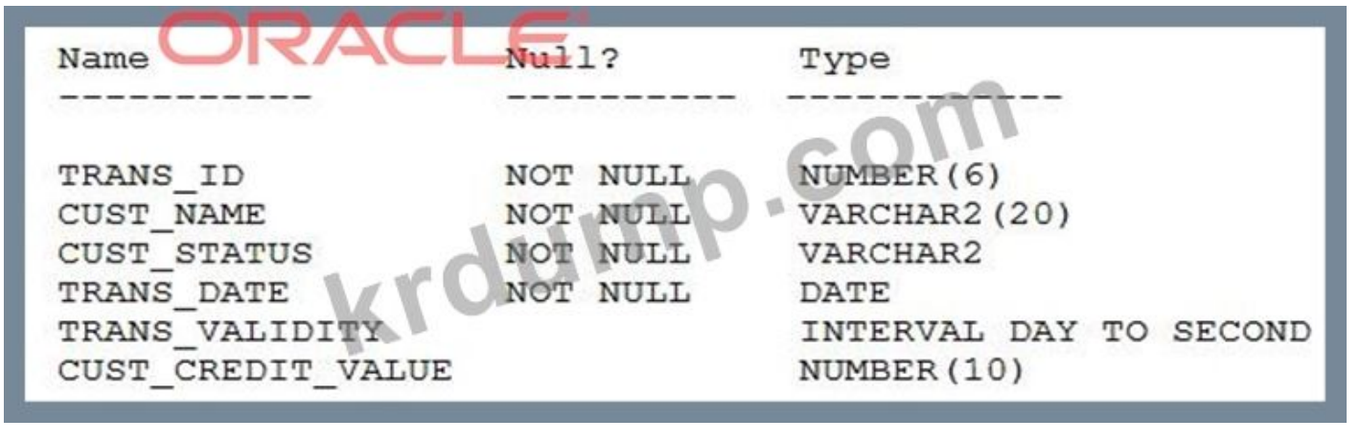
□□□ □□ □□□□ □□ □□?

A. NATURAL JOIN □□ ON □□□□ □□ □□□ □□□□ □□□□ □□□□.
B. NATURAL JOIN □□ USING □□□□ □□ □□□ □□□□ □□□□ □□□□.
C. □□□□ □□□□ □□□ □□□ □□□□□.
D. WHEREclause□ NATURAL JOINclause □□ □□ □□□ □□ □□□□ □□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 107
□□□□□ □ □□□ □□ □□□□ □□ □□?
A. □□ □ □□ □□□ □ □□□□.
B. □□ □□□□ □□ □□□ □□□□ □□ □□□ □ □□□□.
C. ALTER TABLE □□□ □□□□ □□ □□ □□□ □□□ □ □□□□.
D. □□ □□□□□ □□□□ □□ □□□□□ □□□□□ □□□.
E. □□□□ □□ □□□□ □□ □□□ □ □□□□.
F. □ □□□ □□□□□ □□□□□.
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 108
□□□ NLS_DATE_FORMAT□ DD Mon YYYY□ □□□□□.
2019□ 1□ 1□ □□ □□□□ □ □□□ □□□□□?
A. SELECT TO_DATE('2019-01-01') FROM DUAL;
B. □□□□ □□ '2019-01-01' □□;
C. □□□□ '2019-01-01' □□;
D. SELECT TO_CHAR('2019-01-01') FROM DUAL;
E. SELECT TO_DATE('2019-01-01', 'YYYY-MM-DD') FROM DUAL;
Answer: B,E (LEAVE A REPLY)
□□ □□□□□ '2015-01-10' □□ □□ d
SELECT to_date('01-01-2007' , 'DD-MM-YYYY') Tarih FROM □□;
□□: https://livesql.oracle.com/apex/livesql/file/content_BFKOA5UQHDV1MV7HRUDDUEPM4.html
https://www.baskent.edu.tr/~tkaracay/etudio/ders/dbase/sql/pdfSQL/DateConversions.pdf

NEW QUESTION: 109
TRANSACTIONStable□ □□ □□□ □□□ □□□□□.



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)

□□ □□□ □□□□ □□□□ □□□ □□ □ □□ □□□□ □□ □□ □□? (2□□ □□□□□.)
A. CUST_STATUS □□ □□ VARCHAR2□□ 4,000□□□ □□□□ □□□ □ □□□□.
B. TRANS_VALIDITY □□ □□□□ □, □, □, □ □□□ □□ □□□ □□□ □ □□□□.
C. TRANS_DATE □□ dd-mon-yyyy □□□□□ □□□ □□□ □ □□□□.
D. CUST_CREDIT_VALUE □□ □□ □ □□ □□□ □□□ □ □□□□.
Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 110
□□ □□□ □□□□□.
TRUNCATE TABLE □□;
□□ □□ □□□□□?
A. □□□□ □□□ □□ □□□□ □□□□□.
B. □□□ □□ □□□□ □□□ □□ □□□□□.
C. ROLLBACK □□ □□□□ □□□ □□□□ □□□ □ □□□□.
D. Flashback TABLE □□ □□□□ □□□ □□□□ □□□ □ □□□□.
E. □□□□ □□□ □□□□ □□□□□.
F. □□□□ □□□ □□□ □□ □□□ □□□□□.
Answer: E,F (LEAVE A REPLY)

NEW QUESTION: 111
Oracle Database□ □□ □□ □□□□ □□ □□ □□? (3□□ □□□□□.)
A. □□ □□□□ □□□□ □□ □□□□ □□□□ □□ □□□ □ □□□□.
B. □□ □□□ WHEREclause□ □□□ □□ □□□ □ SELECT □□ WHERE □□ □□□ □ □□□ □.
C. □□□ □□□□□ □□□□.
D. □□ □□□ □□□□ □□□ □□□□ □□□□ □□□ □ □□□□.
E. □□□ □□ □□□ □□□□.
F. □□ □□□□ □□□□ □□□ □□ □□ □□□ □□ □□□□ □□□□□.
Answer: A,C,F (LEAVE A REPLY)

NEW QUESTION: 112

Oracle SQL, DELETE □ TRUNCATE □□ □□ □□□□ □□ □□?

A. TRUNCATE □□ □□□□ DELETE □□ □□□□ □□□□□ □□ □□□□□ □□□ □ □□□□.

B. DELTE □ TRUNCATE □□ □□□ □□□□ □□□□ □□□□ □□□ □ □□□□.

C. TRUNCATE □□ □□□□ DELETE □□ □□□□ □□□□ □□□ □□ □□□ □□□□ □□□ □ □□□□.

D. DELETE □ TRUNCATE □□ □□□□ □□□□ □□ □□□□ □□□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 113

□□ □□ □□□ □□□□□.

SELECT ch.channel_type, t.month, co.country_code, SUM(s.amount_sold) SALES

FROM □□ s, □□ t, □□ ch, □□ □□

WHERE s.time_ id = t.time id

AND s.country_ id = co. □□ ID

□□□ s. □□ ID = ch.□□ ID

AND ch.channel type IN('□□ □□', '□□□')

AND t.month IN('2000-09', '2000-10')

AND □□ □□ □□ IN('GB', 'US')

□□ □□□ □□□□□□.

CHANNEL TYPE	MONTH	co	SALES
internet	2000-09	GB	16569
internet	2000-09	US	124224
internet	2000-09		140793
internet	2000-10	GB	14539
internet	2000-10	US	137054
internet			292387
Direct Sales	2000-09	GB	85223
Direct Sales	2000-09	US	638201
Direct Sales	2000-09		723424
Direct Sales	2000-10	GB	91925
Direct Sales	2000-10	US	638201
Direct Sales	2000-09		774222
Direct Sales			1497646

□□□ □□□ □□□ □□□□□ □□ GROUP BY □□ □□□□ □□□?

