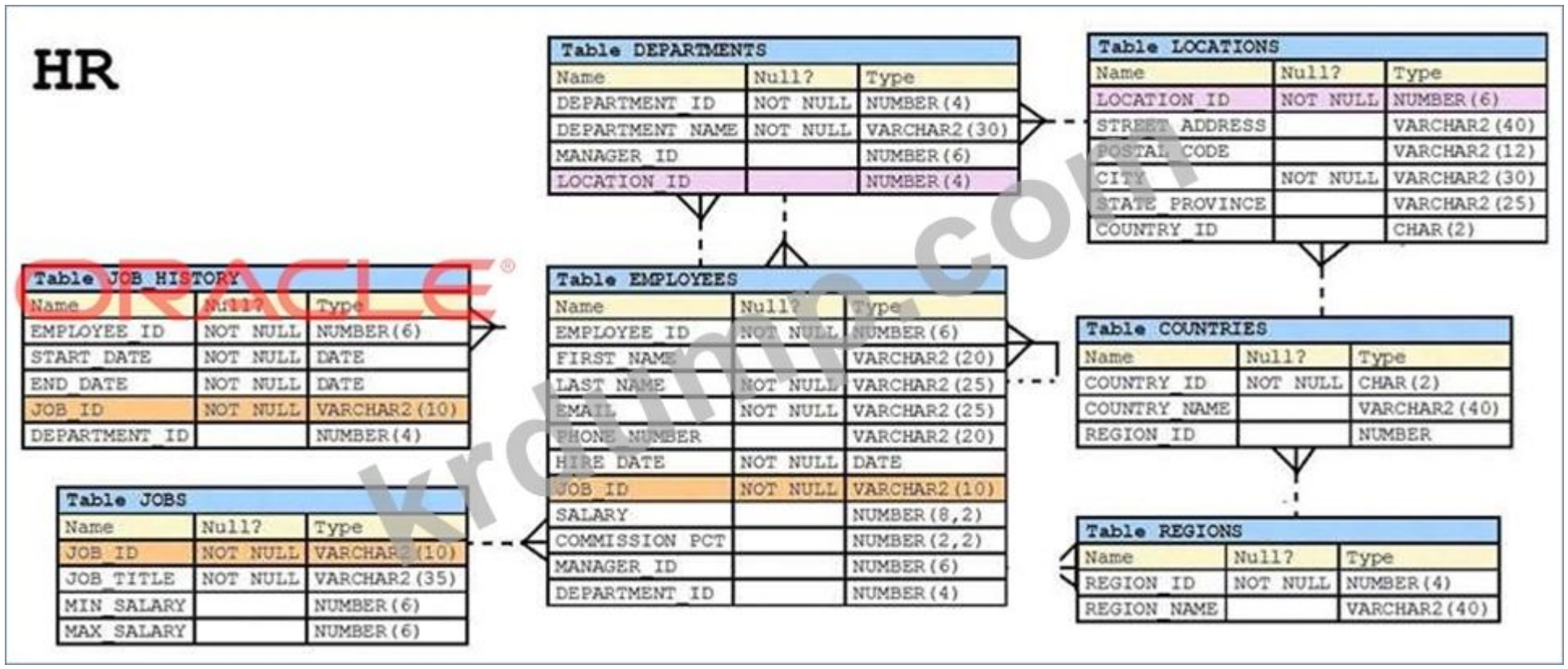
Oracle.1z0-071.v2022-08-23.q100

□□□□:	1z0-071
□□□□:	Oracle Database SQL
□□□:	Oracle
□□ □□ □□□:	100
□□:	v2022-08-23
# □□ □:	1872
# □□ □□□:	1000
https://www.krdump.com/Oracle.1z0-071.v2022-08-23.q100.html	

NEW QUESTION: 1

□□□□ □□ EMPLOYEES □□□□ □□□ □□□□□□.



MANAGER_ID□ □□□□ 100□□ □□ □□□ □□ □□□□ □□□□□ □□□.

□ □□ □□ □□□ □□□□. □ □□ □□□ □□□ □□□□ LAST_NAME□ □□□ □□ □ □□ □□□ □□□ LAST_NAME□ □□□ □□□.

□□ SQL □□ □□□□□□□□?

A. SELECT m.last_name "Manager", e.last_name "Employee"FROM □□ m JOIN □□ eON m.employee_id = e.manager_idWHERE m.manager_id=100;

B. SELECT m.last_name "Manager", e.last_name "Employee"FROM □□ m JOIN □□ eON e.employee_id = m.manager_idWHERE m.manager_id=100;

C. SELECT m.last_name "Manager", e.last_name "Employee"FROM □□ m JOIN □□ eON m.employee_id = e.manager_idWHERE e.manager_id=100;
D. m.last_name "Manager", e.last_name "Employee"FROM □□ □□ m JOIN □□ eWHERE m.employee_id = e.manager_id AND e.manager_id=100
Answer: (SHOW ANSWER)

NEW QUESTION: 2
□□ □ □□ □□□□ □□□□□ □□□□□?
A. PL/SQL □□ □□□ □□□ □
B. DELETE □□ □□□ □
C. ROLLBACK □□□ □□□ □
D. □□□ □□ □□(DDL) □□ □□□ □
E. □□ □□ □□□□ □□□ TRUNCATE □□ □□□ □
Answer: C,D,E (LEAVE A REPLY)
□□:
https://docs.oracle.com/cd/B19306_01/server.102/b14220/transact.htm

NEW QUESTION: 3
□□□□ □□ ORDER_ITEMS □□□□ □□ □□□ □□□□□□.

ORDER_ID	LINE_ITEM_ID	PRODUCT_ID	UNIT_PRICE	QUANTITY
2356	2	2274	148.5	34
2356	7	2316	22	55
2356	8	2323	18	55
2356	5	2308	58	47
2356	6	2311	95	51
2356	1	2264	199.1	38
2357	7	2276	236.5	38
2357	8	2289	48	41
2357	1	2211	3.3	140
2357	4	2257	371.8	29
2357	6	2268	75	32
2357	2	2245	462	26
2357	3	2252	788.7	26
2357	5	2262	95	29
2358	4	1803	55	13
2358	3	1797	316.8	12
2358	5	1808	55	14

□□ SQL □□ □□□□□.
□□ 1:
SELECT MAX(□□*□□) "□□ □□□"
□□_□□□□;

□□□ 2:
SELECT MAX(□□*□□) "□□ □□□"
□□_□□□□
GROUP BY order_id;
□ SQL □□ □□□ □□ □□□□ □□ □□? (□□□□ □□ □□ □□□□□□.)
A. □ □□□ □□ □□□ □□□ □□□□□.
B. □□□ 1□ □□□ □□□ □□□□ □□□□.
C. □□□ 2□ □□ □□ □□□ □□□□□.
D. □ 1□ □□□ □ □□ □□□□□.
E. □ □ □□ UNIT_PRICE □ QUANTITY □□ □□ NULL □□ □□□□□.
Answer: C,D,E (LEAVE A REPLY)

NEW QUESTION: 4
BOOK_SEQ□ □□□□ □□ □□□□□□.
□□ □ □□ CREATE TABLE □□□ □□□□□?
A)

```
CREATE TABLE bookings (
  bk_id       NUMBER(4) DEFAULT book_seq.CURRVAL,
  start_date  DATE      DEFAULT SYSDATE,
  end_date    DATE      DEFAULT start_date);
```

B)

```
CREATE TABLE bookings (
  bk_id       NUMBER(4) DEFAULT book_seq.NEXTVAL PRIMARY KEY,
  start_date  DATE      DEFAULT SYSDATE,
  end_date    DATE      DEFAULT SYSDATE NOT NULL);
```

C)

```
CREATE TABLE bookings (
  bk_id       NUMBER(4) NOT NULL DEFAULT book_seq.CURRVAL,
  start_date  DATE      NOT NULL,
  end_date    DATE      DEFAULT SYSDATE);
```

D)

```
CREATE TABLE bookings (
  bk_id       NUMBER(4),
  start_date  DATE      DEFAULT SYSDATE,
  end_date    DATE      DEFAULT (end_date >= start_date));
```

□□□)

```
CREATE TABLE bookings (
  bk_id       NUMBER(4) NOT NULL PRIMARY KEY,
  start_date  DATE      NOT NULL,
  end_date    DATE      DEFAULT SYSDATE);
```

A. □□ D
B. □□ B
C. □□ E
D. □□ A
E. □□ C

Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 5
UNION□ □□ SET □□□□ □□□ SQL □□ □□ □□□ □ ORDER BY □□ □□ □□□□ □□ □□?
A. UNION □□□□ □□□□ □□ □□□ □ □□ SELECT□ □□ □ □□ □□ □□□□□ ORDER BY □□ □□ □ □□□ □□□□ □ □□□□□.
B. □□ □□□ □ □□ SELECT □□ □ □□□ □□□□□.
C. □□ □□□ □ SELECT □□□ □□□ ORDER BY □□ □□ □ □□□□.
D. □□ □□□ □ SELECT □□□ □□ ORDER BY □□ □□□ □□□.
E. ORDER BY □□□ □ □□□ □□□□ □□□.
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 6
INVOICE □□□□ □□□ □□□□□□.

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

□□ □ □□ SQL □□ □□□□□ □□□□□?
SELECT inv_no, NVL2(inv_date, '□□ □', '□□□')
A. □□□□;
SELECT inv_no, NVL2(inv_amt, inv_amt*.25, '□□□ □ □□')
B. □□□□;
C. □□□□;
SELECT inv_no, NVL2(inv_date, sysdate-inv_date, sysdate)
D. □□□□;
SELECT inv_no, NVL2(inv_amt, inv_date, '□□□ □ □□')
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 7
□ □□ DROP □□□ □□ □□□ □□□□ PRODUCTS □□□□□ □□□□□.
DROP TABLE □□ PURGE;
□□ □□ □ □□□ □□□□ FLASHBACK □□□ □□□□□.
FLASHBACK TABLE □□□ □□□□ □□;
FLASHBACK □□□ □□□ □□ □□□□ □□ □□ □□?
A. □□□ □□□ □□□□□.
B. □□□ □□, □□□, □□□□ □□□□□.
C. □□□ □□ □ □□□□ □□□□□ □□ □□□□ □□□□ □□□□.

D. □□□ □□, □□□ □□ □□ □□□□ □□□ □ □□□□.

Answer: D (LEAVE A REPLY)

□□

https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9003.htm

NEW QUESTION: 8

□□□□ □□ 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

USINGmonth_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_MASTERtable□□ ORDER_IDs1, 2, 3 □ 4□ □□□□□.

B. ORDERS_MASTERtable□□ ORDER_IDs1, 2 □ 4□ □□□□□.

C. ORDERS_MASTERtable□ ORDER_IDs1, 2 □ 3□ □□□□□.

D. ORDERS_MASTER □□□□□ ORDER_IDs1 □ 2□ □□□□□.

Answer: (SHOW ANSWER)

□□/□□:

□□:

https://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9016.htm

NEW QUESTION: 9
EMPLOYEES_ID □ LOGIN_ID □□ □□ null□ □□□ □□□□ □□ □□□ EMPLOYEES□ □□□□ □□□.
□□ □ □□□ □□□□ □□□□ □ □□ SQL □□ □□□□□? (2□□ □□□□□.)
A. CREATE TABLE □□
(□□ ID NUMBER,
□□□ ID NUMBER,
□□ □□ VARCHAR2(100),
□□ □□ DATE,
CONSTRAINT emp_id_uk UNIQUE(employee_id, login_id);
CONSTRAINT emp_id_nn NOT NULL (employee_id, login_id));
B. CREATE TABLE □□
(□□ ID NUMBER,
□□□ ID NUMBER,
□□ □□ VARCHAR2(25),
□□ □□ DATE,
CONSTRAINT emp_id_pk □□ □(employee_id, login_id));
C. CREATE TABLE □□
(□□ ID NUMBER CONSTRAINT emp_id_nn NOT NULL,
login_id NUMBER CONSTRAINT login_id_nn NOT NULL,
□□ □□ VARCHAR2(100),
□□ □□ DATE,
CONSTRAINT emp_num_id_uk UNIQUE (employee_id, login_id));
D. CREATE TABLE □□
(□□ ID NUMBER,
□□□ ID NUMBER,
□□ □□ VARCHAR2(100),
□□ □□ DATE,
CONSTRAINT emp_id_uk UNIQUE (employee_id, login_id));
E. CREATE TABLE □□
(□□ ID NUMBER CONSTRAINT emp_id_pk □□ □,
□□□ ID NUMBER □□,
□□ □□ VARCHAR2(25),
□□ □□ □□);
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 10
□□ □□□□ □□□□□□.

```
SELECT 1 AS id, 'John' AS first_name, NULL AS commission
  FROM dual
 INTERSECT
SELECT 1, 'John', null
  FROM dual
 ORDER BY 3;
```

□□ □ □□□ □□□□□?

A. □□

B. 0□

C. 2□

D. 1□

Answer: (SHOW ANSWER)

NEW QUESTION: 11

□□□□□ □ □□□ □□ □□□□ □□ □□?

A. □□ □□□□ □□ □□□ □□□□ □□ □□□ □ □□□□.

B. □ □□□ □□□□□ □□□□□.

C. □□ □□□□□ □□□□ □□ □□□□□ □□□□□ □□□.

D. □□ □ □□ □□□ □ □□□□.

E. ALTER TABLE □□□ □□□□ □□ □□ □□□ □□□ □ □□□□.

F. □□□□ □□□□ □□ □□□□ □□ □□□ □ □□□□.

Answer: D,E (LEAVE A REPLY)

NEW QUESTION: 12

□□ SQL □□ □□□□□.

SQL> cust_id, cust_last_name "□" □□

□□□□□□

□□ ID = 10

□□ □□

cust_id CUST_NO, cust_last_name □□

□□□□□□

□□□ country_id = 30

□□□ □□□ □ □□ ORDERBY□ 3□□ □□□□□□.

