

**1Z0-051\_CertifyMe\_formatted\_SSX**

Number: 000-000

Passing Score: 600

Time Limit: 120 min

File Version: 1.0



Oracle 1Z0-051

Oracle Database: SQL Fundamentals I

175 Q&A

Version: O9.0

## **Exam A**

### **QUESTION 1**

View the Exhibit and examine the structure of the SALES, CUSTOMERS, PRODUCTS, and TIMES tables.

The PROD\_ID column is the foreign key in the SALES table, which references the PRODUCTS table.

Similarly, the CUST\_ID and TIME\_ID columns are also foreign keys in the SALES table referencing the CUSTOMERS and TIMES tables, respectively.

Evaluate the following CREATE TABLE command:

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

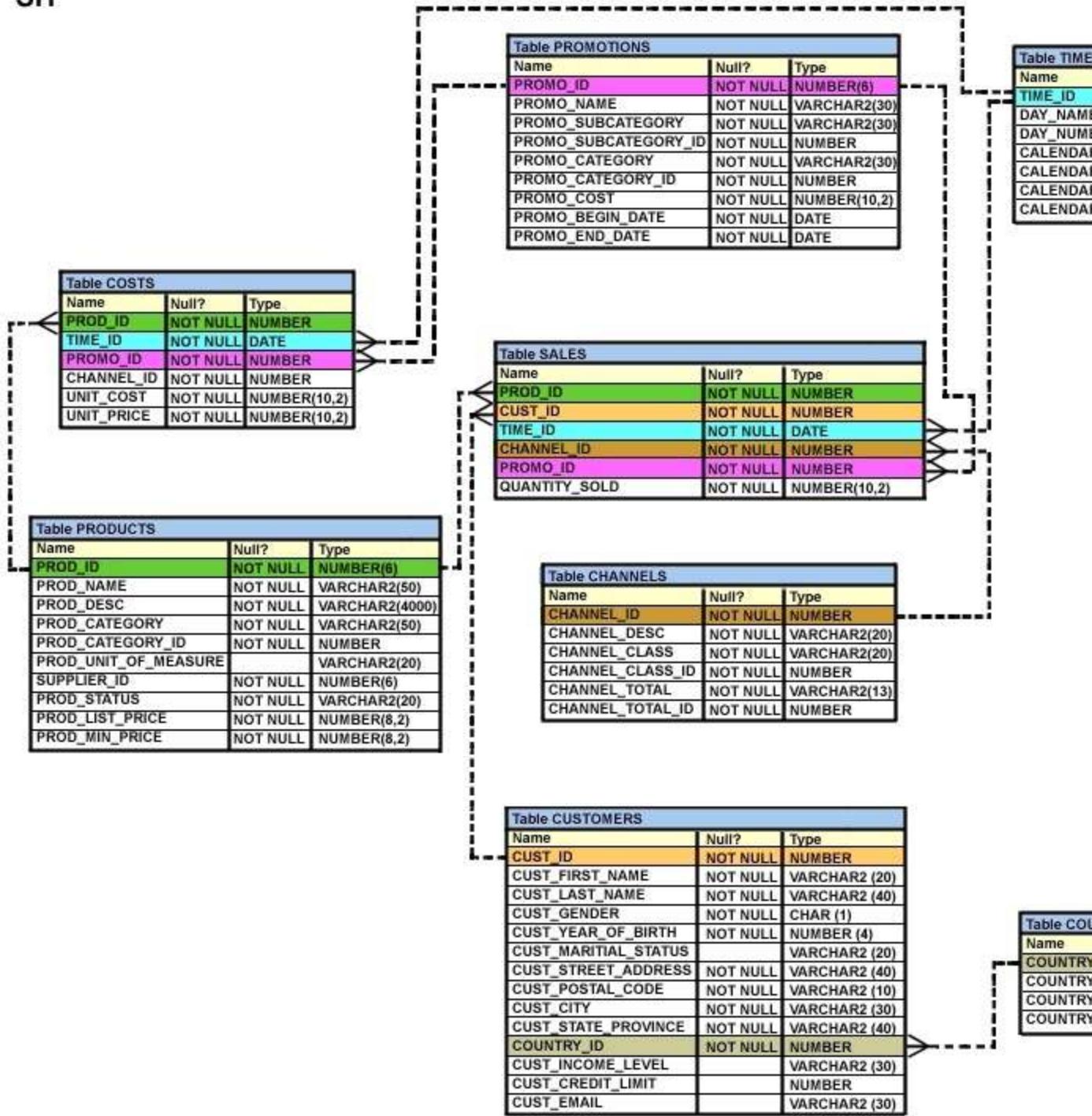
```
AS
```

```
SELECT prod_id, cust_id, time_id
```

```
FROM sales;
```

Which statement is true regarding the above command?

SH



- A. The NEW\_SALES table would not get created because the DEFAULT value cannot be specified in the column definition.
- B. The NEW\_SALES table would get created and all the NOT NULL constraints defined on the specified columns would be passed to the new table.
- C. The NEW\_SALES table would not get created because the column names in the CREATE TABLE command and the SELECT clause do not match.
- D. The NEW\_SALES table would get created and all the FOREIGN KEY constraints defined on the specified columns would be passed to the new table.

Answer: B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 2**

View the Exhibit to examine the description for the SALES table.

Which views can have all DML operations performed on it? (Choose all that apply.)