A. GROUP BY ch.channel_type,ROLLUP(t □, co.country_code) ;

B. □□ BYch. channel_ type, t.month,ROLIUP (co. country_ code) ;

C. GROUP BY ch.channel_type, t.month, co.country □□;

D. GROUP BY CUBE(ch. channel_ □□, t .month, co. □□ □□);

Answer: (SHOW ANSWER)

NEW QUESTION: 114

□□□□□ □□□□ □□ □□□□ □□□□□□.

□□ □□ □□ NLS_DATE_FORMAT = 'DD-MON-YYY HH24:MT:SS;'

□□ □□ □□ □□□ = '-5:00;:

□□□ □□□□□.

```
DBTIMEZONE   SYSDATE
----------   --------------------
+00.00       11-JUL-2019 11:00:00
```

LOCALTIMESTAMP□ □□□ □□□□ □□□ □□□□□?

A. 2019□ 7□ 11□ □□ 6.00.00.00000000

B. 2019□ 7□ 11□ 11.00.00.00000000 AM -05:00

C. 2019□ 7□ 11□ 11.00.00.00000000 AM

D. 2019□ 7□ 11□ 6.00.00.00000000 AM -05:00

Answer: A (LEAVE A REPLY)

NEW QUESTION: 115

□□ □□□ □□□□□.

SQL > DEFINE □□ □□ = '01-APR-2011'

SQL > SELECT □□ ID, □□, □□

□□□□□□

WHERE hire_date > '□□(&H)'

AND manager_id > &mgr_id;

□□□ □□ □□ □□ □□□ □□ □□□□□ □□□□□?

A. □□ □□ "hiredate' □ 'mgr_id' □ □.

B. □□□□'

C. 'mgr_id'□

D. □□□ □□□□ □□ □□□ □□

Answer: C (LEAVE A REPLY)

NEW QUESTION: 116

MEMBERS □□□□ □□□ □□□□□□.

```
Name                 Null?      Type
---------------      ----------  ----------------------
MEMBER_ID            NOT NULL    VARCHAR2 (6)
FIRST_NAME                       VARCHAR2 (50)
LAST_NAME            NOT NULL    VARCHAR2 (50)
ADDRESS                          VARCHAR2 (50)
CITY                             VARCHAR2 (25)
STATE                            NOT NULL VARCHAR2 (3)
```


□□ □ MO □ MI □□ □□□□ □□□□ □□ □□ □□□ □□□□ □ □□□ □ □□ □□□ □□□ □□?

A. □ □□, □□ FROM □□ WHERE state ='MO' AND state ='MI';

B. SELECT □, □□ FROM □□ WHERE state IN ('MO', 'MI');

C. SELECT DISTINCT □, □□ FROM □□ WHERE state ='MO' OR state ='MI';

D. SELECT □, □□ FROM □□ WHERE □□ LIKE 'M%';

Answer: B (LEAVE A REPLY)

NEW QUESTION: 117
SQL □□ GROUP BY □□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. GROUP BY □□□ □ □□□ □□□ □ □□□□.
B. SELECT □□□ □□ □□□ □□□□ □□ GROUP BY □□ □□□□□.
C. GROUP BY □ □□ WHERE □□ □□□□ □□ □□ □ □□ □□□□□.
D. SELECT □□ □□ □□□ □□ □□ SELECT □□ □□ □□□ □□ □□ □□ GROUP BY □□ □ □□□□ □□□.
E. GROUP BY □ □□ WHERE □□ □□□□ □□□ □□□□ □□ □□ □□□□□.
Answer: D,E (LEAVE A REPLY)

NEW QUESTION: 118
□□□□ □□ PRODUCT_INFORMATION □□□□ □□□□□ □□□□□□.

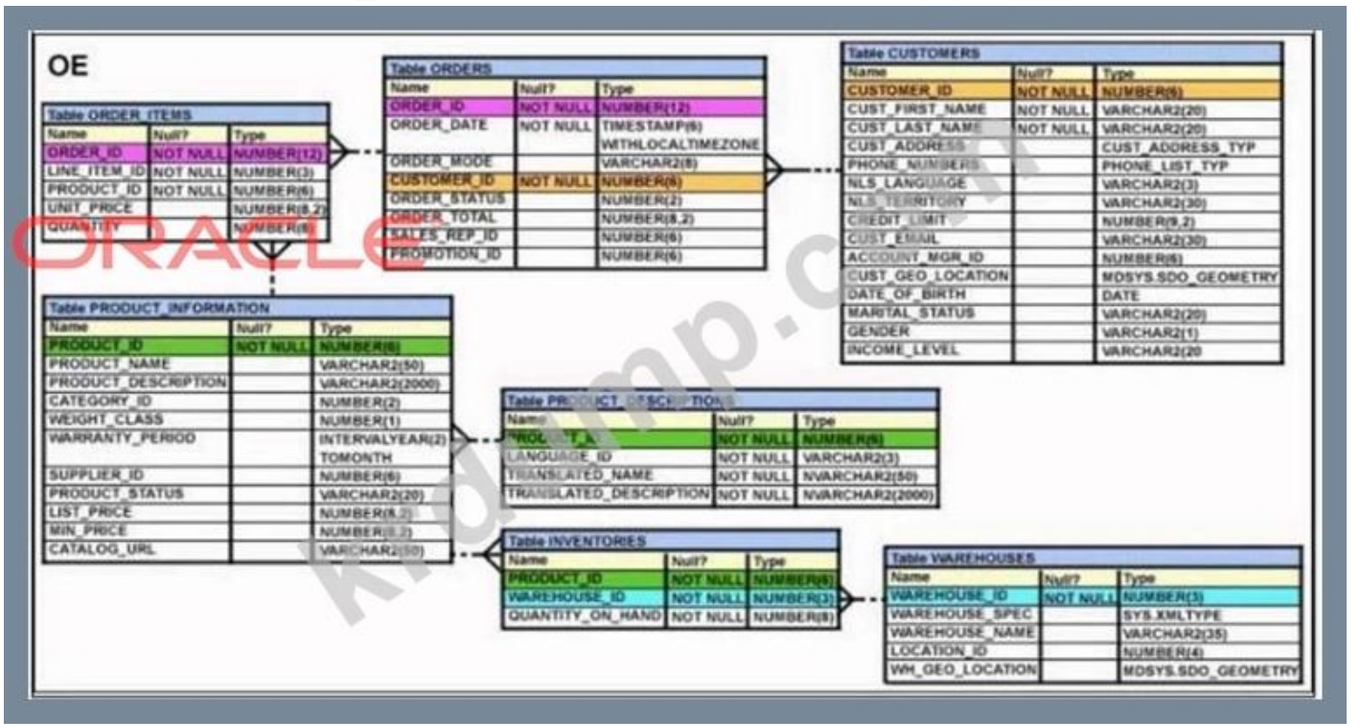
PRODUCT_NAME	CATEGORY_ID	SUPPLIER_ID
nkjet C/8/HQ	12	102094
nkjet C/4	12	102090
.aserPro 600/6/BW	12	102087
.aserPro 1200/8/BW	12	102099
nkjet B/6	12	102096
ndustrial 700/ID	12	102086
ndustrial 600/DQ	12	102088
Compact 400/LQ	12	102087
Compact 400/DQ	12	102088
HD 12GB /R	13	102090
HD 10GB /I	13	102071
HD 12GB @7200 /SE	13	102057
HD 18.2GB @10000 /E	13	102078
HD 18.2GB @10000 /I	13	102050
HD 18GB /SE	13	102083
HD 6GB /I	13	102072
HD 8.2GB@5400	13	102093

CATEGORY_ID □□ □□ 12 □□ 13□□ SUPPLIER_ID □□ □□ 102088□ □□□□□
PRODUCT_NAME□ □□□□ □□ □□ □□□ □□□□. □□ SQL □□ □□□□□□.
□□ □□ □□
FROM □□ □□
WHERE (category_id = 12 AND category_id = 13) AND Supplier_id = 102088; □□ □□□ □□ □□□ □ □□ □□?