A. CUST_NO □□

B. ORDER BY "□"

C. "CUST_NO"□ □□

D. ORDER BY 2, cust_id

E. 2, 1□ □□

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

NEW QUESTION: 13

□□ □□□□ □□ □□□□ □□ □□ □□?

A. DML □□ □□□ □ □□□□.
B. SQL □□ PL/SQL□ □□□□ □□ □□□□ □□□ □ □□□□.
C. □□□□ □□□ □ □□□□.
D. CREATE TABLE AS SELECT □□□ □□□□ □□□□□□□ □□ □□□□□ □□□□ □□ □ □□□□.
E. □□□□□□ □□ □□□□ □□ □□□□□□ □□□ □□□□□.
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 14
□□□ HR□ CREATE SESSION, CREATE ANY TABLE □ UNLIMITED TABLESPACE □□□ □□□□.
□□□ SCOT□ CREATE SESSION, CREATE TABLE □ UNLIMITED TABLESPACE □□□ □□□□.
HR□ □□ □□□□ □□□□□ □□□□□.

```
CREATE TABLE scott.products (
  prod_id NUMBER(2),
  prod_name VARCHAR2(20));
```

HR□□ □□:
1. scott.products □□ □□(1, '□□□');
SCOTT□□ □□:
2. □□□□ *□ □□□□□□.
3. scott.products □(2, 'HDD')□ □□
4. □□□ □□ □□□ □□;
□□ □□ □□□□□ □□□□□?
A. 2 □ 3□
B. 2, 3, 4
C. 1, 2 □ 3
D. 1□□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 15
□□□□ □□ 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

'□□□' □□□□□ □□□ □□□□ □□□ □□□ □□ □□□□□ □□□□ □□□ □□ □□□ □□□□ □□□□ □□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?

A. SELECT promo_name, promo_begin_date FROM □□□□
WHERE promo_begin_date> ALL(SELECT MAX(promo_begin_date)
□□□□□□) □□□
promo_category= '□□□';

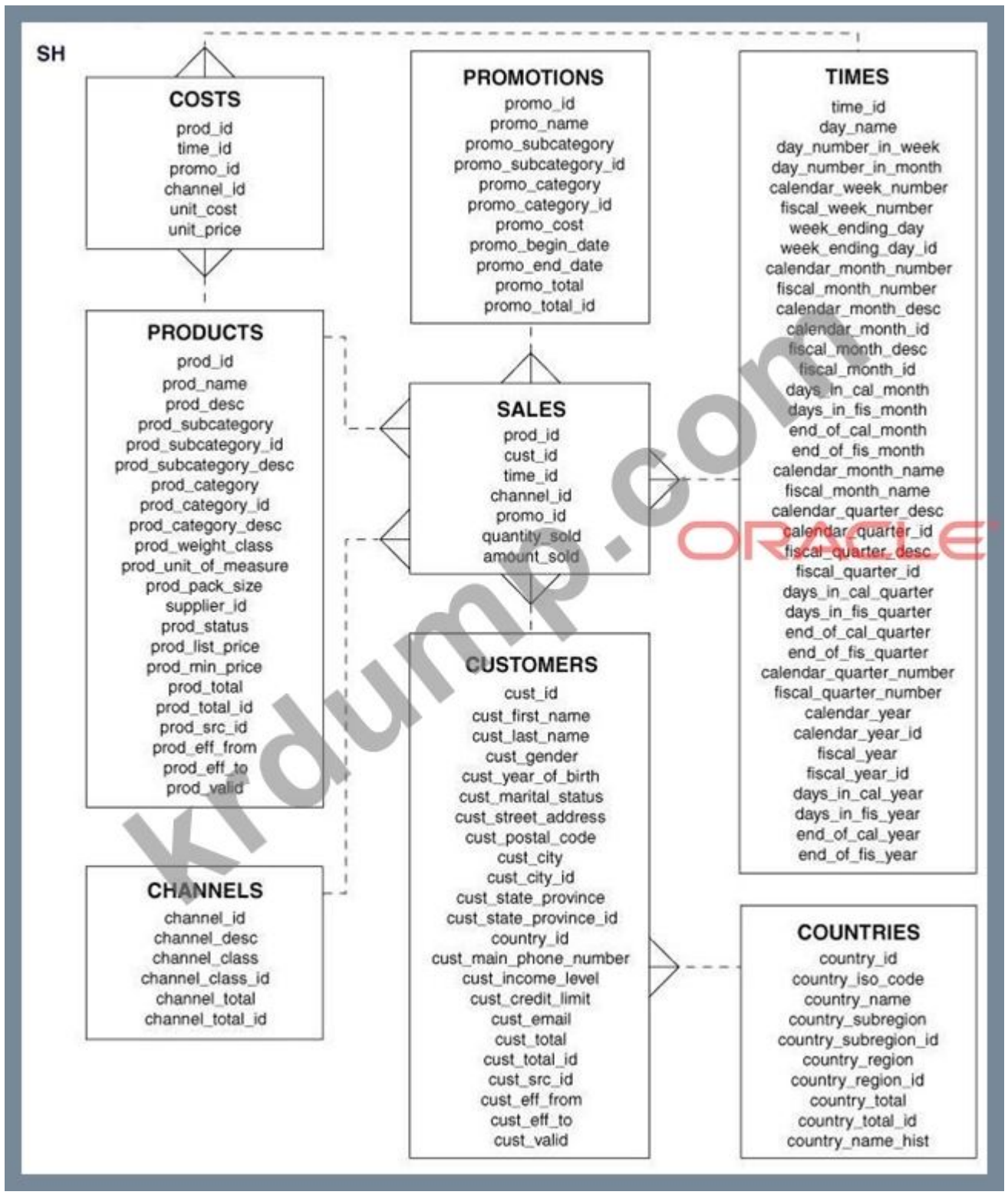
B. promo_name, promo_begin_date FROM □□□□ □□
WHERE promo_begin_date > ALL(promo_begin_date □□
□□□□□□
WHERE promo_category = '□□□');

C. promo_name, promo_begin_date FROM □□□□ □□
WHERE promo_begin_date> ANY (SELECT promo_begin_date
□□□□□□
WHERE promo_category= '□□□');

D. SELECT promo_name, promo_begin_date FROM □□□□
WHERE promo_begin_date IN (promo_begin_date □□
□□□□□□
WHERE promo_category= '□□□');

Answer: (SHOW ANSWER)

NEW QUESTION: 16
□□□□ □□ SALES □ CHANNELS □□□□ □□ □□□ □□□□□□.



□□ SQL □□ □□□□□□.

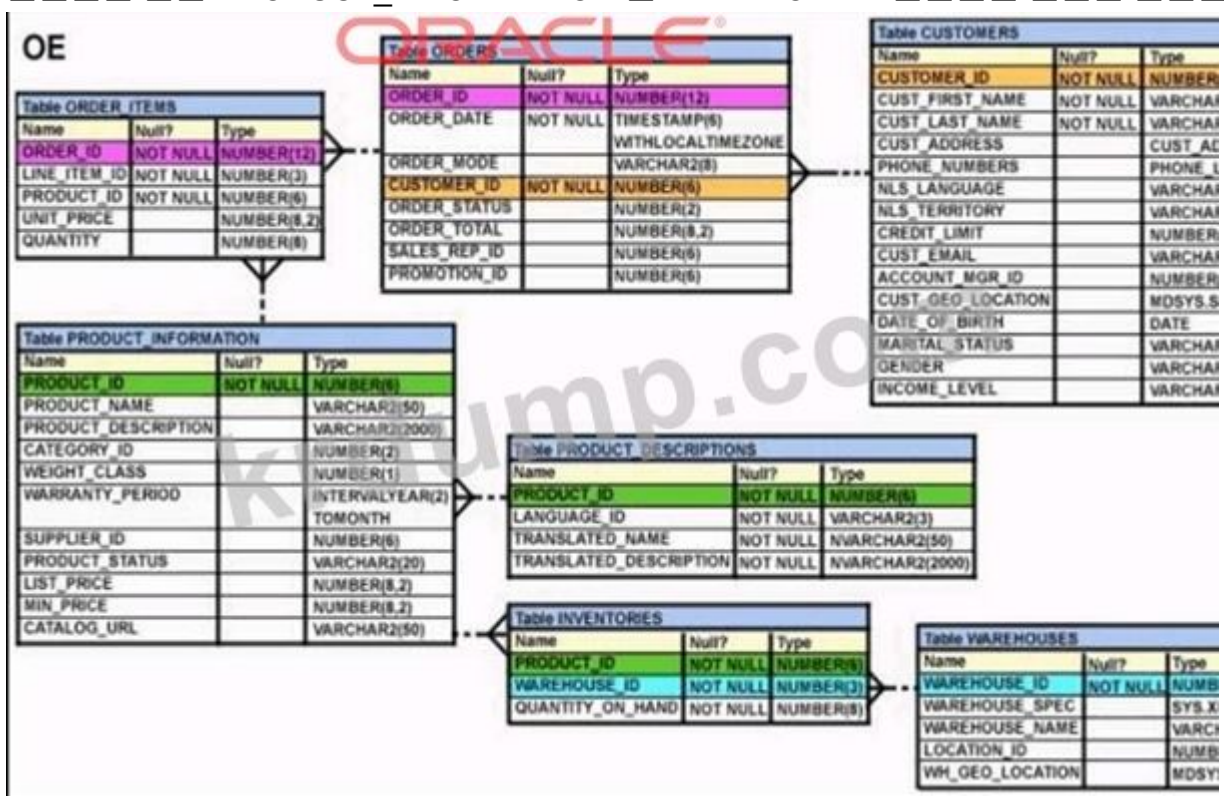
```
INSERT INTO sales VALUES (23, 2300, SYSDATE,
                          (SELECT channel_id
                           FROM channels
                           WHERE channel_desc='Direct Sales'),
                          12, 1, 500);
```

□□□ □□ □□□□ □□ □□?

A. VALUES □□ □□ □□□ □□□□□□ □□□ □□ □□□ □□ □□□□□.

B. □□ □□□ VALUES □□ □□□□ □□ □□□ □□□□ □□□□□.
C. VALUES □□□ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□□.
D. □□□□ □□□□ SALES □□□□ □ □□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 17
□□□□ □□ PRODUCT_INFORMATION □ INVENTORIES □□□□ □□□ □□□□□□.



QUANTITY_ON_HAND□ 5 □□□ □□ □□□ □□ PRODUCT_ID, SUPPLIER_ID □ QUANTITY_ON_HAND□ □□□ □□□ □□□□ □ □□ □□□ □□ □□□ □□□□.
□□□ □□□ □ □□ □ □□ SQL □□ □□□□□? (2□□ □□□□□.)
SELECT i.product_id, i.quantity_on_hand, pi.supplier_id
A. □□ □□□□
NATURAL JOIN □□ □ □□_□ < 5;
SELECT i.product_id, i.quantity_on_hand, pi.supplier_id
B. FROM product_information pi JOIN □□□□ i
ON(pi.product_id=i.product_id)
USING (product_id) AND quantity_on_hand < 5;
C. FROM product_information pi JOIN □□□□ i

ON (pi.product_id=i.product_id) AND quantity_on_hand < 5;
SELECT i.product_id, i.quantity_on_hand, pi.supplier_id
D. FROM product_information pi JOIN □□□□ i
ON(pi.product_id=i.product_id)
WHERE □□_□ < 5;
SELECT product_id, quantity_on_hand, Supplier_id
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 18
AMOUNT_SOLD□ □□ □□ □□□ □□ SALES □□□□□ □□ 5%□ □□□□ □□□ □□ 5%□ □□□□ □□□□ □□□
AMOUNT_SOLD□ □□ □□ □□□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?
SELECT prod_id, cust_id, amount_sold
A. □□□□
ORDER BY amount_sold
□□□□ □□ □ □□ 5% □□ □□□□□.
SELECT prod_id, cust_id, amount_sold
B. FROM □□
ORDER BY amount_sold
□□ 5% □□ □□□□□.
SELECT prod_id, cust_id, amount_sold
C. FROM □□
ORDER BY amount_sold
□□□□ □□ □□□□ □□ 5% □□ □□□□□.
SELECT prod_id, cust_id, amount_sold
D. □□□□
ORDER BY amount_sold
□□□ □□ □ □□ 5% □□ □□□□□.
SELECT prod_id, cust_id, amount_sold
Answer: A (LEAVE A REPLY)

NEW QUESTION: 19
□□ □□□□ □□□□□□.

```
SELECT 1 AS id, 'John' AS first_name, NULL AS commission
  FROM dual
 INTERSECT
SELECT 1, 'John', null
  FROM dual
 ORDER BY 3;
```

□□ □ □□□ □□□□□?
A. 2□
B. □□
C. 0□
D. 1□

Answer: D (LEAVE A REPLY)

NEW QUESTION: 20
□□□□ NLS_DATE_FORMAT□ DD_MM_YYYY□□□.
□□□ 86400□□ □□□□.
□ □□□ □□□□□□.

```
DATE
-----------
02-JAN-2020
```

□□ □□□ □□□ □□□□□?
A. SELECT TO_CHAR(TO_DATE('29-10-2019') + INTERVAL '2' MONTH + INTERVAL '6' DAY INTEVAL '120 SECOND DD-MON-YYYY) AS "DATE"
B. SELECT TO_CHAR(TO_DATE('29-10-2019') + INTERVAL '2' MONTH + INTERVAL '5' DAY INTEVAL '120 SECOND DD-MON-YYYY) AS "DATE"
C. SELECT TO_CHAR(TO_DATE('29-10-2019') + INTERVAL '2' MONTH + INTERVAL '6' DAY INTEVAL '120 SECOND DD-MON-YYYY) AS "DATE"
D. SELECT TO_CHAR (TO_DATE ('29-10-2019') + INTERVAL '2' MONTH + INTERVAL '4' DAY INTEVAL '120 SECOND DD-MON-YYYY) AS "DATE"
E. SELECT TO_CHAR (TO_DATE ('29-10-2019') + INTERVAL '3' MONTH + INTERVAL '7' DAY INTEVAL '120 SECOND DD-MON-YYYY) AS "DATE"
Answer: E (LEAVE A REPLY)

NEW QUESTION: 21
CUSTOMERS □□□□ CUST_NAME □□ □□ □□□□ □□□□□□.
CUST_NAME

□□□ □□□□
□□□ □□
□□□ □□□
□□ MCE□
□□□ □□□□□
□□□ □□□
□ □□ □□□ "Mc" □□ "MC"□ □□□□ □□□ □ □□ □□□ □□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?
A. SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)FROM □□WHERE INITCAP(SUBSTR(cust_name, INSTR(cust_name,' ')+1))='Mc';
B. SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)FROM □□WHERE SUBSTR(cust_name, INSTR(cust_name,' ')+1) LIKE INITCAP('MC%');
C. SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)FROM □□WHERE INITCAP(SUBSTR(cust_name, INSTR(cust_name,' ')+1)) = INITCAP('MC%');
D. SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)FROM □□WHERE

INITCAP(SUBSTR(cust_name, INSTR(cust_name,' ')+1)) LIKE 'Mc%';
Answer: D (LEAVE A REPLY)

NEW QUESTION: 22
□□□□ □□ PROJ_TASK_DETAILS □□□□ □□□□ □□□□□□. (□□ □□ □□ □□□□□.)