A. □□□□ □□□ □□□ □□□ □□□□□.
B. □□□□□ □□□ □□ □□□□ □□□□.
C. □□ WHEREclause □□□ □□ □□ □□□□ □□ □□ □□□ □□□□ □□□□.
D. □□□ □□□□ □□ AND □□ □□□□ □□□ □□□ □□ □□□□□ □□□ □□□□ □□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 119
□□□□ □□ PRODUCT_INFORMATION □□□□ □□□□□ □□□□□□.



□ SQL □□ □□□□□□.
SELECT TO_CHAR(□□ □□, '$9,999')
□□ □□□□;
□□□ □□ □ □□ □□ □ □□ □□? (2□□ □□□□□.)

A. LIST_PRICE □□ □ 1123.90□ □□□ □□ $1,123□ □□□□□.
B. LIST_PRICE □□ □ 11235.90□ □□□ □□ $1,123□ □□□□□.
C. LIST_PRICE □□ □ 11235.90□ □□□ □□ #######□□ □□□□□.
D. LIST_PRICE □□ □ 1123.90□ □□□ □□ $1,124□ □□□□□.

Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 120

SALES1 □□□□ □□ □□□ □□□□□.

```
Name                 Null            Type
-----------------    -------------   ------
SALES_ID             NOT NULL        NUMBER
STORE_ID             NOT NULL        NUMBER
ITEMS_ID                             NUMBER
QUANTITY                             NUMBER
SALES_DATE                           DATE
```

SALES2□ SALES1□ □□□ □□□ □□ □□□□□□.

□□ □□ □□□□ □ □□□ □□□ □□□□□.

SALES2 □□□□ □□ SALES1 □□□□ □□ □□□□□ □□□.

□□ □□ □□□□ □□□ □□□ □□□□□?

A. □□

B. □□

C. □□□□

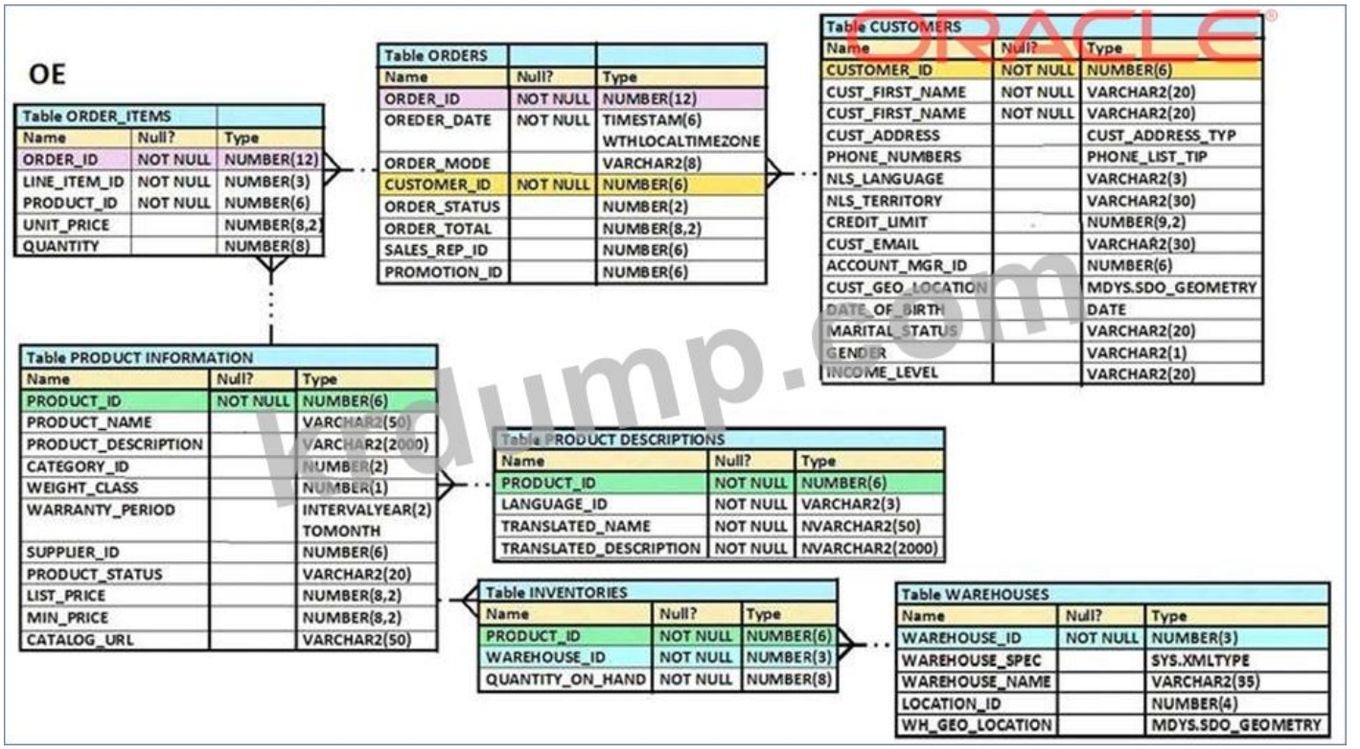
D. □□□ □

E. □□□

Answer: C (LEAVE A REPLY)

NEW QUESTION: 121

□□□□ □□ ORDER_ITEMS □□□□ □□□ □□□□□□.



□□ SQL □□ □□□□□.

SELECT order_id, product_id, unit_price

□□_□□□□

WHERE □□

(SELECT MAX(□□)

□□_□□□□

□□ BY order_id);

ORDER_ID□ UNIT_PRICE□ □□ □□ □□□ PRODUCT_ID□ □□□□□ □□□.
□□ □□□□ □□ □□ SQL □□□ □□ □□□ □□□□□ □□□?
A. □□ □□□□ GROUP BY □□ □□□□ □ □□□ □□
B. =□ >ANY □□□□ □□□□.
C. =□ >ALL □□□□ □□□□.
D. =□ IN □□□□ □□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 122
□□ □□□ □□□□□.
```
TRUNCATE TABLE depts;
```
□□ □□ □□□□□? (2□□ □□□□□.)
A. □□□□ □□□ □□ □□□□ □□□□□.
B. □□□ □□ □□□□ □□□ □□ □□□□□.
C. ROLLBACK □□ □□□□ □□□ □□□□ □□□ □ □□□□.
D. □□□□ □□□ □□□ □□ □□□ □□□□□.
E. □□□□ □□□ □□□□ □□□□□.
F. FLASHBACK TABLE □□ □□□□ □□□ □□□□ □□□ □ □□□□.
Answer: D,E (LEAVE A REPLY)
□□/□□: https://docs.oracle.com/html/E25494_01/general003.htm

NEW QUESTION: 123
□□ SQL □□□ □□□□□.

```
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));
```

□□□ □□ □□□ □□□□□. SQL □ □□□ □□□ □□□□□.

A. ORD_NO□ FOREIGN KEY□□ □□□ ORD_NO □ ITEM_NO□□ □□ □□ □□ □□□ □ □□□ □.

B. CHECK □□ □□□□□ BETWEEN□□ □□□ □ □□□□.

C. NEXTVALsequence □□ □□ DEFAULT □□□ □□□ □ □□□□.

D. CHECK □□ □□□□□ SYSDATE□ □□□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 124

□□ Oracle □□□□ □□□□ □□ □□□□ □□□□□□.

```
CREATE TABLE product (pcode NUMBER(2), pname VARCHAR2(20));

INSERT INTO product VALUES (1, 'pen');

INSERT INTO product VALUES (2, 'pencil');

INSERT INTO product VALUES (3, 'fountain pen');

SAVEPOINT a;

UPDATE product SET pcode = 10 WHERE pcode = 1;

COMMIT;

DELETE FROM product WHERE pcode = 2;

SAVEPOINT b;

UPDATE product SET pcode = 30 WHERE pcode = 3;

SAVEPOINT c;

DELETE FROM product WHERE pcode = 10;

ROLLBACK TO SAVEPOINT b;

COMMIT;
```


□ □□ □□ □ □□ □□? (3□□ □□□□□.)

A. □□□□ □□□ □□ □□□□.

B. □□ □□ 3□ □□.

C. □□□ □□□ □□ □□□□.

D. □□ 1□ □□ □□.

E. □□ □□□ □□ □□□□.

F. □□□ 10□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 125

□□ □ □□ □□□ □□ □□□□ □□ □□? (3□□ □□□□□.)

A. □□ □□ □□□ □□□□ □□ □□□□ □□ □□ □□□□□.

B. □□ □□□ □□□□ □□ □□□□□.

C. □ □□□□ □□ □□ □□□ □□ □□ □□ □□□ □□□ □ □□□□.

D. □□ □□□ □□□ □ □ □□□□□ □□□ □ □□□□.
E. □□ □□ □□□ Oracle □□□ □□□□ □□□.
F. □□ □□ □□□ □□ □□□ □□□□ □□ □□ □□□□□.
Answer: A,B,F (LEAVE A REPLY)

NEW QUESTION: 126
ALTER TABLE....DROP COLUMN.... □□□ □□ □□□□ □□ □□? (3□□ □□□□□.)
A. CASCADE □□□ □□ □□ PRIMARY KEY□ □□ □□□ □ □□□□.
B. □□□□ □□□□ □□ □□ □□□□ □□ □□□ □ □□□□.
C. □□□□ □□ □□ □□□□ □□□□ □□ □□□ □ □□□□.
D. □□□ □□ □□□ □ □□□□.
E. □□□□ □□ □ □□ □□□ □ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 127
Exhibit□ □□□□ CUSTOMERS □□□□ □□□ □□□□□□.

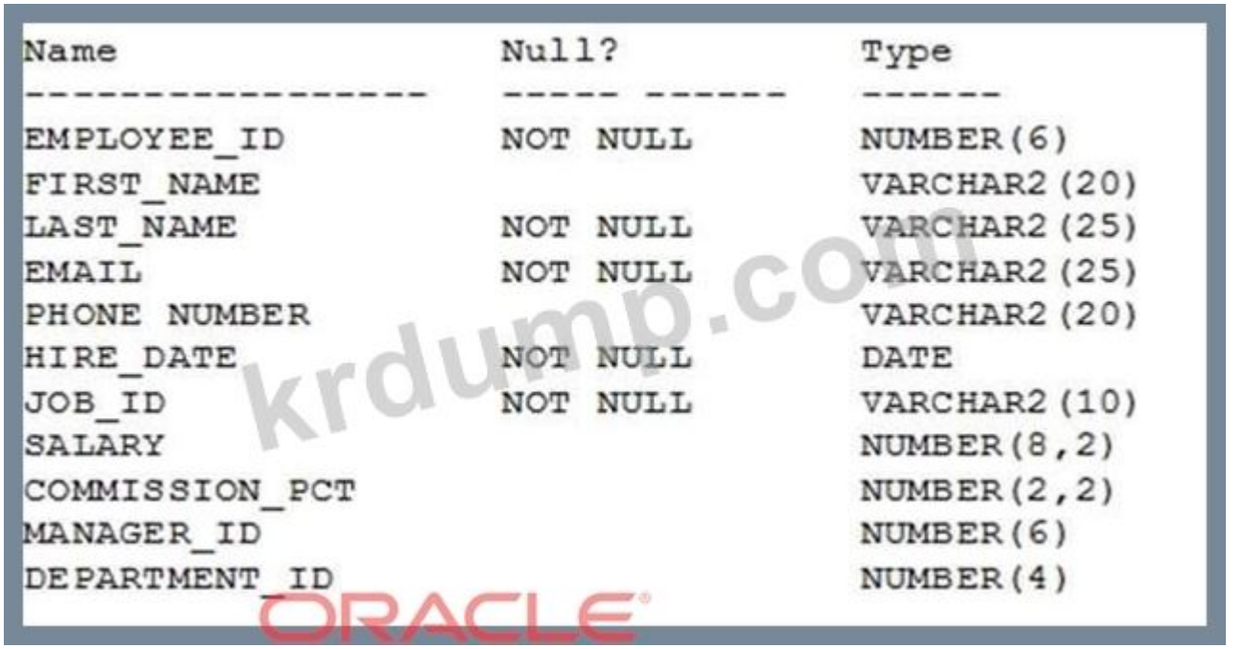
Table CUSTOMERS

Name	Null?	Type
CUST_ID	NOT NULL	NUMBER
CUST_FIRST_NAME	NOT NULL	VARCHAR2 (20)
CUST_LAST_NAME	NOT NULL	VARCHAR2 (40)
CUST_GENDER	NOT NULL	CHAR (1)
CUST_YEAR_OF_BIRTH	NOT NULL	NUMBER (4)
CUST_MARITIAL_STATUS		VARCHAR2 (20)
CUST_STREET_ADDRESS	NOT NULL	VARCHAR2 (40)
CUST_POSTAL_CODE	NOT NULL	VARCHAR2 (10)
CUST_CITY	NOT NULL	VARCHAR2 (30)
CUST_STATE_PROVINCE	NOT NULL	VARCHAR2 (40)
COUNTRY_ID	NOT NULL	NUMBER
CUST_INCOME_LEVEL		VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2 (30)

CUSTOMERS □□□□ □□□□ □□ □□□ □□ □□□ 15% □□□ □□ □□□□ □□□□ □□□ □ □□□. □□ □□□ □□□□ □□ □□□ "□□□ □ □□" □□□□ □□□□□ □□□.
□□ SQL □□ □□□ □□□ □□□□□?
A. SELECT NVL(cust_credit_limit,'□□□ □ □□')*.15 "NEW CREDIT" FROM □□;
B. SELECT NVL(cust_credit_limit*.15,'□□□ □ □□') "NEW CREDIT" FROM □□;
C. SELECT TO_CHAR(NVL(cust_credit_limit*.15,'□□□ □ □□')) "NEW CREDIT" FROM □□;
D. SELECT NVL(TO_CHAR(cust_credit_limit*.15),'□□□ □ □□') "NEW CREDIT" FROM □□;
Answer: D (LEAVE A REPLY)
NVL □□
null □□ □□ □□□ □□□□□.
□□□ □ □□ □□□ □□□ □□, □□ □ □□□□□.
□□□ □□□ □□□ □□□□ □□□.
-
NVL(□□□_pct,0)
-

NVL(□□□□, ' 01-1□-97')
- NVL(job_id, '□□ □□ □□')

NEW QUESTION: 128
EMPLOYEEStable□ □□□ □□□□□□.



Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

EMPLOYEE_ID□ MANAGER_ID □□□□ □□/□□ □□□ □□□□.
EMPLOYEE_ID□ 123□ □□□ □□□ □□□□□ □□□ □□□ □ □ □□□ ID□ □□□□□ □□ □.
□□ □□□ □□□ □□□ □□□□□?
SELECT e.last_name, m.manager_id
A. FROM □□ e RIGHT OUTER JOIN □□ m
on (e.employee_id = m.employee_id)
□□ e.employee_id = 123;
□□ m.last_name, e.manager_id
B. FROM □□ e LEFT OUTER JOIN □□ m
on (e.manager_id = m.manager_id)
□□ e.employee_id = 123;
C. FROM □□ e RIGHT OUTER JOIN □□ m
on (e.manager_id = m.employee_id)
□□□ e.employee_id = 123;
SELECT e.last_name, m.manager_id
D. FROM □□ e LEFT OUTER JOIN □□ m
on (e.employee_id = m.manager_id)
□□ e.employee_id = 123;
SELECT e.last_name, e.manager_id
Answer: B (LEAVE A REPLY)

NEW QUESTION: 129
CURRENT_TIMEITAMP□ □□ □□□□ □□ □□?
A. □□□ DBTIMEZONE□ □□□□□□.
B. SESSIONTIMEZONE □□□ □□ □□ □□□□□.
C. □□□ DBTIMEZONE□ □□□□□□.
D. □□ SYSTIMESTAMP□ □□□ □□ □□□□□.
E. CURRENT_DATE□ □□□ □□□ □□□□□.
F. TIMESTAMP □□□ □□□ □□ □□□□□.
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 130
MEMBERS □□□□ □□□ □□□□□.