PROJ_TASK_DETAILS

TASK_ID	BASED_ON	TASK_IN_CHARGE	TASK_START_DATE	TASK_END_DATE
P01		KING	10-SEPT-07	12-SEPT-07
P02	P01	KOCHAR	13-SEPT-07	14-SEPT-07
P03		GREEN	14-SEPT-07	18-SEPT-07
P04	P03	SCOTT	19-SEPT-07	20-SEPT-07

PROJ_TASK_DETAILS □□□□ □□□□ □□□ □□ □□ □□□ □□ □□□ □□□□□.
BASED_ON □□ □□ □□ □□□□ □□□□□.
□□ □□□ □□ □□□ □□□ □□□□ □□□□.
□□ □□ ID, □□□ □□□ □□ ID □ □□□ □□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?

A. SELECT p.task_id, p.based_on, d.task_in_chargeFROM proj_task_details p LEFT OUTER JOIN proj_task_details dON (p.based_on = d.task_id);
B. SELECT p.task_id, p.based_on, d.task_in_chargeFROM proj_task_details p JOIN proj_task_details dON (p.based_on = d.task_id);
C. SELECT p.task_id, p.based_on, d.task_in_chargeFROM proj_task_details p JOIN proj_task_details dON (p.task_id = d.task_id);
D. SELECT p.task_id, p.based_on, d.task_in_chargeFROM proj_task_details p FULL OUTER JOIN proj_task_details dON (p.based_on = d.task_id);
Answer: A (LEAVE A REPLY)

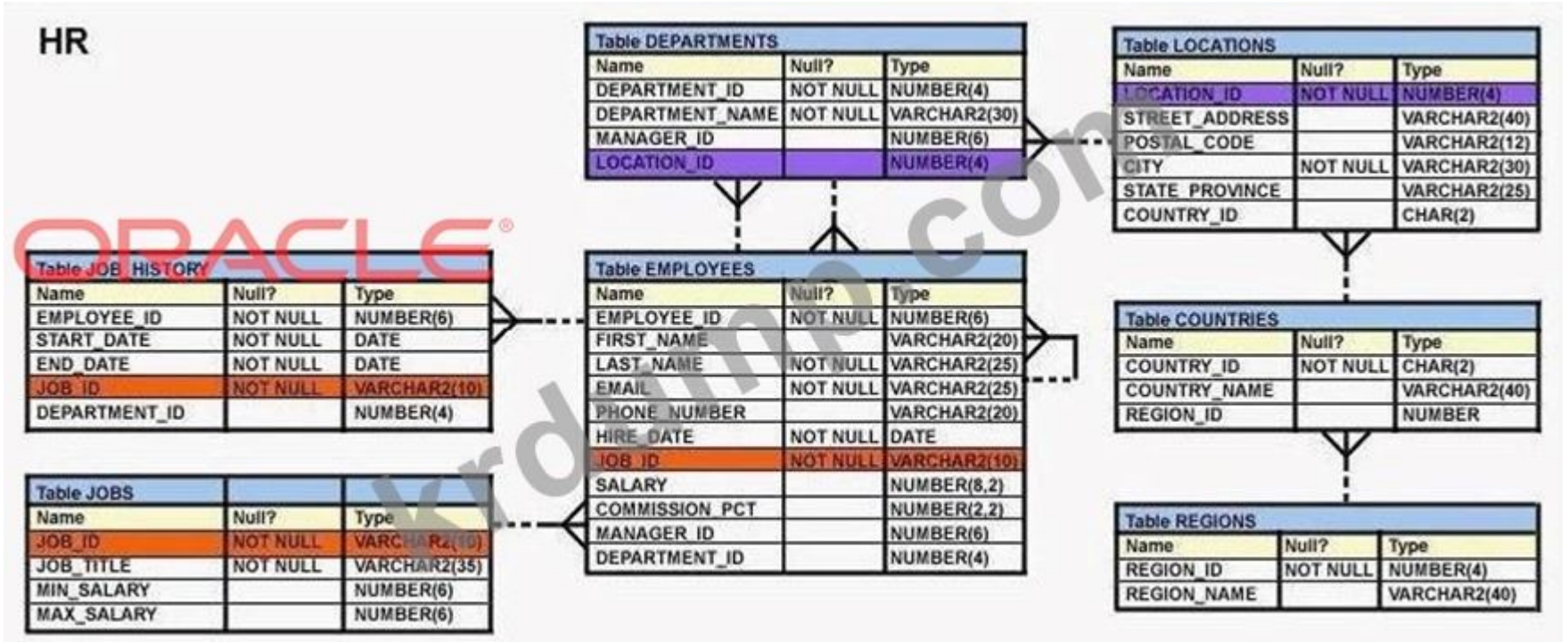
NEW QUESTION: 23
□□ □ □□□ □□ □ □□ □□ □ □□ □□?
A. □□ □□□ □□□ □ □□□□.
B. □□□ □□ □□□ □□□ □□□ □□□ □□ □ □□□□.
C. □□□ □□□ □□□□ □ □□□□.
D. □□□□ □□ □□ □□ □□□□□.
E. □□□ □ □□, □□, □□□ □□ □□□□ □ □□□□.
F. select □□ where □□□□ □□□ □ □□□□.
Answer: A,B,E (LEAVE A REPLY)

NEW QUESTION: 24
ORDER BY □□ □□ □□□ □□ □□□□ □□ □□?

A. □□ □□□□ □□ □□□□□ □□□□□.
B. NULL□ □□ □□□ □□□□ □□□□.
C. ORDER BY □□□□ SELECT □□□ □□□ □□ □□□ □ □□□□.
D. □□ □□ □□□□ □□ □□ □□□□□□ □□□□□.
E. ORDER BY □□□ □ □□□ □□□ □ □□□□.
Answer: A (LEAVE A REPLY)
□□
ORDER BY □□ □□ □□□ □□□□ □□ □□□ □□□□□.

NEW QUESTION: 25
□□ □ □□□ □□ □□ □□□□ □□ □□ □□□□ □□ □□ □□□□□?
A. □2□□□
B. □ □□ □□□
C. □3□□□
D. □4□□□
Answer: (SHOW ANSWER)
□□
□□:
https://blog.udemy.com/database-normal-forms/

NEW QUESTION: 26
□□□□ □□ EMPLOYEES □ JOB_HISTORY □□□□ □□□ □□□□□□.



□□ □ □□□□ SA_MAN□□□ □□□ □□ □□ □□□ □□ ID□ □□□□ □□ □ □□□ □□□□□□.
EMPLOYEE_ID □□
□□□□□□
WHERE JOB_ID = 'SA_MAN'

EMPLOYEE_ID □□
FROM JOB_HISTORY
WHERE JOB_ID = 'SA_MAN';
□□□ □□□ □□□ □□□□□ □□ □ □□ □□□ SET □□□□ □□□□□□.

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

NEW QUESTION: 27
□□□□ □□ 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)

□□ □□□□□ □□ □□ □□ □□□ □□□□ □□ □ □□ □□□ □□□□□?

A. '□□'□ '□□□'□ □□□□ □□ □□□ □□ □□ □□ □□□
B. □□□□□ '□□'□ □□□□ □□□ □□□□□ □□□ □□□ □□□□□.
C. □□□□□ □□ 1980□ □□ □□□ □□
D. □ □□□□ □□ □□□ '□□'□ □□ □□ □□□□.
E. □ □□□□ □□ □□□ □□ □□ □□□□ □□ □□□ □□ □□ □□ □□□□.
Answer: B,E (LEAVE A REPLY)

NEW QUESTION: 28
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 Course_id, Department_id, FROM Department_details d RIGHT OUTER JOIN Course_details c USING (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 c.course_id, d.department_id FROM Course_details c FULL OUTER JOIN Department_details d ON (c.department_id=d. Department_id)

Answer: (SHOW ANSWER)

NEW QUESTION: 29

INVOICE □□□□ □□□ □□□□ □□ □□□□ □□□□□.

```
INVOICE

Name          Null?     Type
------------- --------- --------------
INV NO        NOT NULL  NUMBER
INV_DATE                DATE
CUST_NAME     NOT NULL  VARCHAR2 (20)
CUST_CAT                CHAR(1)
INV_AMT                 NUMBER(8,2)


INV_NO  INV_DATE    CUST_NAME   CUST_CAT  INV_AMT
------- ----------- ----------- --------- ----------
101     15-FEB-08   JAMES       1         255982.55
102     18-MAR-08   SMITH       2         100000.00
```


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

A. inv_no BETWEEN '101' AND '110': □□□ □□ □□

B. inv_date > '01-02-2008': □□□ □□ □□

C. CONCAT(inv_amt, inv_date): □□□ □□ □□

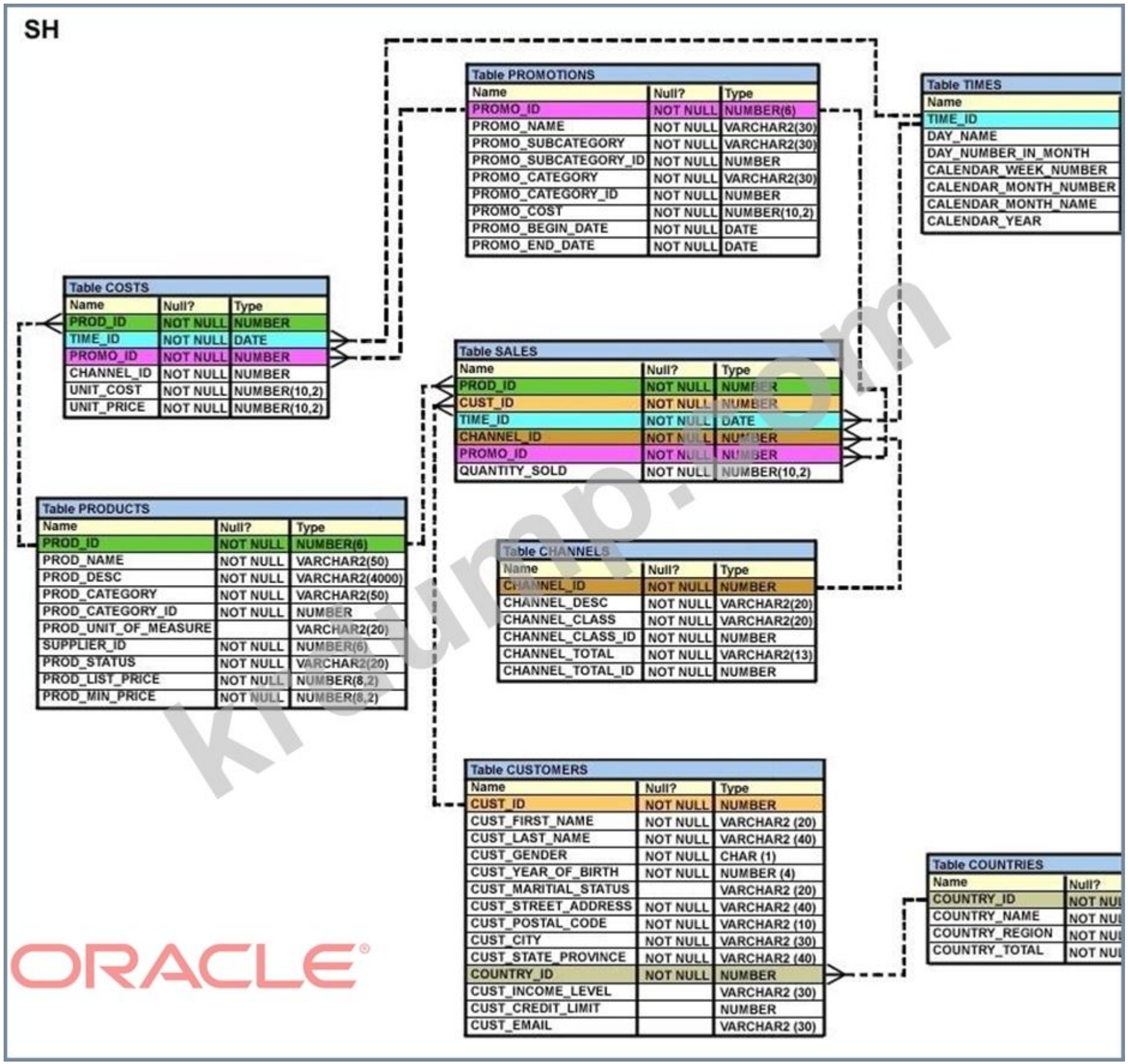
D. inv_amt = '0255982': □□□ □□□ □□□□□.
E. inv_date = '2008□ 2□ 15□': □□□ □□ □□
Answer: (SHOW ANSWER)

NEW QUESTION: 30
PROGRAMStable□ □□□ □□□□□□.