NAME	NULL?	TYPE
MEMBER_ID	NOT NULL	VARCHAR2(6)
FIRST_NAME		VARCHAR2(50)
LAST_NAME	NOT NULL	VARCHAR2(50)
ADDRESS		VARCHAR2(50)
CITY		VARCHAR2(25)
STATE		VARCHAR2(3)

SQL □□ □□□□□.
SQL > SELECT □□, last_name LNAME FROM MEMBERS ORDER BY 1, LNAME DESC;
□□ □□□ □□□ □□□? (□□ □□ □□ □□□□□.)
A. □□ □□□ □□□□□□ □□□□, □□ □□□□□□ □ □□□□□.
B. ORDER BY □□□ □□ □□□ □□ □□□ □□ □□□ □ □□ □□□ □□□□.
C. ORDER BY □□□ □□ □□□ □□□ □ □□ □□□ □□□□□.
D. □□ □□□ □□□□□□ □□□□, □□ □□□□□□ □ □□□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 131
□□ □□ □□□ □□□□□.

```
CREATE TABLE cust (
  cust_id NUMBER(2),
  credit_limit NUMBER(10)
)
ORGANIZATION EXTERNAL
```

□ □□□ □□□□□ □□□□□ □□ □ □□ □□□□□?
A. DEFAULT DIRECTORY □
B. REJECT LIMIT □

C. ACCESS PARAMETERS □
D. □□□ □□□□ TYPE □
E. LOCATION □
Answer: (SHOW ANSWER)

NEW QUESTION: 132
BOOKS_TRANSACTIONS □□□□ □□ □□□ □□□□□.

Name	Null?	Type
TRANSACTION_ID	NOT NULL	VARCHAR2(6)
TRANSACTION_TYPE		VARCHAR2(3)
BORROWED_DATE		DATE
BOOK_ID		VARCHAR2(6)
MEMBER_ID		VARCHAR2(6)

□□ □□ SQL □□ □□□□□.
SELECT * FROM books_transactions
□□□ □□□ □□□□ □ □□ WHERE □□□ □□□□□?
A. WHERE □□ □□ = SYSDATE AND (transaction_type = 'RM' AND member_id = 'A101' OR member_id = 'A102');
B. WHERE broken_date = SYSDATE AND (transaction_type = 'RM' OR member_id IN ('A101', ('A102'));
C. WHERE rider_date = SYSDATE AND (transaction_type = 'RM' AND (member_id = A101' OR member_id = 'A102'));
D. WHERE broken_date = SYSDATE AND transaction_type = 'RM' OR member_id IN('A101', 'A102');
E. WHERE(borrowed_date = SYSDATE AND transaction_type = 'RM') □□ member_id IN('A101', 'A102');
Answer: C,E (LEAVE A REPLY)

NEW QUESTION: 133
ORDER BY □□ □□ □□□□ □□ □□?
A. ORDER BY □□□□ □ □□□ □□□ □ □□□□.
B. □□ □□□□ □□ □□□□□ □□□□□.
C. NULLS□ □□ □□□ □□□□ □□□□.
D. □□ □□ □□□□ □□ □□ □□□□□□ □□□□□.
E. SELECT □□□ □□□ □□ ORDER BY □□□ □□□ □ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 134
□□□□ □□ ORDERS_MASTER □ MONTHLY_ORDERS □□□□ □□□□ □□□□□.

RDERS_MASTER

ORDER_ID	ORDER_TOTAL
1	1000
2	2000
3	3000
4	

ONTHLY_ORDERS

ORDER_ID	ORDER_TOTAL
2	2500
3	

□□ MERGE □□ □□□□□.
MERGE_INTO □□_□□□ o
USINGmonths_orders m
ON (o.order_id = m.order_id)
□□□ □
□□□□ SET o.order_total = m.order_total
DELETE WHERE(m.order_total IS NULL)
□□□□ □□ □
INSERT VALUES (m.order_id, m.order_total);
□ □□□ □□□ □□□ □□□?
A. ORDERS_MASTER □□□□□ ORDER_ID 1, 2, 3 □ 4□ □□□□□.
B. ORDERS_MASTER □□□□□ ORDER_ID 1, 2 □ 4□ □□□□□.
C. ORDERS_MASTER □□□□□ ORDER_ID 1, 2 □ 3□ □□□□□.
D. ORDERS_MASTER □□□□□ ORDER_ID 1□ 2□ □□□□□.
Answer: (SHOW ANSWER)
□□
□□:
https://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9016.htm

NEW QUESTION: 135
□□ □□ □□□□□?
A. CONCAT□ □ □ □□□ □□□□ □□□□□.
B. FLOOR□ □□□ □□□□ □□□ □□ □□ □ □□□ □□□□□.
C. FLOOR□ □□□ □□□□ □□□ □□ □□ □ □□ □□□ □□□□□.
D. CONCAT□ □ □□ □□□□ □□□□□.
E. INSTR□ □□ □□□ □□□ □□□□ □□□□ □□□□.
F. INSTR□ □□ 0□□ □□□□ □□□ □□□ □□□□ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 136

□□ □□□ □□□ □ □□ □□□ □□□□□?
A. □□ □□□ □□□ □□□(102)□ □□ □□□ □□□ □ □□ □□□□□.
B. PROD_LIST_PRICE□ 1000 □□□ □□□□ □□□□□.
C. PROD_LIST_PRICE□ □□ PROD_LIST_PRICE□□ □ □□□ □□ □□□□□.
D. □ □□ □□□ □□ □□ PROD_LIST_PRICE□ □□□□□.
E. PROD_MIN_PRICE□ □□ □□□ □□ PROD_LIST_PRICE□□ □□ □□□ □□ □□□ □□□ □ □□□□.
Answer: C,E (LEAVE A REPLY)

NEW QUESTION: 137
□□ □□□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. □□□□ □□ INSERT □□□□ □□ □□□ □□□□□.
B. □□ □□ NULL □□ □□□ □ □□□□.
C. □□ □□ □□ □□□□ □□□□ □□□ □□ □□□ □□□□□ □ □□□□.
D. □□ □□ □□□ □□□ □□□□ □□□ □ □□□□□ □□□ □ □□□□.
E. UNIQUE □□ □□□ □□ □□□ NULL□ □□□ □ □□□□.
Answer: C,E (LEAVE A REPLY)

NEW QUESTION: 138
COUNT □□□ □□ □□□□ □□ □□?
A. DISTINCT □□□□ □□ COUNT □□□ □□□□ SELECT □□ WHERE □□ □□ □ □□□□.
B. COUNT(DISTINCT inv_amt)□ INV_AMT □□□ □□ □ NULL □□ □□□□ □□ □□□ □ □□ □ □□□□.
C. COUNT □□□ CHAR, VARCHAR2, NUMBER □□□ □□□□ □□□ □ □□□□.
D. COUNT(cust_id)□ CUST_ID □□ □□□ □□ ID□ NULL □□ □□ □□ □□□□ □ □□ □□□□ □.
E. COUNT(*)□ □□ □□ □□□□ □ □□ □□□ □□ NULL □□ □□□□ □ □□ □□□□□.
Answer: B,E (LEAVE A REPLY)

NEW QUESTION: 139
□□□□ □□ □□ □ □□ □□□□ □□□ □□□□□□. (2□□ □□□□□.)