Name	Null?	Type
PROG_ID	NOT NULL	NUMBER (3)
PROG_COST		NUMBER (8, 2)
START_DATE	NOT NULL	DATE
END_DATE		DATE

□□ □ □□ SQL □□ □□□□□ □□□□□?
□□ NVL(ADD_MONTHS(END_DATE,1)SYSDATE)
A. □□□□□□;
SELECT NVL(TO_CHAR(MONTHS_BETWEEN(start-date, end_date)),'□□ □')
B. □□□□□□;
SELECT NVL(MONTHS_BETWEEN(start_date,end_date),'□□ □')
C. □□□□□□;
D. □□□□□□;
SELECT TO_DATE(NVL(SYSDATE-END_DATE,SYSDATE))
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 31
□□□□ □□ 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. CREATETABLE □□□ □ □□□ SELECT □□ □□□□ □□ □□□ NEW_SALEStable□ □□□□ □□□□.
C. □ □□□ DEFAULT □□ □□□ □ □□ □□□ NEW_SALEStable□ □□□□ □□□□.
D. NEW_SALEStable□ □□□□ □□□ □□ □□□ □□ FOREIGNKEY □□ □□□ □ □□□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 32
CREATE TABLE □□□ □□ □□□□ □□ □□? (3□□ □□□□□.)
A. □□ □ □□ □□□ □□□□ □□ □□□□ □□□□ □□ CREATE..INDEX □□ □□□ □ □□□□.
B. □□□□□ □□□ □□□□□.
C. □□□□ □□□□ □□□□□ CREATE ANY TABLE □□□ □□□ □□□.
D. □□ □□ □□□□□ □□□□□ □□□□□.
E. □□□ □□□□ □□□□ □□□ □□□□□□□□□ □□ □□□ □□ □□□□ □□□ □□□.
F. □□□ □□□□ UNLIMITED TABLESPACE □□□ □□□ □□□ □□□.
Answer: (SHOW ANSWER)
□□/□□: https://docs.oracle.com/html/E25494_01/tables003.htm

NEW QUESTION: 33
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□□ □□ □□ □□□ □□ □□□ □□□□ □□□ □□□□□.
B. RM□ TRANSACTION_TYPE□□ □□ □□ □□ □□□ □□□ A101 □□ A102 □□□ □□□ □□□□□.
C. RM□ TRANSACTION-TYPE□□ □□ □□ □□□ □□ □□ A101, A102□ □□ □□□ □□□□□.

D. RM□ TRANSACTION_TYPE□□, MEMBER_ID□ A101, A102□ □□ □□ □□□ □□ □□□ □□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 34
□□□ □□ □□□□ □□ □□?
A. SAVEPOINT□ □□□ □□□□ □□□□ DML(□□□ □□ □□) □ □□□ □□ □□□□□ □□□□□.
B. □□□□ □ □□□ □□ □□(DDL) □□ □□ □□□□□ □□□□□.
C. COMMIT□ □□□ □□□□ □□□□ DDL □ □□□ □□ □□□□□ □□□□□.
D. COMMIT□ □□□ □□□□ □□□□ DDL □ DML □□ □□□ □□ □□□□□ □□□□□.
Answer: B (LEAVE A REPLY)
□□
□□:
https://docs.oracle.com/database/121/CNCPT/transact.htm#CNCPT038

NEW QUESTION: 35
□□□□ □□ ORDERS □□□□ □□□ □□□□□□. ORDER_MODE □ ORDER_TOTAL □□ □□□□ □□ 'direct' □ 0□□□.

ORDERS ORACLE

Name	Null?	Type
ORDER_ID	NOT NULL	NUMBER(12)
ORDER_DATE	NOT NULL	TIMESTAMP(6)
ORDER_MODE		VARCHAR2(8)
CUSTOMER_ID	NOT NULL	NUMBER(6)
ORDER_TOTAL		NUMBER(8,2)

□□ 2□□ INSERT□□ □□□□□? (2□□ □□□□□.)
A. INSERT INTO □□(order_id,order_date,order_mode,order_total)VALUES(1,'10-mar-2007','□□□',1000);
B. INSERT INTO □□(order_id,order_date,order_mode,(customer_id,order_total)VALUES(1,TO_DATE(NULL), '□□□', 101, NULL);
C. INSERT INTO(□□ ID, □□ □□, □□ ID FROM □□)VALUES(1,'09-mar-2007', 101);
D. INSERT INTO ordersVALUES (1,'09-mar-2007', 'online','', 1000);
E. INSERT INTO orderVALUES (1,'09-mar-2007', DEFAULT, 101, DEFAULT);
Answer: (SHOW ANSWER)

NEW QUESTION: 36
INVOICE □□□□ □□□ □□□□ □□ □□□□ □□□□□. (2□□ □□□□□.)

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

□□ □□□□ □□□ □□ □□□ □□ □□□□ □□ □□ □□?

A. inv_date > '01-02-2008' : □□□ □□ □□

B. inv_date = '2008□ 2□ 15□' : □□□ □□ □□

C. inv_amt = '0255982' : □□□ □□ □□

D. inv_no BETWEEN '101' AND '110' : □□□ □□ □□

E. CONCAT(inv_amt, inv_date) : □□□ □□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 37

Oracle □□ □ ANSI □□ □□□ □□ □ □□ □□ □ □□ □□?

A. Oracle □□ □□□ SQL:1999 □□ ANSI □□ □□□□ □□□ □□□□.

B. Oracle □□ □□□ □□ □□□ □□□□□.

C. SQL:1999 □□ ANSI □□ □□□ □ □□□□ □□□□ □ □□□ □□□□□.

D. Oracle □□ □□□ SQL:1999 □□ ANSI □□ □□□□ □□□ □□□□□.

E. Oracle □□ □□□ □ □□□□ □□□□ □ □□□ □□□□□.

F. Oracle □□ □□□ □□□ □□ □□□ □□□□□.

G. SQL:1999 □□ ANSI □□ □□□ □□ □□□ □□□□□.

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

NEW QUESTION: 38

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                                NOT NULL VARCHAR2 (3)
```

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

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

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

C. SELECT □, □□ FROM □□ WHERE state LIKE 'M%';

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

Answer: B (LEAVE A REPLY)

NEW QUESTION: 39

□□ □□□ □□□□□.

```
SQL> DEFINE hiredate = '01-APR-2011'
SQL> SELECT employee_id, first_name,salary
     FROM employees
     WHERE hire_date > '&hiredate'
     AND manager_id > &mgr_id;
```

□□□ □□ □□ □□ □□□ □□ □□□□□ □□□□□?

A. □□□ □□ □□ □□□ □□

B. □□ □□ 'hiredate' □ 'mgr_id\

C. 'mgr_id'□

D. '□□'□

Answer: B (LEAVE A REPLY)

NEW QUESTION: 40

□□□ □□□ □□□ □□□ □□□ □□ □□□ □□□□□.

1. □□ □□□ WHEREclause□ □□□□□.
2. □□ □□□ □□□ □□□□□ □□ □□ □□□□□.
3. □□□ □□ □□ □□□ □□□ □□□□ □□ □□ □□ □□□□□.
4. □□ □□ □□□ □□ □ □□□ □□□ □ □□ □□□ □□ □□□□□.

Oracle □□□ □□□ □□ □□□ □□□□ □□□ □□□ □□□□□?

A. 2, 1, 4, 3

B. 4, 1, 2, 3

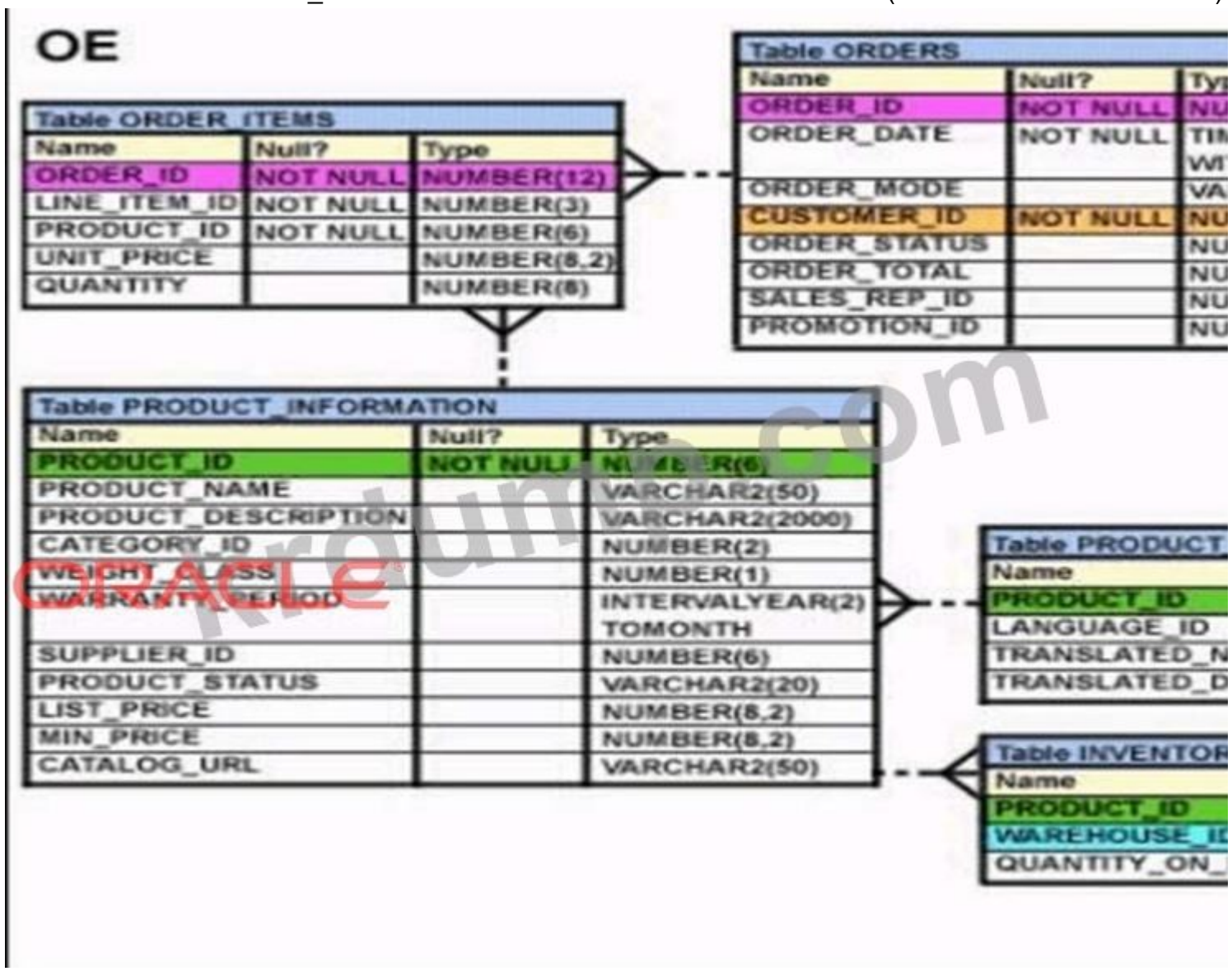
C. 4, 2, 1, 3
D. 2, 4, 1, 3
Answer: D (LEAVE A REPLY)
□□/□□:
□□:
http://rajanimohanty.blogspot.co.uk/2014/01/correlated-subquery.html

NEW QUESTION: 41
□□□□ □□ ORDERS_MASTER □ MONTHLY_ORDERS □□□□ □□□□ □□□□□.
ORDERS_MASTER
□□ □□□
ORDER_TOTAL
1
1000
2
2000□
□
3000
4
MONTHLY_ORDERS
□□ □□□
ORDER_TOTAL
2
2500
□
□□ MERGE □□ □□□□□.
MERGE_INTO □□_□□□ o
USINGmonth_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: 42
DESCRIBE □□□ □□ □ □□ □□ □ □□ □□?
A. SQL*Plus□□□ □□□ □ □□□□.
B. □ □□ □□□ □□ □□ □□□ □□□□□.
C. □□ □□ □□□ □□ □ □□ □□ □□ PRIMARY KEY □□ □□□ □□□□□.
D. □□ □□ □□□ □□ □□ □□ □□ NOT NULL □□ □□□ □□□□□.
E. SQL Developer□□ □□□ □ □□□□.
F. □□ □□□ □□□ □□□□ □ □□□ □ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 43
□□□□ □□ PRODUCT_INFORMATION □□□□ □□□ □□□□□□. (□□ □□ □□ □□□□□.)



PRODUCT_ID □□ □□ □□□□.
□□ □□□ □□□□ □□□□ □□□□□.

SQL > □□□ □□ upper_name_idx
ON product_information(UPPER(product_name));
PRODUCT_INFORMATION □□□□ □□ □□□□ □□□□.
□□ □□□ UPPER_NAME_IDX □□□□ □□□□□?
A. SELECT UPPER(□□□)
□□ □□□□;
B. □□ ID □□
FROM □□ □□
WHERE UPPER(□□□) IN ('LASERPRO', 'CABLE');
C. □□ ID □□, UPPER(□□ □□)
FROM □□ □□
WHERE UPPER(product_name) = 'LASERPRO' □□ list_price > 1000;
D. SELECT UPPER(□□□)
FROM □□ □□
□□ □□ ID = 2254;
Answer: B (LEAVE A REPLY)

NEW QUESTION: 44
□□ □□□□ □□□□□□.

```
SELECT 1 AS id, 'John' AS first_name
  FROM dual
 UNION
SELECT 1, 'John' AS name
  FROM dual
 ORDER BY 1;
```

□□ □ □□□ □□□□□?
A. 2□
B. 0□
C. 1□
D. □□
Answer: A (LEAVE A REPLY)

NEW QUESTION: 45
□□ □□□□ Oracle □□□□□□□ □□□ □□□□ □□ □□□□ □□ □□?
A. □□□□ □□ MAXVALUE □□□ □□□□ ALTER SEQUENCE □□ □□□□ □□ □ □□□□.
B. □□□□□ □□□ □□□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□□ □ □□□□.
C. CURRVAL□ □□ □□□□ □□ □□□ □□ □□ □□□ □□□ □□□□ □ □□□□□.
D. □□□□□□ □□□□□ □□□□□□ □□□□ □□ □□□□□□ □□□□ □□ □□□ □□□ □□□□□ □□□□ □ □□ □□□ □ □□□□.
E. DELETE <sequencename>□ □□□□□□□□ □□□□ □□□□□.
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 46
BOOKS_TRANSACTIONStable□ □□□□□□□ □□□□.
SQL>SELECT * FROM books_transactions ORDER BY 3;
□□□ □□□ □□□□□?
A. □□ □ □□ □□□□ □□□□ □□□ □□□□ □□ □□□□□.
B. □ □□□□ □□ □□ □□ □□ 3□□ □□ □□□□ □□□□ □□□ □□□□ □□ □□□□.
C. ORDER BY□□ □□ 3□ □ □□□□ □□□ □□□ □□□ □□□□□.
D. □□ □□□□ □ □□ □□ □□ □□ □□□□□□ □□□□ □□□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 47
CUSTOMERS □□□□ □□□ □□□□□□. (2□□ □□□□□□.)

NAME	NULL?	TYPE
CUSTNO	NOT NULL	NUMBER(3)
CUSTNAME	NOT NULL	VARCHAR2(25)
CUSTADDRESS		VARCHAR2(35)
CUST_CREDIT_LIMIT		NUMBER(5)

CUSTNO□ □□ □□□□.
□□ □□ □□□ □□□□ □□ CUSTNO□ □□□□ □□□ □□ □□□ □ □ □□ □□□□□ □□□□ □□□.
□□□ □□□ □□ □□ □□ □ □□ □□□ □□□ □ □□□□?
A. □□ □□□ □□ □□ □□ □□
B. □□ □□
C. □□ □□□ □□ □□□ □□ □□
D. □□ □□□ □□ □□ □□ □□
E. □□□□
Answer: (SHOW ANSWER)

NEW QUESTION: 48
EMPLOYEES □□□□ □□ □□□ □□□□□.

```
Name                 Null?    Type
-------------------- -------- ----------------------------
EMP_ID               NOT NULL NUMBER
EMP_NAME                      VARCHAR2(10)
DEPT_ID                       NUMBER(2)
SALARY                        NUMBER(8,2)
JOIN_DATE                     DATE

NLS_DATE_FORMAT is set to DD-MON-YY.
```

□□□ □□□ □□ □□□ □□□ □□□ □□□□□?

A. □□ □□ □□ || ' ' || □□ □□;

B. SELECT SUBSTR(join_date, 1, 2) - 10 FROM □□;

C. SELECT □□ + '120.50' FROM □□;

D. SELECT join_date + '20' FROM □□;

E. SELECT join_date FROM □□ WHERE join_date > '10-02-2018';

Answer: (SHOW ANSWER)

NEW QUESTION: 49

□□ □□□ □□□□□.

```
SQL> ALTER TABLE books_transactions
     ADD CONSTRAINT fk_book_id FOREIGN KEY (book_id)
     REFERENCES books (book_id) ON DELETE CASCADE;
```

ON DELETE CASCADE□ □□□ □□□□□?

A. BOOKStable□ □□□□ BOOK_TRANSACTIONStable□ □□□□□.

B. BOOKStable□ □□□□ BOOK_TRANSACTIONStable□ □□ □□ □□□□□ □□□ □□□ □□□□□.

C. BOOKStable□ □□ □□□□ BOOK_ID□ BOOKStable□ □□□ □□ □□□□ BOOK_TRANSACTIONStable□ □□ □□□□□.

D. BOOKS.BOOK_ID □□□ □□ □□□□ BOOKS_TRANSACTIONS.BOOK_ID □□□ □□ □□ □□□□□□.

Answer: (SHOW ANSWER)

□□

NEW QUESTION: 50

□□ SQL □□ □□□□□.

```
SELECT cust_id, cust_last_name "Last Name"
  FROM customers
 WHERE country_id = 10
  UNION
SELECT cust_id CUST_NO, cust_last_name
  FROM customers
 WHERE country_id = 30
```

□□□ □□□□□ □□□ □ □□ □ □□ order by □□ □□□□□.

A. CUST_NO □□

B. "□"□□ □□

C. ORDER BY 2, cust_id
D. 2, 1□ □□
E. "CUST NO"□ □□
Answer: B,C,D (LEAVE A REPLY)

NEW QUESTION: 51
□□ □□□ □□□□□.
SQL> SELECT TRUNC(ROUND(156.00, -1),-1)
□□□□;
□□□ □□□ □□□?
A. 150
B. 200
C. 160
D. 16
E. 100
Answer: (SHOW ANSWER)
□□:
https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions135.htm
https://docs.oracle.com/cd/B28359_01/olap.111/b28126/dml_functions_2127.htm

NEW QUESTION: 52
□□ □□ □□ □□□□ □□ □□?
A. □□ □□ □□ □ □□□ □□□ □ □□□□.
B. □□□ □□□ □□□□□ □□□ □ □□□□.
C. update □□ debete □□ where □□□ □□□ □ □□□□.
D. □□□□□ □□ □□□□ □□□□ □□ set □□□ □□□ □ □□□□.
E. □□□ □□□ □ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 53
EMPLOYEES_ID □ LOGIN_ID □□ □□ null□ □□□ □□□□ □□ □□□ EMPLOYEES□ □□□□ □□□.
□□ □ □□□ □□□□ □□□□ □ □□ SQL □□ □□□□□? (2□□ □□□□□.)
A. CREATE TABLE □□
(employee_id NUMBER CONSTRAINT emp_id_nn NOT NULL,
Login_id NUMBER CONSTRAINT login_id_nn NOT NULL,
□□□ VARCHAR2(100),
□□ □□ DATE,
CONSTRAINT emp_id_ukUNIQUE (employee_id, login_id));
B. CREATE TABLE □□
(□□ ID NUMBER,
□□□ ID NUMBER,

□□□ VARCHAR2(100),
□□ □□ DATE,
CONSTRAINT emp_id_ukUNIQUE (employee_id, login_id));
C. CREATE TABLE □□
(employee_id NUMBER CONSTRAINT emp_id_pk □□ □,
□□□ ID NUMBER UNIQUE,
□□ □□ VARCHAR2(25),
□□ □□ □□);
D. CREATE TABLE □□
(□□ ID NUMBER,
□□□ ID NUMBER,
□□ □□ VARCHAR2(25),
□□ □□ DATE,
CONSTRAINT emp_id_pk □□ □(employee_id, login_id));
E. CREATE TABLE □□
(□□ ID NUMBER,
□□□ ID NUMBER,
□□□ VARCHAR2(100),
□□ □□ DATE,
CONSTRAINT emp_id_uk UNIQUE(employee_id, login_id);
CONSTRAINT emp_id_nn NOT NULL (employee_id, login_id));
Answer: (SHOW ANSWER)

NEW QUESTION: 54
STORES □□□□□ □□ □□□ □□□ □□□□ □□□ □□ DATE□ START_ DATE □□ □□□□.
START_ DATE□ □□ 25□□ □□□ □□ □□□□□ □□□□□ □□□.
□□ WHERE □□ □□□ □ □□□□?
A. WHERE TO_ NUMBER(□□_□□ - SYSDATE) <= 25
B. WHERE MONTHS_ BETWEEN (SYSDATE, □□□) <= 25
C. WHERE MONTHS_ BETWEEN(□□_ □□, SYSDATE) <= 25
D. WHERE ADD_ MONTHS(□□_□□, 25) <= SYSDATE
Answer: (SHOW ANSWER)

NEW QUESTION: 55
□□ □□□ □□□□□? (□□□□ □□ □□ □□□□□□.)
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: 56
STUDENT □ FACULTY □□□□ □□□ □□ □□□□ □□□□.

```
STUDENT
Name                      Null?                    Type
------------------------  -----------------------  -------------
STUDENT_ID                NOT NULL                 NUMBER(2)
STUDENT_NAME                                       VARCHAR2(20)
FACULTY_ID                                         VARCHAR2(2)
LOCATION_ID                                        NUMBER(2)