SALES

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(3)
CUST_ID	NOT NULL	NUMBER(4)
TIME_ID		DATE
QTY_SOLD		NUMBER(10,2)

PRODUCTS

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(3)
PROD_NAME		VARCHAR2(30)
PROD_LIST_PRICE		NUMBER(8,2)

SALES □□□□□ PROD_ID□ PRODUCTS □□□□ PROD_ID□ □□□□ □□ □□□□. □ □□ ID □ □□ □□□ □□□□ □□□.

JOINOperator□ □□□ □□ □□□ □□□□□.

```
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;
```

□□□ □□□ □□ □□ □□ □ □□ JOIN □□□ □□□ □ □□□□? (2□□ □□□□□.)

A. □□ □□ □□

B. □□□ □□ □□

C. □□ □□ □□

D. □□

Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 140

MEMBERS □□□□ □□□ □□□□□.

Name	Null?	Type
MEMBER_ID	NOT NULL	VARCHAR2 (6)
FIRST_NAME		VARCHAR2 (50)
LAST_NAME	NOT NULL	VARCHAR2 (50)
ADDRESS		VARCHAR2 (50)
CITY		VARCHAR2 (25)
STATE		VARCHAR2 (3)

A □□□ □□□□ □ □□ □□□ □ □□□ □□ □□ □□□□ □□ □□□□ □□ □□□ □□□□□ □□□.

□□ SQL □□ □□□□ □□□?

A. SELECT * FROM MEMBERS WHERE state LIKE 'A%';

B. SELECT * FROM MEMBERS WHERE state LIKE 'A_';

C. SELECT * FROM MEMBERS WHERE state LIKE 'A_%';
D. SELECT * FROM MEMBERS WHERE state LIKE '%A_';
Answer: (SHOW ANSWER)

NEW QUESTION: 141
INVOICE □□□□ □□□ □□□□ □□ □□□□ □□□□□.

Name	Null	Type
INV_NO	NOT NULL	NUMBER(3)
INV_DATE		DATE
INV_AMT		NUMBER(10,2)

□□ □□□□ □□□ □□ □□□ □□ □□□□ □□ □□ □□? (2□□ □□□□□.)
A. CONCAT(inv_amt, inv_date) : □□□ □□ □□
B. inv_date = '2008□ 2□ 15□' : □□□ □□ □□
C. inv_no BETWEEN '101' AND '110' : □□□ □□ □□
D. inv_date > '01-02-2008' : □□□ □□ □□
E. inv_amt = '0255982' : □□□ □□ □□
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 142
CUSTOMERS □□□□□ NUMBER □□□ □□□ CUST_CREDIT_LIMIT □□ □□□□.
□□ □ □□□ □□□□□ □□□□□?
A. SELECT NVL2(cust_credit_limit, TO_CHAR(cust_credit_limit * .15), '□□□ □ □□') FROM □□;
B. SELECT TO_CHAR(NVL(cust_credit_limit * .15, '□□□ □ □□')) FROM □□;
C. SELECT NVL(TO_CHAR(cust_credit_limit * .15), '□□□ □ □□') FROM □□;
D. SELECT NVL2(cust_credit_limit * .15, '□□ □□') FROM □□;
E. SELECT NVL(cust_credit_limit * .15, '□□□ □ □□') FROM □□;
Answer: (SHOW ANSWER)

NEW QUESTION: 143
□□□□ □□ □□□□ □□ □□? (3□□ □□□□□.)
A. □□□□ □□ □□□□ PRIMARY KEY □□ UNIQUE KEY □□ □□□ □□□□ □□ □□□ □□ □ □□□□□.
B. □□□□ □□□ □□ □□□□ □□ □□□□ □□□□ □□□.
C. □□□□ □□□□ □□ □□□□ □□□□ □□□□□.
D. □□□□□ □□□□ □ DML □□□ □□ □□□ □□ □□ □□□□ □□□□ □□□□□□□.
Answer: A,C,D (LEAVE A REPLY)
□□:
http://viralpatel.net/blogs/understanding-primary-keypk-constraint-in-oracle/

NEW QUESTION: 144

□□□□ □□ CUSTOMERS □□□□ □□□ □□□□□□.

CUSTOMERS

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(6)
CUST_NAME		VARCHAR2(20)
CUST_EMAIL		VARCHAR2(30)
INCOME_LEVEL		VARCHAR2(20)

CUSTOMER_VU□ CUSTOMERS □□□□ □□□ □□□ □□ CUSTOMERS_BR1 □□□□ □□□□ □□ □□□.

□□□ □□ □□ □□□ □□□□□ CUSTOMERS□ □□□□□□ □□□.

□□ MERGE □□ □□□ □□□□□?

```
MERGE INTO customers c
    USING customer_vu     cv
    ON (c.customer_id = cv.customer_id)
WHEN MATCHED THEN
    UPDATE SET
   c.customer_id = cv.customer_id,
   c.cust_name = cv.cust_name,
   c.cust_email = cv.cust_email,
   c.income_level = cv.income_level
WHEN NOT MATCHED THEN
    INSERT VALUES(cv.customer_id,cv.cust_name,cv.cust_email,cv.income_level)
      WHERE cv.income_level >100000;
```

A. INTO □□ □□□ □□ □□□□□□□.

B. CUSTOMER_ID □□ □□□□□ □ □□□□.

C. WHERE □□ INSERT□ □□ □□□ □ □□□□.

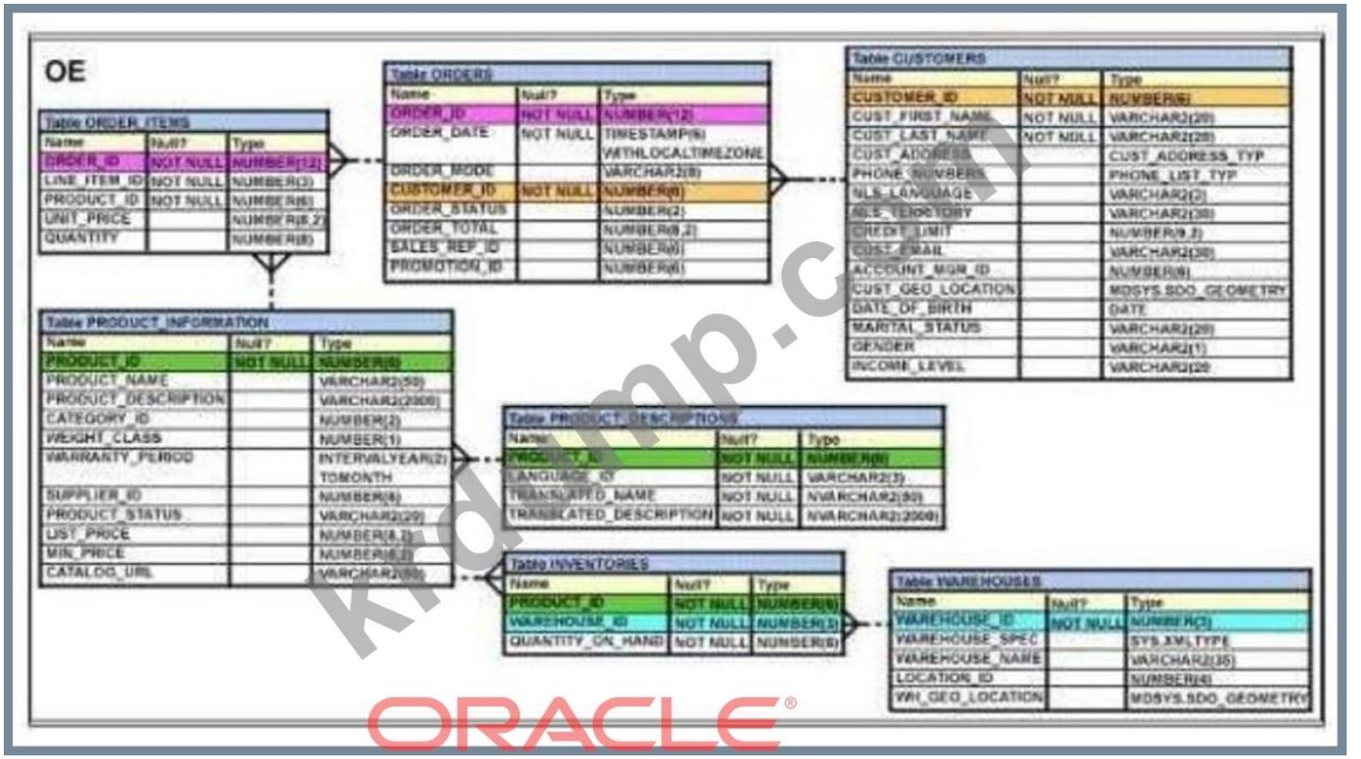
D. CUSTOMER_VU□ □□□ □□□ □□□ □ □□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 145

□□□□ □□ PODUCT_INFORMATION □□□□ □□□ □□□□□□.