FACULTY
Name                      Null?                    Type
------------------------  -----------------------  -------------
FACULTY_ID                NOT NULL                 NUMBER(2)
FACULTY_NAME                                       VARCHAR2(20)
LOCATION_ID                                        NUMBER(2)
```

□□ □□□□ □□□□ □□□□ □□ □ □□□ □□□□ □□□□ □□□.
□□ □ SQL □□ □□□□□.
□□ 1
SQL>SELECT □□_□□, COUNT(□□_ID)
FROM □□ JOIN □□□
USING (faculty_id, location_id)
GROUP BY □□□ □□;
□□ 2
SQL>SELECT □□_□□, COUNT(□□_ID)
FROM □□ NATURAL JOIN □□□
GROUP BY □□□ □□;
□□□ □□ □□□□ □□ □□?
A. □□□ 1□ □□□□□ □□□□ □□□ □□□ □□□□□.
B. □□□ 1□ 2□ □□ □□□□□ □□□□ □□ □□□ □□□□□.
C. □□□ 2□ □□□□□ □□□□ □□□ □□□ □□□□□.
D. □□□ 1□ 2□ □□ □□□□□ □□□□ □□□ □□ □□□ □□□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 57
PROGRAMS □□□□ □□□ □□□□□□.

Name	Null?	Type
PROG_ID	NOT NULL	NUMBER (3)
PROG_COST		NUMBER (8, 2)
START_DATE	NOT NULL	DATE
END_DATE		DATE

□□ □ □□ SQL □□ □□□□□ □□□□□? (2□□ □□□□□.)
A. SELECT TO_DATE(NVL(SYSDATE-END_DATE,SYSDATE))FROM □□□□;
B. SELECT NVL(MONTHS_BETWEEN(start_date,end_date),'□□ □')FROM □□□□;
C. SELECT NVL(TO_CHAR(MONTHS_BETWEEN(start-date,end_date)),'□□ □')FROM □□□□;
D. SELECT NVL(ADD_MONTHS(END_DATE,1)SYSDATE)FROM □□□□;
Answer: (SHOW ANSWER)

NEW QUESTION: 58
□□ □□□□□ 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. □□□□ □
B. □□ □□□
C. □□□□ PA
D. □□□□ IS
Answer: A (LEAVE A REPLY)

NEW QUESTION: 59
□□□ □□ □ □□ □□ □ □□ □□?
A. ALTER ROLE □□ □□□□ □□□ □□□ □□□□□.
B. □□□ □□□ □□ □□□□□ □□□ □ □□□□.
C. □□□ □□□□□□ □□□ □ □□ □□ □□□ □□□ □□□□□.
D. ALTER ROLE □□ □□□□ □□□ □□□ □□□□□.
E. □□□ GRANT □□ □□□□ □□□ □□□□□.
F. □□ □□□□□ □□ □□□ □□□ □ □□□□.

G. ALTER USER □□ □□□□ □□□□□ □□□ □□□□□.
Answer: B,E,F (LEAVE A REPLY)

NEW QUESTION: 60
□□□□ □□ ORDERS □□□□ □□□ □□□□□□.
ORDER_MODE □ ORDER TOTAL □□ □□ □□□ 'direct'□ □□□□.
□□ 2□□ INSERT□□ □□□□□? (2□□ □□□□□.)
A. INSERT INTO □□ VALUES(1, '09-mar-2007', '□□□',' ',1000);
B. INSERT INTO □□ VALUES('09-mar-2007',DEFAULT,101, DEFALT);
C. INSERT INTO(□□ ID □□, □□ □□, □□ ID FROM □□) VALUES(1, '09-mar-2007",101);
D. INSERT INTO □□(order_id, order_date, □□ □□, customer_id, order_total) VALUES (1, TO_DATE (NULL),'online',101, NULL) ;
E. INSERT INTO □□(□□ ID, order_date, □□ □□, order_total)VALUES (1,'10-mar-2007','online', 1000)
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 61
□□□ □□ □□□ □□□ □□ □□□□□□.

1 One-to-one	a) teacher to Student
2 One-to-many	b) Employees to Manager
3 Many-to-one	c) Person to SSN
4 Many-to-many	d) Customers to Products

□□ □ □□□□ □□□□ □□□ □□□□ □□□ □□□□□? (□□ □□ □□ □□□□□.)
A. 1-a, 2-b, 3-c □ 4-d
B. 1-d, 2-b, 3-a □ 4-c
C. 1-c, 2-d, 3-a □ 4-b
D. 1-c, 2-a, 3-b □ 4-d
Answer: A (LEAVE A REPLY)

NEW QUESTION: 62
□□□□ □□□□ PROMOTIONS □□□□ □□□ □□□□□□.
PROMOTIONS □□□□ □□□□,
□□ A□□ $0-2000 □ $2000-5000 □□□ □□ □□□□□ □□ □□ □□□ □□□ □□□.
□□ SQL □□ □□□□□.
□□□ □□□ □□□?
A. WHEN □□ □□ □□□ □□□ □ □□□□ □□□ □□□□□.
B. □□□□□ □□□□ □□□ □□□ □□□□□.

C. CASE□ □□ □□□ □□ □□□ □ □□□□ □□□ □□□□□.
D. □□ □□□ NULL□ □□□ □ □□ □□□ □□□ □□□□□.
Answer: (SHOW ANSWER)
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: 63
□□□ □□ □□□□□? (□□□□ □□ □□ □□□□□□.)
A. PL/SQL □□ □□□ □□□ □
B. DELETE □ □□ □
C. □□□ □□ □□ □□□ □□□ □
D. □□ □□ □□□□ □□□ TRUNCATE □□ □□□ □
E. ROLLBACK □□□ □□□ □
Answer: C,D,E (LEAVE A REPLY)
□□/□□:

NEW QUESTION: 64
□□ □□□ □□□□ □□□□□.
SQL> DROP TABLE □□;
□ □□ □□ □ □□ □□?
A. □□□ □□□ □ □□□ □□□ □□□□□.
B. □□□□ □□□□ □□ □□ □□ □ □□□□ □□□□□□ □□□□□.
C. □□□ □□□□ □□□□□ □□□ □□□ □□□□□.
D. □□□□ □□□□ □□ □□ □□□□□ □□□□□.
E. □□□□ □□ □□□(□□□□ □□)□ □□□□□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 65
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: A (LEAVE A REPLY)

NEW QUESTION: 66
'e' □□□ □□□□ □□□ □ □□ □□□ 'a' □□□ □□ CUSTOMERS □□□□ □□ □□ □□□ □□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?
cust_first_name □□
A. □□□□□□
WHERE INSTR(cust_first_name, 'e')<>'' AND
SUBSTR(cust_first_name, -2, 1)='a';
cust_first_name □□
B. □□□□□□
WHERE INSTR(cust_first_name, 'e')<>0 AND
SUBSTR(cust_first_name, -2, 1)='a';
cust_first_name □□
C. □□□□□□
WHERE INSTR(cust_first_name, 'e')IS NOT NULL AND
SUBSTR(cust_first_name, 1, -2)='□';
cust_first_name □□
D. □□□□□□
WHERE INSTR(cust_first_name, 'e')<>0 AND
SUBSTR(cust_first_name, LENGTH(cust_first_name), -2)='a';
Answer: (SHOW ANSWER)

NEW QUESTION: 67

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

SALES1 table

Name	Null	Type
SALES_ID		NUMBER
STORE_ID		NUMBER
ITEMS_ID		NUMBER
QUANTITY		NUMBER
SALES_DATE		DATE

SALES2 table

Name	Null	Type
SALES_ID		NUMBER
STORE_ID		NUMBER
ITEMS_ID		NUMBER
QUANTITY		NUMBER
SALES_DATE		DATE

□□ □□ □□□□ □□□ □□□ □□□□□?

A. □□

B. □□□

C. □□□

D. □□□□

E. □□

Answer: (SHOW ANSWER)

□□/□□:

□□:

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

NEW QUESTION: 68

EMPLOYEEStable□ □□□ □□□□□□.

ame	Null?	Type
MPLOYEE_ID	NOT NULL	NUMBER(6)
IRST_NAME		VARCHAR2(20)
AST_NAME	NOT NULL	VARCHAR2(25)
MAIL	NOT NULL	VARCHAR2(25)
HONE NUMBER		VARCHAR2(20)
IRE_DATE	NOT NULL	DATE
OB_ID	NOT NULL	VARCHAR2(10)
ALARY		NUMBER(8,2)
OMMISSION_PCT		NUMBER(2,2)
ANAGER_ID		NUMBER(6)
EPARTMENT_ID		NUMBER(4)

EMPLOYEE_ID□ MANAGER_ID □□□□ □□/□□ □□□ □□□□.
EMPLOYEE_ID□ 123□ □□□ □□□ □□□□□ □□□ □□□ □ □ □□□ ID□ □□□□□ □□□.
□□ □□□ □□□ □□□ □□□□□?

A. e.last_name, m.manager_id □□
FROM □□ e LEFT OUTER JOIN □□ m
on (e.employee_id = m.manager_id)
□□ e.employee_id = 123;

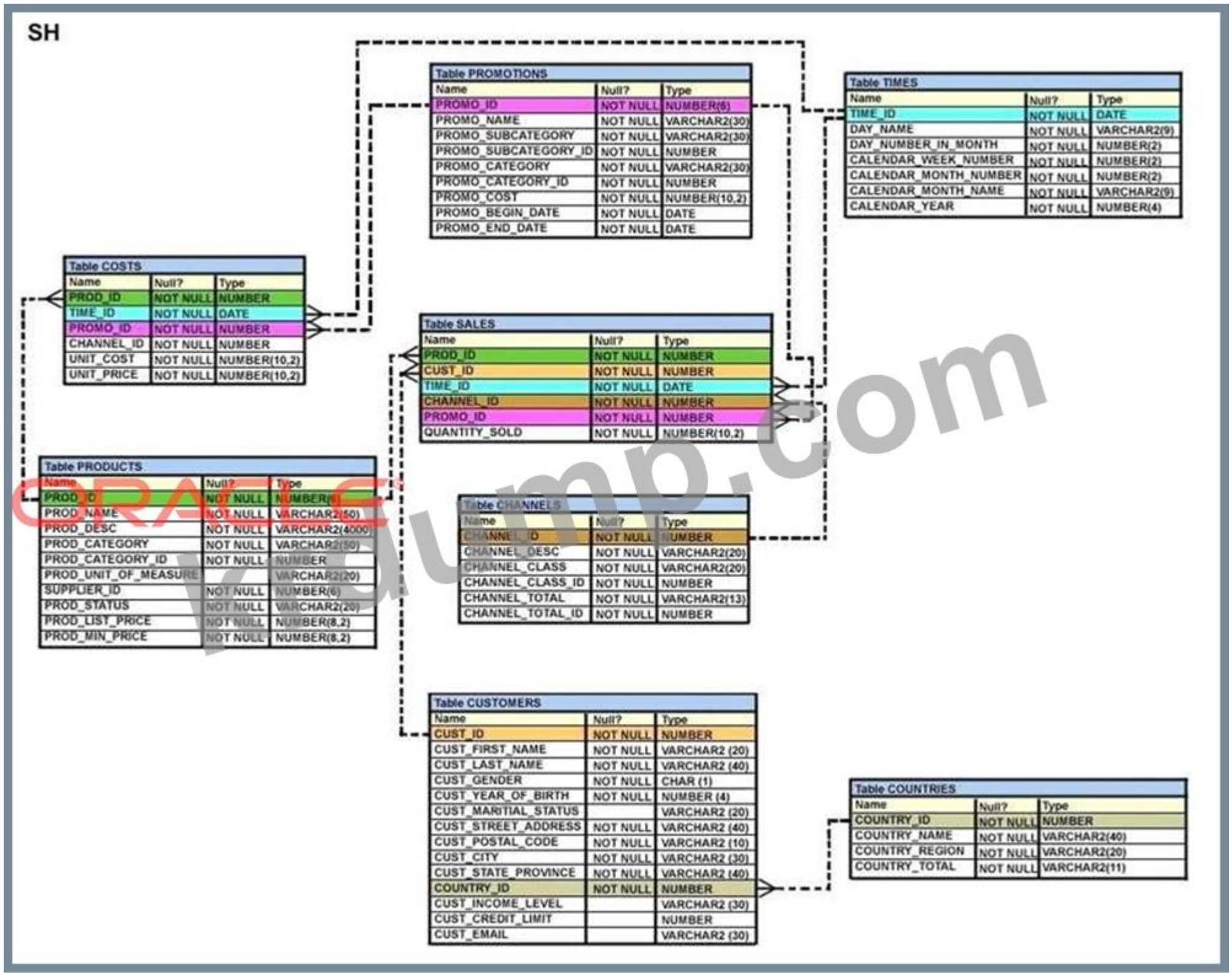
B. m.last_name, e.manager_id □□
FROM □□ e LEFT OUTER JOIN □□ m
on (e.manager_id = m.manager_id)
□□ e.employee_id = 123;

C. e.last_name, m.manager_id □□
FROM □□ e RIGHT OUTER JOIN □□ m
on (e.manager_id = m.employee_id)
□□□ e.employee_id = 123;

D. e.last_name, e.manager_id □□
FROM □□ e RIGHT OUTER JOIN □□ m
on (e.employee_id = m.employee_id)
□□ e.employee_id = 123;

Answer: (SHOW ANSWER)

NEW QUESTION: 69
□□□□ □□ SALES □ CHANNELS □□□□ □□ □□□ □□□□□□.



□□ SQL □□ □□□□□□.

```
INSERT INTO SALES VALUES (23, 2300, SYSDATE,
                          (SELECT CHANNEL_ID
                          FROM CHANNELS
                          WHERE CHANNEL_DESC='DIRECT SALES'),
                        12, 1, 500);
```

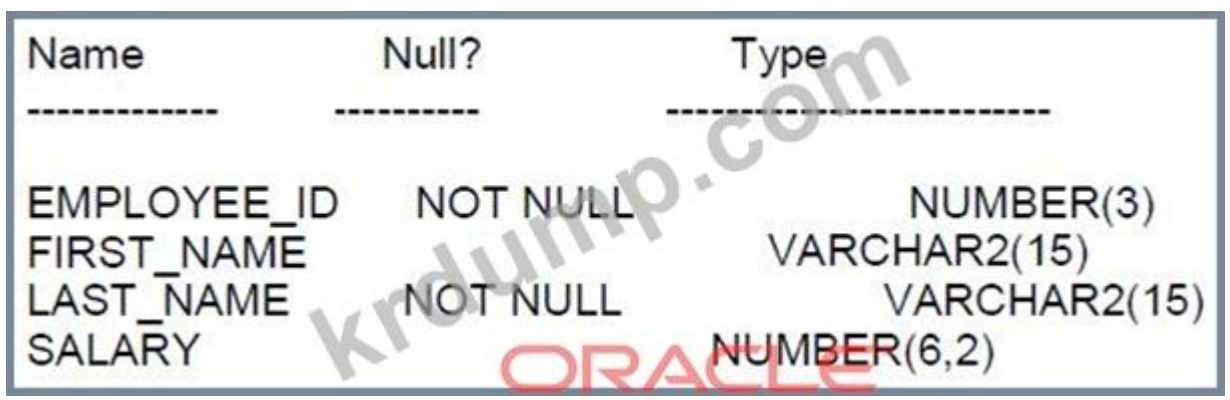
□□□ □□ □□□□ □□ □□? (□□ □□ □□ □□□□□.)

A. □□□□ □□□□ SALES □□□□ □ □□ □□□□□.

B. VALUES □□□ □□ □□□ □□□ □ □□□□ □□□□ □□□□□.

C. □□ □□□ VALUES □□ □□□□ □□ □□□ □□□□ □□□□□.
D. VALUES □□ □□ □□□ □□□□□□ □□□ □□ □□□ □□□□ □□□□□.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 70
EMPLOYEES □□□□ □□ □□□ □□□□□.



□□ □ □□□ □□□□□ □□□□□?
A. SELECT '□□□ '"□□□ ||□□□||'"FROM □□ ;
B. SELECT '□□□ '|| □□|'' □□□□□□;
C. SELECT '□□□ \"□□□. || □□ || □□□□□□ '\";
D. SELECT '□□□ "□□□. || □□ || " □□□□□□ ;
E. SELECT '□□□ '"||□□ ||"" FROM □□ ;
Answer: (SHOW ANSWER)

NEW QUESTION: 71
□□ ALTER TABLE □□ □□□□□.
ALTER TABLE □□ SET UNUSED order_date; □□ □□ □□□□?
A. ALTER TABLE □□□ □□□ □ ORDER_DATE□□ □ □□ ORDERS □□□□ □□□ □ □□□□.
B. ROLLBACK□ □□□□ ORDERS □□□□ ORDER_DATE □□ □□ □□□ □ □□□□.
C. ALTER TABLE □□□ □□□□□ □□□□□ ORDER_DATE □□ □□ □□□ □□□.
D. DESCRIBE □□□ □□□ ORDER_DATE □□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 72
SALES □□□□ □□□ □□□□□□.

NAME	NULL?	TYPE
PRODUCT_ID	NOT NULL	NUMBER(10)
CUSTOMER_ID	NOT NULL	VARCHAR2(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)

□□ □□□□ □□□□□□.

```
SQL > CREATE TABLE sales1 (prod_id, cust_id, quantity_sold, price)
      AS
      SELECT product_id, customer_id, quantity_sold, price
      FROM sales
      WHERE 1 = 2;
```

SALES1 □□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)

A. □ □□ □□□□□.

B. SELECT □□ CREATE TABLE □□ □□ □□ □□□ □□□□ □□ □□□□ □□□.

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

D. □□□□ □□ WHERE □□ □□ □□□□ □□□□.

E. SALES □□□□ □□□ □□ □□□ □□ □□□ □□ □□ PRIMARY KEY □ UNIQUE □□ □□□ □□□□.

Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 73

□□□□ □□ 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 orderVALUES(1, '10-mar-2007', 'direct',(SELECT customer_idFROM customersWHERE cust_last_name='Roberts' ANDcredit_limit=600), 1000);

B. INSERT INTO □□(order_id, order_data, order_mode,(SELECT customer_idFROM customersWHERE cust_last_name='Roberts' ANDcredit_limit=600), order_total)VALUES(1,
'2007□ 3□ 10□', '□□', &customer_id, 1000).

C. INSERT INTO □□(order_id, order_data, order_mode,(SELECT customer_idFROM customersWHERE cust_last_name='Roberts' ANDcredit_limit=600), order_total)VALUES(1,
'2007□ 3□ 10□', '□□', &&customer_id, 1000);

D. INSERT INTO(SELECT o.order_id, o.order_date, o.order_mode, c.customer_id, o.order_totalFROM □□ o, □□ cWHERE o.customer_id = c.customer_idAND c.cust_last_name='Roberts' AND c.credit_limit =600)VALUES(1, '2007□ 3□ 10□', '□□', (SELECT customer_idFROM □□WHERE cust_last_name='□□□' ANDcredit_limit=600), 1000);

Answer: A (LEAVE A REPLY)

NEW QUESTION: 74

SHIPMENTS □□□□ □□□ □□□□□□.

Name	Null?	Type
PO_ID	NOT NULL	NUMBER (3)
PO_DATE	NOT NULL	DATE
SHIPMENT_DATE	NOT NULL	DATE
SHIPMENT_MODE		VARCHAR2 (30)
SHIPMENT_COST		NUMBER (8,2)

SHIPMENT_DATE□ PO_DATE□□□ 1□□ □□□ □□ PO_ID □ □□□ □□□□ □□□□ □□□□ □□□□□ □□□. □□□ □□□ $20□□□.

□□ □ □□□ □□□□□.

```
SQL> SELECT po_id, CASE
WHEN MONTHS BETWEEN (shipment_date,po_date)>1 THEN
TO_CHAR ((shipment_date - po_date) * 20) ELSE 'No Penalty' END PENALTY
FROM shipments;

SQL>SELECT po_id, DECODE
(MONTHS_BETWEEN (po_date, shipment_date)>1,
TO_CHAR ((shipment_date - po_date) * 20) 'No Penalty' PENALTY
FROM shipments;
```

□ □□□ □□ □□□□ □□ □□?

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

B. □ □□ □□□ □□□□□ □□□□ □□□ □□□ □□□□□.

C. □ □ □□□□□ □□□□ □□□ □□□ □□□□□.

D. □ □□ □□□ □□□□□ □□□□ □□□ □□□ □□□□□.

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

Answer: B (LEAVE A REPLY)

NEW QUESTION: 75

□ □□ □□ □ □□ □□? (3□□ □□□□□.)

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: 76

□□ □□□□ □□ □□□□ □□ □□?

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

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

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

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

Answer: (SHOW ANSWER)

□□:

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

NEW QUESTION: 77

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

A. □□□□ MAXVALUElimit□ □□□□ ALTERSEQUENCE□□ □□□□ MAXVALUElimit□ □□ □ □□□□.

B. DELETE <sequencename>□ □□□□□□□□ □□□□ □□□□□.

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

D. CURRVALis□ □□□ □□□ □□□ □□□ □□□□ □ □□□□□.

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

Answer: A,D (LEAVE A REPLY)

□□/□□:

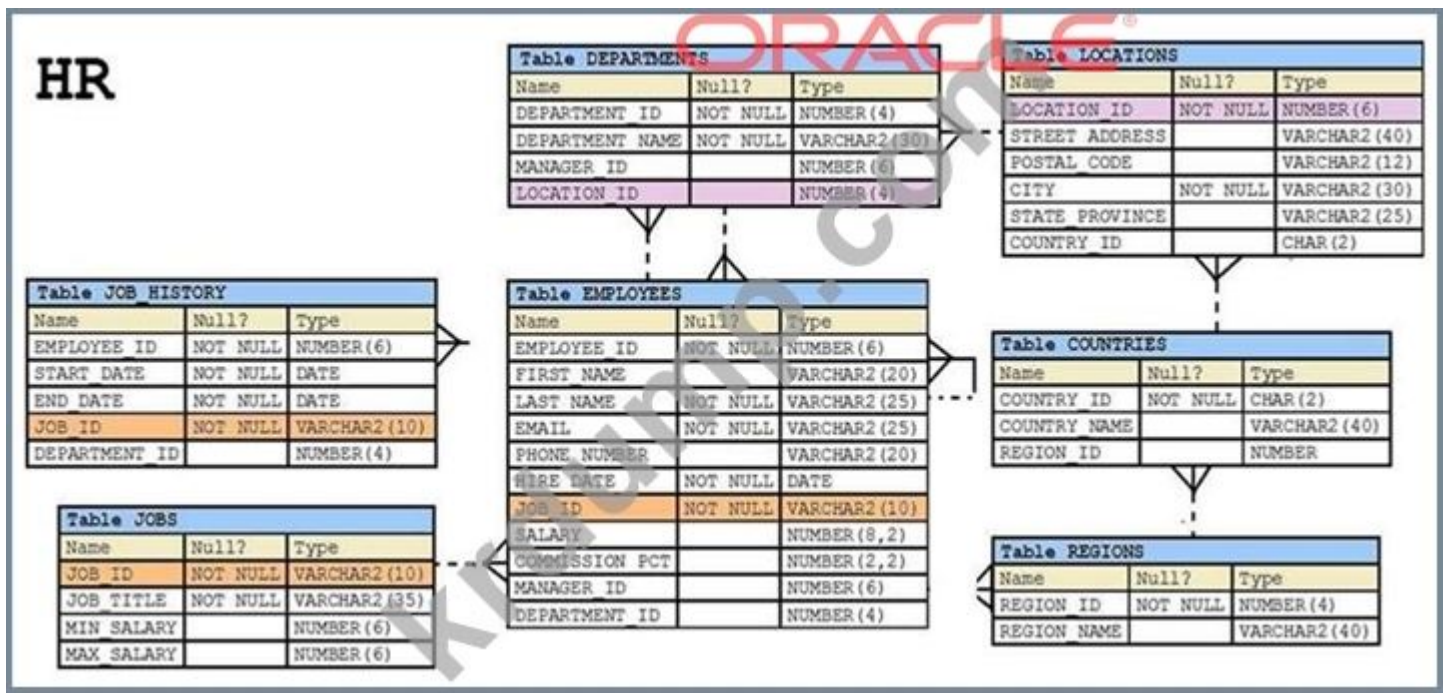
□□:

http://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_2012.htm#SQLRF00817

https://docs.oracle.com/cd/A84870_01/doc/server.816/a76989/ch26.htm

NEW QUESTION: 78

□□□□ □□ EMPLOYEEStable□ □□ □□□ □□□□□□.



□ □□□ □ □□□ □□□□□ □□□. □ □□□ □□□ 1□ □□ □□ □□□ □□□ □□□□□. □□□ □□□□ □□□□ □□□□.

□□□ □□□ □□ □□ □□ SQL □□ □□□□□□□□?

SELECT □□, □□, □□*12+(□□*NVL2(commission_pct,

A. □□□□□□;
SELECT □□, □□(□□ + NVL(commission_pct, 0)*□□)*12 "□□"
B. □□□□□□;
C. □□,□□+□□□_pct))"□□"
□□□□□□;
SELECT □□, □□, □□*12+salary*commission_pct "□□"
D. □□□□□□;
SELECT □□, □□, □□*12 + NVL(□□,0)*commission_pct, "□□"
Answer: (SHOW ANSWER)

NEW QUESTION: 79
□□□□ □□ 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

□□ SQL □□ □□□□□.

```
SQL>SELECT promo_name,CASE
                  WHEN promo_cost >=(SELECT AVG(promo_cost)
                                     FROM promotions
                                     WHERE promo_category='TV')
                  THEN 'HIGH'
                  ELSE 'LOW'
                  END COST_REMARK
FROM promotions;
```

□ □□ □□□ □□ □□□□ □□ □□?
A. □□□□ □□ □□□□□ □□ COST_REMARK□ □□□□□.
B. □□□□□ □□□ □□□□ □□□ □□□□□.
C. □□□□ □□□□ 'TV'□ □□ □□□□□ □□ COST_REMARK□ □□□□□.
D. CASE □□□□ □□ □□□□□ □□□ □ □□ □□□ □□□ □□□□□.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 80
Oracle □□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)
A. □□□□ □□□□ □□ □ □□□□.
B. □□ □□ □□□ □□□ □□□□□□□□ □□□□ □□□.
C. □□ □□□□ PUBLIC □□□□ □□ □ □□□□.
D. □□□□ □□□ □□□□ □□ □ □□□□.

E. □□□□□ □□ □□□ □□□□.

Answer: B,D (LEAVE A REPLY)

□□/□□: https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7001.htm

NEW QUESTION: 81

□□□□ □□ 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)

A. SELECT NVL(cust_credit_limit), '□□□ □ □□') "NEW CREDIT"
□□□□□□;

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

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: (SHOW ANSWER)

NEW QUESTION: 82

□□□□ □□ 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. □□ □□□ ITEM_CNT□ □□□ □ □□ □□□ □□□ □□□ □□□□□.

B. □□□□ □□□□□ □□□□ □□□ □□□ □□□□□.

C. FROM □□ □□ □□□ □□ □□□ □□ □□□ □ □□ □□□ □□□ □□□ □□□□□.

D. FROM □□ □□ □□□□ GROUP BY □□ □□□ □ □□ □□□ □□□ □□□ □□□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 83

□□ SELECT□□ □□□□ □□□□ □□ □□□ □□□□□□.

SELECT constraint_name, constraint_type, search_condition, r_constraint_name, delete_rule, status, FROM user_constraints
WHERE table_name = '□□';

CONSTRAINT_NAME	CON	SEARCH_CONDITION	R_CONSTRAINT_NAME	DELETE_RU
ORDER_DATE_NN	C	"ORDER_DATE" IS NOT NULL		
ORDER_CUSTOMER_ID _NN	C	"CUSTOMER_ID" IS NOT NULL		
ORDER_MODE_LOV	C	order _mode in ('direct', 'online')		
ORDER TOTAL MIN	C	order total >= 0		
ORDER PK	P			
ORDERS CUSTOMER ID	R		CUSTOMERS ID	SET NULL
ORDERS SALES REP	R		EMP EMP ID	SET NULL

□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)

A. □ □□ □□□ 'c'□ □□ □□ □□□ □□□□□.

B. DELETE_RULE □□□ □□ □□ □□ □□□□□ □□□ □ □□ □□□□ □□ □□ □□□ □□□□□.

C. R_CONSTRAINT_NAME □□ □□ □□□ □□ □□□ □□□□□.

D. STATUS □□ □□□□ □□ □□ □□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 84

□□ □□□ □□□□□□.

```
NSERT ALL
  WHEN order_total < 10000 THEN
    INTO small_orders
  WHEN order_total > 10000 AND order_total < 20000 THEN
    INTO medium_orders
  WHEN order_total > 200000 AND order_total < 20000 THEN
    INTO large_orders
  SELECT order_id, order_total, customer_id
    FROM orders;
```


INSERT □□ □□□□ □□ □□□ □□ □□□ □□ □□□□ □□ □□ □□?

A. □□ WHEN □□ □□ □□□ □□□□ □ □□ WHEN □ □□□ □□ □□□□□.

B. □ □□ WHEN □□ □□ □□□□□. □□□ □□□ □□ □□ WHEN □□ □□ □□□□□.

C. □ □□ WHEN □□ □□ □□□□□. □□□ false□□ □□ □□ WHEN □□ □□ □□□□□.

D. WHEN □ □ □□ □□ □□ □□ □□ □□□ □□ ELSE □□ □□□□ □□ □□□ insert □□ □□□ □□□□□.

Answer: A (LEAVE A REPLY)

http://psoug.org/definition/WHEN.htm

NEW QUESTION: 85
PROGRAMS □□□□ □□□ □□□□□□.

Name	Null?	Type
PROG_ID	NOT NULL	NUMBER (3)
PROG_COST		NUMBER (8, 2)
START_DATE	NOT NULL	DATE
END_DATE		DATE

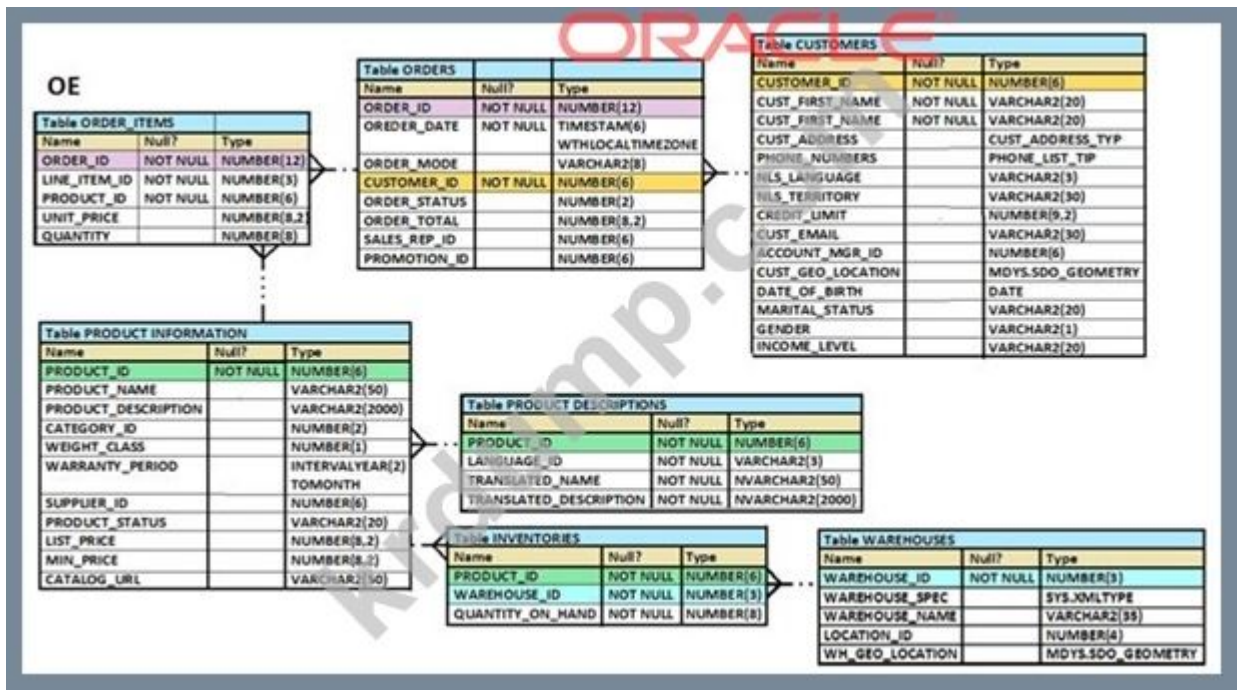
□□ □ □□ SQL □□ □□□□□ □□□□□?
A. SELECT NVL(TO_CHAR(MONTHS_BETWEEN(start-date, end_date)), '□□ □') FROM □□□□
B. SELECT TO_DATE(NVL(SYSDATE-END_DATE, SYSDATE)) FROM □□□□;
C. SELECT NVL (ADD_MONTHS (END_DATE,1) SYSDATE) FROM □□□□;
D. SELECT NVL(MONTHS_BETWEEN(start_date, end_date), '□□ □') FROM
□□□□□;
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 86
□□ SQL □□ □□□□□.