(□□ □□ □□ □□□□□.)



PRODUCT_ID □□ □□ □□□□.

□□ □□□ □□□□ □□□□ □□□□□.

SQL > □□□ □□ upper_name_idx

ON product_information(UPPER(product_name));

PRODUCT_INFORMATION □□□□ □□ □□□□ □□□□.

□□ □□□ UPPER_NAME_IDX □□□□ □□□□□?

A. SELECT UPPER(product_name)FROM product_informationWHERE product_id = 2254;

B. SELECT UPPER(product_name)FROM product_information;

C. SELECT product_idFROM product_informationWHERE UPPER(product_name) IN ('LASERPRO', 'CABLE');

D. SELECT product_id, UPPER(product_name)FROM product_informationWHERE UPPER(product_name) = 'LASERPRO' □□ list_price > 1000;

Answer: C (LEAVE A REPLY)

NEW QUESTION: 146

□□ □□□ □□□□ PRODUCTStable□ □□□□□.

SQL > DROP TABLE □□;

□ □□□ □□□ □□ □ □□ □□ □ □□ □□?

A. □□□□ □□ □□ □□□□□ □□□□□.

B. □□□□ □□ □□ □□□□ □□ □□□ □□□□□□.

C. □□□ □□□ □□ □□ □□□□ □□□□□.

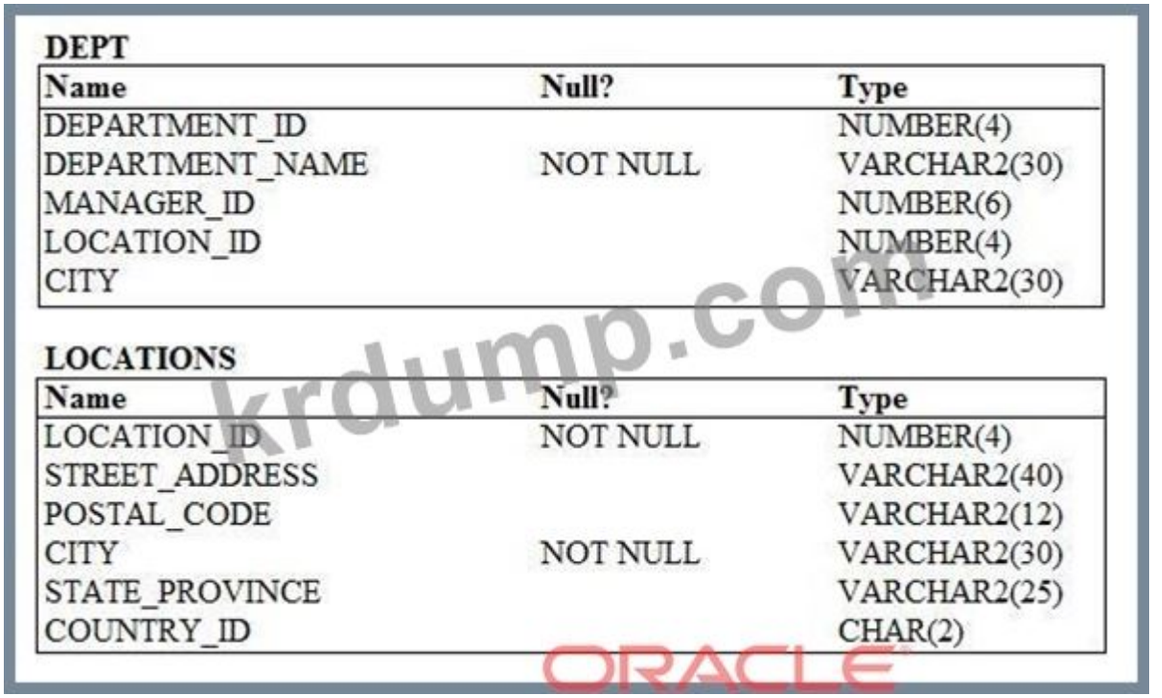
D. □□□□ □□ □□□□ □□□□□ □□□ □□□ □□□ □□□□□.

E. □□□□ □□ □□□□ □□ □□□ □□□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 147

□□□□ □□ DEPT □ LOCATIONS □□□□ □□ □□□ □□□□□□.



DEPT

Name	Null?	Type
DEPARTMENT_ID		NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)
CITY		VARCHAR2(30)

LOCATIONS

Name	Null?	Type
LOCATION_ID	NOT NULL	NUMBER(4)
STREET_ADDRESS		VARCHAR2(40)
POSTAL_CODE		VARCHAR2(12)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_ID		CHAR(2)

□ □□□ □□ LOCATIONS □□□□ CITY □□ □□ □□ □□□ □□ □□ □□ DEPT □□□□ CITY □□ □□□□□□□ □□□.

□□□ □□□□ □□ □□ SQL □□ □□□□□□□□?

A. □□□□ □□ dSET □□ = (SELECT cityFROM □□ lWHERE d.location_id = l.location_id);

B. □□□□ □□ dSET □□ = ALL (SELECT cityFROM □□ lWHERE d.location_id = l.location_id);

C. □□□□ □□ dSET □□ = (SELECT cityFROM □□ l)WHERE d.location_id = l.location_id;

D. □□□□ □□ dSET □□ = ANY(SELECT cityFROM □□ l)

Answer: A (LEAVE A REPLY)

NEW QUESTION: 148

Oracle □□□□□□□ □□□, □□ □□□ □□ □ □□□□□ □□□ □□□ □□ □□□□ □□ □ □?

A. SESSIONTIMEZONE □□□ □□ □□□(UTC)□□ □□□□ □□□ □ □□□□.

B. TIMESTAMP WITH LOCAL TIMEZONE □□□ □□ □□□ □□ □□□ □□□ □□□□ □□□□ □□□□□□□ □□□□□.

C. TIMESTAMP □□□ □□ □□□ □□, □, □□ □□ □□□ □□□□□.

D. CURRENT_TIMESTAMP □□□ □□□ □□ □□ □□□□ □□□□□.

E. DBTIMEZONE □□□ UTC(□□ □□□)□□ □□□□ □□□ □ □□□□.

Answer: A,B,E (LEAVE A REPLY)

NEW QUESTION: 149

□□ ALTER TABLE□□ □□□□□□.