```
SELECT cust_id, cust_last_name "Last Name"
  FROM customers
 WHERE country_id = 10
  UNION
SELECT cust_id CUST_NO, cust_last_name
  FROM customers
 WHERE country_id = 30
```

□□□ □□□□□ □□□ □ □□ □ □□ order by □□ □□□□□.
A. "CUST NO"□ □□
B. CUST_NO □□
C. 2, 1□ □□
D. ORDER BY 2, cust_id
E. "□"□□ □□
Answer: C,D,E (LEAVE A REPLY)

NEW QUESTION: 87
□□□□ □□ 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: 88

□□□□ □□ □□□□ □□ □□? (2□□ □□□□□.)

A. SELECT □□ WHERE □□ □□ □□ □□□ □□ 3□□□□□ □□□ □ □□□□.

B. □□ □□□ 0□□ □□ □□□□ □□ □□ □□ □□□□□ □□□ □□ NULL□□□.

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

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

E. □ □□□ □ □□ □□ □□□ □□□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 89

BOOKStable□ □□ □□□ □□□□□.

```
Name                 Null?    Type
-------------------- -------- ----------------------------
TRANSACTION_ID       NOT NULL VARCHAR2(6)
TRANSACTION_DATE              DATE
AMOUNT                        NUMBER(10,2)
CUSTOMER_ID                   VARCHAR2(6)
```

□□□□□ 100□□ □□ □□□□.

□ □□□□ □□□ □□ □□□ □□□□ □□□□□□.

```
INSERT INTO books VALUES ('ADV112', 'Adventures of Tom Sawyer', NULL, NULL);

SAVEPOINT a;

DELETE FROM books;

ROLLBACK TO SAVEPOINT a;

ROLLBACK;
```

□□ □ □□ □□□ □□□□□? (2□□ □□□□□.)

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

B. □ □□ ROLLBACK □□□ □□□ □□□□□.

C. □ □□ ROLLBACK □□□ □□□ 101□ □□ □□□□ □□□ □□ □□□□□.

D. □ □□ ROLLBACK □□□ □□□ 101□ □□ □□□□ □□□ □□ □□ □□□ □□□ □□□.

E. □ □□ ROLLBACK □□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 90

BOOKS_TRANSACTIONStable□ □□ □□□ □□□□□.

```
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 □□□ □□□ □□□ □□□□□? (2□□ □□□□□.)

WHERE □□ □□ = SYSDATE AND (transaction_type = 'RM' OR member_id IN

A. ('A101', 'A102');

WHERE □□ □□ = SYSDATE AND (transaction_type = 'RM' AND (member_id

B. 'A101' □□ Member_id = 'A102');

C. ('A101', 'A102'));

WHERE(borrowed_date = SYSDATE AND transaction_type = 'RM') □□ member_id IN

D. A101' OR member_id = 'A102'));

WHERE □□ □□ = SYSDATE AND transaction_type = 'RM' OR member_id IN

E. ('A101', 'A102');

WHERE □□ □□ = SYSDATE AND (transaction_type = 'RM' AND member_id
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 91
□□ □□□□ □□□□□□.
SELECT 1 AS id, ' John' AS □□
□□□□
□□ □□
SELECT 1 , 'John' AS □□
□□□□
1□ □□
□□ □ □□□ □□□□□?
A. 2□
B. 1□
C. 0□
D. □□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 92
□□ □□□ □□ □□□□ □□ □□? (3□□ □□□□□.)
A. □□□ □□□□ □□□ □ □□□□.
B. □□ □□ □□□ □□□ □□□ □ □□□□.
C. GROUP BY □□ □□ SQL □□ □□ □□□ □ □□□□.
D. SELECT □□ □ □□□ □□□ □ □□□□.
E. SELECT □□□ □□ □ □□□ □□ □□□ □ □□□□.
Answer: (SHOW ANSWER)
□□
https://www.safaribooksonline.com/library/view/mastering-oracle-sql/0596006322/ch04.html

NEW QUESTION: 93
PROMOTIONS □□□□□ PROMO_BEGTN_DATE □□ □□□ □□□ DATE□□ □□ □□ □□□ DD-MON-RR□□□.
□□□ □□□ PROMO_BEGIN_DATE□ □□□□ □□□□ □□ □□□□ □□ □□?
A. PROMO_BEGIN_DATE-SYSDATE□ □□□ □□□□□.
B. TO_DATE(PROMO_BEGIN_DATE * 5)□ □□□ □□□□□.
C. TO_NUMBER(PROMO_BEGIN_DATE)-5□ □□□ □□□□□.

D. PROMO_BEGIN_DATE-SYSDATE□ □□□ □□□□□.
E. PROMO_BEGIN_DATE-5□ □□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 94
ENPLOYEES □□□□ □□□□ □□□□□.

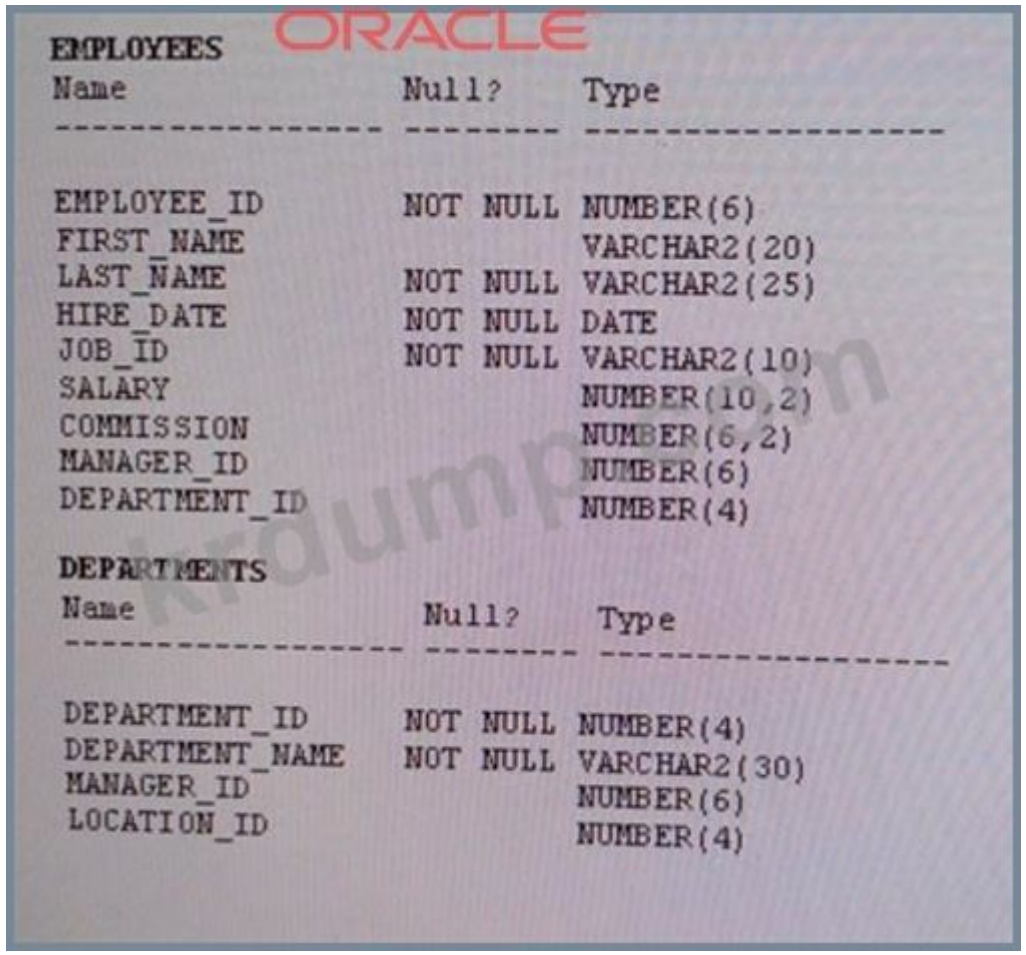
ENPLOYEE_ID	LAST_NAME	MONTHLY_SALART	MONTHLY_CONMISSION_PCT
101	Rochhar	24000	<NULL>
102	Ernet	17000	.5
103	Rajs	21000	.2
104	Lorontr	25000	<NULL>
105	morria	12000	<NULL>

□□ □ □ □□□ □□ □ □□ □□□ □□□□ □□□□ □□□□□?
A. SELCECT last_namo, (monthly_salary * 12) + (menthy_salary * 12 * NVL
(monthly_commission_pct, 0)) AS Annual_comp FROM □□
B. SELCECT last_namo, (monthly_salary * 12) + (menthy_salary * 12 *monthly_commission_pct)
AS Annual_comp FROM □□
C. SELCECT last_namo, (monthly_salary * 12) + (monthly_commission_pct * 12) AS Annual_comp
□□□□□□
D. SECECT last_namo, (menthy_salary +monthly_commission_pct) * 12 AS Annual_comp
□□□□□□;
Answer: (SHOW ANSWER)

NEW QUESTION: 95
□□ □□□ □□□□□.
SQL> SELECT prod_id, amount_sold
□□□□
ORDER BY amount_sold
□□ 5% □□ □□□□□.
□ □□□ □□□ □□□□□?
A. □□ □□ □□ □□□ 5%□ □□□□□.
B. SALES □□□□□ □□ □□ 5%□ □□□□□.
C. □□ □□ □□ □□□ 5%□ □□□□□.
D. ORDER BY □□ □□□ □□□□ □□□ □□□ □□□□□.
Answer: C (LEAVE A REPLY)
□□
□□:
https://oracle-base.com/articles/12c/row-limiting-clause-for-top-n-queries-12cr1

NEW QUESTION: 96

□□□□ □□ □□ □ □□ □□□□ □□□ □□□□□□.



```
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)
```

□□ □□ □□□ □□ □□ □□□□ □□□□□□ □□□.

-□□□ □□ □□□(□□ 2900 □ 2700)□□ □□□ □□□ □□□□□□□.

-□ □□□ □□ ID□ □□(□□ ID 2100)□ □□□□ □□ ID□ □□□□□.

- iocation_id 2100□ □□ □□□ □□ □□ □□ □□□ 1.1□□ □□□□□.

- location_id 2100□□ □□□ □□□□ □□ □□ □□□□ 1.5□□ □□□□□.

□□ □□□ □□□□□.

```
SQL> UPDATE employees
     SET department_id =
          (SELECT department_id
           FROM departments
           WHERE location_id = 2100),
        (salary, commission) =
          (SELECT 1.1*AVG(salary), 1.5*AVG(commission)
           FROM employees, departments
           WHERE departments.location_id IN(2900,2700,2100))
     WHERE department_id IN
          (SELECT department_id
           FROM departments
           WHERE location_id = 2900
           OR location_id = 2700);
```


□□□ □□□□□?

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

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

C. □□□□□ □□□□ □□□ □□□□□ □□□□□.

D. □□□□□ □□□□□ □□□ □□□□□ □□□□ □□□□.

Answer: D (LEAVE A REPLY)

NEW QUESTION: 97

Oracle □□□□□□ □□□□□ SQL □□ □□□ □□ □□□□ □□ □□ □□?

(□□ □□ □□ □□□□□.)

A. □□ □□ □ WHERE □□ □□□□ □□□□ SQL □□ □□ □□□□ □□□□ □□□□ □□□□ □□ □□ □□□ □□□□□□.

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

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

D. □□□□ □□ □□□ □□□□ □□□□ □□□ Oracle □□□ □□□□□□ □□□□ □□ □ □□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 98

EMPLOYEES □□□□ □□□ □□□□□□.

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)
COMISSION_PCT		NUMBER (2,2)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)

EMPLOYEE_ID□ MANAGER_ID □□□□ □□/□□ □□□ □□□□.

□□ □□□ □□, □□□ □ □□□□ □□□□□ □□□. □□□□□ □□ □□□ □□□□ □□□□ □□□□□. □□□ □□ MANAGER □□ '□□□ □□'□ □□□□□ □□□.

□□ SQL □□□ □□□ □□□ □□□□?

A. SELECT e.last_name, e.hire_date, NVL(m.last_name, '□□□ □□') ManagerFROM □□ e RIGHT OUTER JOIN □□ mON(e.manager_id = m.employee_id);

B. SELECT e.last_name, e.hire_date, NVL(m.last_name, '□□□ □□') ManagerFROM □□ e JOIN □□ mON(e.manager_id = m.employee_id);

C. SELECT e.last_name, e.hire_date, NVL(m.last_name, '□□□ □□') ManagerFROM □□ e LEFT OUTER JOIN □□ mON(e.manager_id = m.employee_id);

D. SELECT e.last_name, e.hire_date, NVL(m.last_name, 'No Manager') ManagerFROM □□ e NATURAL JOIN □□ mON(e.manager_id = m.employee_id).

Answer: C (LEAVE A REPLY)

NEW QUESTION: 99

□□□ □□□ □□□□ □□ SQL □□ □□□□□□.

□□□ □□□

(emp_no NUMBER (2) CONSTRAINT emp_emp_no_pk □□ □,

□□ VARCHAR 2(15),

□□ NUMBER (8, 2),

mgr_no NUMBER(2) □□ □□ emp_mgr_fk □□ emp (emp_no));

ALTER TABLE EMP

DISABLE CONSTRAINT emp_emp_no_pk CASCADE;

ALTER TABLE EMP

ENABLE □□ □□ emp_emp_no_pk;

□□ □ EMP_MGR_FK□ □□□ □□□ □□□?

A. □□□ □ □□□□□.

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

C. □□□□□ □□ □□□□□.

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

Answer: B (LEAVE A REPLY)

NEW QUESTION: 100
INTERSECT □□□□ □□ □□□□ □□ □□?
A. □□ SELECT □□ □ □□□ □□□□ □□□.
B. NULL □□ □□□□□.
C. □□ □□□□ □□□ □□□ □□ □□□ □□□□□.
D. □□□ □□ SELECT □□ □□ □ □□ □□□ □□□ □□□□ □□□.
Answer: D (LEAVE A REPLY)
INTERSECT □ □□□ □□ □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□□□.
□□ □□□ □□□□ □□□ □□ □□ □□□ □□ □ □□□ □□ □□ □□□ □ □□ □□□ □ □□□ □□□□□.
□□:
http://oraclexpert.com/using-the-set-operators/