ALTER TABLE □□

SET □□□(order_date);
□□ □□ □□□□?
A. ROLLBACK□ □□□□ ORDERS □□□□ ORDER_DATE □□ □□ □□□ □ □□□□.
B. ALTER TABLE □□□ □□□ □ ORDER_DATE□□ □ □□ ORDERS □□□□ □□□ □ □□□ □.
C. ALTER TABLE □□□ □□□□□ □□□□□ ORDER_DATE □□ □□ □□□ □□□.
D. DESCRIBE □□□ □□□ ORDER_DATE □□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 150
□□ □□□□ □□□□ □□ □□□□ DML □□□ □□□□ □□ □□ □ □□ □□□□ □□ □□?
A. □□□ □□ □□□ □□ □□ □□ □□□ □ □□□□.
B. WITH CHECK □□ □□ □□ □□ □□□□□ □□ □□□ □ □□□ □□□□.
C. □□□□ PRIMARY KEY□ □□ PRIMARY KEY □□ □□ □□ □□□□ □□□□ □□ □□ □□ □ □□□ □□ □□□□ □ □□ □□□ □ □□□□.
D. Insert □□ □□ □□ □□ □□□□ □□□ □ □□□□.
E. □□ □□ □□□ DISTINCT □□□□ □□□ □□ □□ □□□□ □□ □□□□□ □□□□ □ □□ □□□ □ □□□□.
F. □□□□ □□ □□ □□□□ □□□□ □□ □□□□ □□ NOT NULL □□ □□□ □□ □□ □□ □ □ □□ □□□□ □□ □□□□ □□ □□□ □ □□□□.
Answer: C,E,F (LEAVE A REPLY)

NEW QUESTION: 151
□□□□□ □ □□□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. ALTER TABLE □□□ □□□□ □□ □□ □□□ □□□ □ □□□□.
B. □□ □□□□□ □□□□ □□ □□□□□ □□□□□ □□□.
C. □□ □ □□ □□□ □ □□□□.
D. □□ □□□□ □□ □□□ □□□□ □□ □□□ □ □□□□.
E. □ □□□ □□□□□ □□□□□.
F. □□ □□□□ □□ □□□□ □□□ □ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 152

□□ SQL □□ □□□□□.

```
DELETE FROM employees e
 WHERE EXISTS
    (SELECT 'dummy'
       FROM emp_history
      WHERE employee_id = e.employee_id);
```

□□ □□ □□□□□? (2□□ □□□□□.)

A. EMPLOYEEStable□ □□ □□ □□ □□ □□□ □□□□□.

B. □□ □□□ □□ □□ □□□ □□□□.

C. DELETE□□ □□□□ □□ □□ □□□ □□□□□.

D. □□ □□□ □□ □□ □□□□□□ DELETE □□ □□□□□ □□□□□.

E. EMPLOYEEStable□ □□ □□ □□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 153

SYSDATE □□□ □□ Oracle Server □□□ □□□ □□ □□□□□.

21 - 5□ 19□

□□□ □□□ □□ □□□□□ □□□.

201□ 5□ 21□ □□□ 9

□□ □□□ □□□ □ □□□□?

A. SELECT TO_ CHAR(SYSDATE, 'FMDAY, DDTH MONTH, YYYY') FROM DUAL;

B. SELECT TO _ CHAR(SYSDATE, ' FMDAY, DD MONTH, YYYY') FROM DUAL;

C. SELECT TO _ DATE(SYSDATE, ' FMDAY, DD MONTH, YYYY') FROM DUAL;

D. SELECT TO_ CHAR(SYSDATE, ' FMDD, DAY MONTH, YYYY') FROM DUAL;

Answer: B (LEAVE A REPLY)

NEW QUESTION: 154

□□ □□□ □□ □□□□ □□ □□? (3□□ □□□□□.)

A. □□ □□ □□□ □□□ □□□ □ □□□□.

B. □□□ □□□□ □□□ □ □□□□.

C. SQL □□ SELECT □□□ □□□ □□□ □□□ □ □□□□.

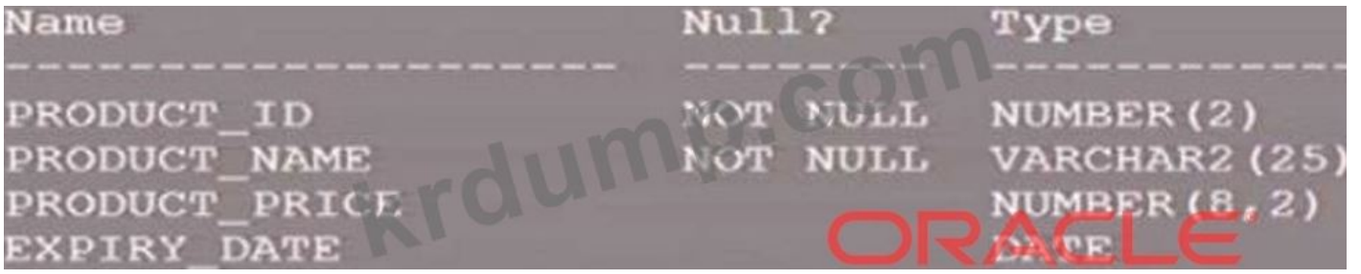
D. SQL □□ SELECT □□□ □□ □ □□□ □□ □□□ □ □□□□.

E. GROUP BY □□ □□ SQL □□□□ □□□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 155

PRODUCT_ DETAILS □□□□ □□ □□□ □□□□□.



□□ □ □□ □□□ □□□□□?

A. EXPIRY_ DATE□ □□ □□□□ □□□ □ □□□□.

B. PRODUCT_ ID□ PRIMARY KEY □□ □□□ □□□ □ □□□□.

C. PRODUCT_ PRICE□□ □□ □□□□ □□ □□ □□□□□ 0 □□ □□□□□.

D. PRODUCT_ PRICE□ □□□ □□ □□ □□□□ □□ □□□□□ □□□ □ □□□□.

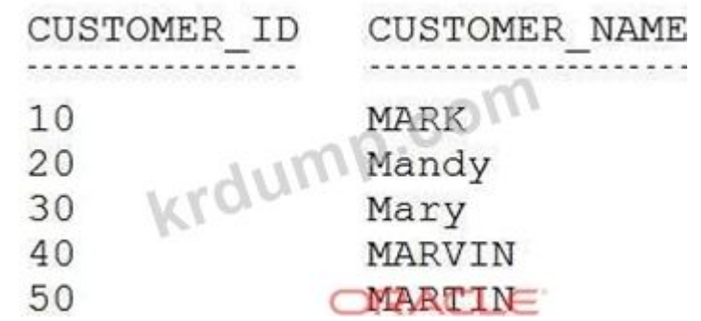
E. PRODUCT_ NAME□□ □□ □□ □□□ □ □□□□.

F. EXPIRY_ DATE□□ □□□ □□□□ □□ □□ □□□□□ SYSDATE□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 156

CUSTOMERS □□□□ □□ □□□ □□□□□.



□ □□□ □□□□ □ □□ SELECT □(2□ □□):

A. SELECT customer_name FROM □□ WHERE customer_name LIKE '*Ma*';

B. SELECT customer_name FROM □□ WHERE UPPER (customer_name) LIKE 'MA*';

C. SELECT customer_name FROM □□ WHERE UPPER(customer_name) LIKE 'MA%';

D. SELECT customer_name FROM □□ WHERE customer_name LIKE 'Ma%';

E. SELECT customer_name FROM □□ WHERE customer_name LIKE '%a%';

F. SELECT customer_name FROM □□ WHERE customer_name LIKE 'Ma*';

G. SELECT customer_name FROM □□ WHERE customer_name = '*Ma*';

Answer: (SHOW ANSWER)

□□ □□□ □□ UPPER □□□ □□□□ □□□□□ □□□□ □□ □□□ □□□ □ □□□□.

UPPER(last_name) LIKE 'SM%'

□□: https://docs.oracle.com/cd/B12037_01/server.101/b10759/conditions016.htm

NEW QUESTION: 157
Oracle Database□ □□□ SQL □□□ □□□□ □□□ □ □□ □ □□ □□□ □□□□□?
A. □□□□ □□ □□□□ □□ □□
B. □□□□ □□ □ □□
C. □□□□ □ □ □□□ □ □□ □□□□ □□ □□ □□
D. □□□ □□□□ □□□□ □□□ □□□□ □□
Answer: A,B,D (LEAVE A REPLY)