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)

- A. CREATE VIEW v3  
AS SELECT \* FROM SALES  
WHERE cust\_id = 2034  
WITH CHECK OPTION;
- B. CREATE VIEW v1  
AS SELECT \* FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
WITH CHECK OPTION;
- C. CREATE VIEW v2  
AS SELECT prod\_id, cust\_id, time\_id FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
WITH CHECK OPTION;
- D. CREATE VIEW v4  
AS SELECT prod\_id, cust\_id, SUM(quantity\_sold) FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
GROUP BY prod\_id, cust\_id  
WITH CHECK OPTION;

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

**QUESTION 3**

You need to extract details of those products in the SALES table where the PROD\_ID column contains

the string '\_D123'.

Which WHERE clause could be used in the SELECT statement to get the required output?

- A. WHERE prod\_id LIKE '%\_D123%' ESCAPE '\_'
- B. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\'
- C. WHERE prod\_id LIKE '%\_D123%' ESCAPE '%\_'
- D. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\\_'

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 4**

Which two statements are true regarding single row functions? (Choose two.)

- A. They accept only a single argument.
- B. They can be nested only to two levels.
- C. Arguments can only be column values or constants.
- D. They always return a single result row for every row of a queried table.
- E. They can return a data type value different from the one that is referenced.

**Answer:** DE

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 5

Which SQL statements would display the value 1890.55 as \$1,890.55? (Choose three.)

- A. SELECT TO\_CHAR(1890.55,'\$0G000D00')  
FROM DUAL;
- B. SELECT TO\_CHAR(1890.55,'\$9,999V99')  
FROM DUAL;
- C. SELECT TO\_CHAR(1890.55,'\$99,999D99')  
FROM DUAL;
- D. SELECT TO\_CHAR(1890.55,'\$99G999D00')  
FROM DUAL;
- E. SELECT TO\_CHAR(1890.55,'\$99G999D99')  
FROM DUAL;

**Answer:** ADE

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 6

Examine the structure of the SHIPMENTS table:

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)

You want to generate a report that displays the PO\_ID and the penalty amount to be paid if the

SHIPMENT\_DATE is later than one month from the PO\_DATE. The penalty is \$20 per day.

Evaluate the following two queries:

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;

Which statement is true regarding the above commands?

- A. Both execute successfully and give correct results.
- B. Only the first query executes successfully but gives a wrong result.
- C. Only the first query executes successfully and gives the correct result.
- D. Only the second query executes successfully but gives a wrong result.
- E. Only the second query executes successfully and gives the correct result.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 7

Which two statements are true regarding the USING and ON clauses in table joins? (Choose two.)

- A. Both USING and ON clauses can be used for equijoins and nonequijoins.
- B. A maximum of one pair of columns can be joined between two tables using the ON clause.
- C. The ON clause can be used to join tables on columns that have different names but compatible data types.
- D. The WHERE clause can be used to apply additional conditions in SELECT statements containing the ON or the USING clause.

**Answer:** CD

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 8

View the Exhibit and examine the structure of the CUSTOMERS table.

Which two tasks would require subqueries or joins to be executed in a single statement? (Choose two.)

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. listing of customers who do not have a credit limit and were born before 1980
- B. finding the number of customers, in each city, whose marital status is 'married'
- C. finding the average credit limit of male customers residing in 'Tokyo' or 'Sydney'
- D. listing of those customers whose credit limit is the same as the credit limit of customers residing in the city 'Tokyo'
- E. finding the number of customers, in each city, whose credit limit is more than the average credit limit of all the customers

**Answer:** DE

**Section:** (none)

**Explanation/Reference:**

**QUESTION 9**

Which statement is true regarding the INTERSECT operator?

- A. It ignores NULL values.
- B. Reversing the order of the intersected tables alters the result.
- C. The names of columns in all SELECT statements must be identical.
- D. The number of columns and data types must be identical for all SELECT statements in the query.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 10**

View the Exhibit; examine the structure of the PROMOTIONS table.

Each promotion has a duration of at least seven days .

Your manager has asked you to generate a report, which provides the weekly cost for each promotion done to I date.

Which query would achieve the required result?

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\_cost/promo\_end\_date-promo\_begin\_date/7 FROM promotions;
- B. SELECT promo\_name,(promo\_cost/promo\_end\_date-promo\_begin\_date)/7 FROM promotions;
- C. SELECT promo\_name, promo\_cost/(promo\_end\_date-promo\_begin\_date/7) FROM promotions;
- D. SELECT promo\_name, promo\_cost/((promo\_end\_date-promo\_begin\_date)/7) FROM promotions;

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 11**

View the Exhibit and examine the structure of the PRODUCTS table.

All products have a list price.

You issue the following command to display the total price of each product after a discount of 25% and a tax of 15% are applied on it. Freight charges of \$100 have to be applied to all the products.

```
SQL>SELECT prod_name, prod_list_price -(prod_list_price*(25/100))
+(prod_list_price -(prod_list_price*(25/100)))*(15/100))+100
```

AS "TOTAL PRICE"  
FROM products;

What would be the outcome if all the parentheses are removed from the above statement?

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)

- A. It produces a syntax error.
- B. The result remains unchanged.
- C. The total price value would be lower than the correct value.
- D. The total price value would be higher than the correct value.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 12

You need to produce a report where each customer's credit limit has been incremented by \$1000. In the output, the customer's last name should have the heading Name and the incremented credit limit should be labeled New Credit Limit. The column headings should have only the first letter of each word in uppercase .

Which statement would accomplish this requirement?

- A. SELECT cust\_last\_name Name, cust\_credit\_limit + 1000  
"New Credit Limit"  
FROM customers;
- B. SELECT cust\_last\_name AS Name, cust\_credit\_limit + 1000  
AS New Credit Limit  
FROM customers;
- C. SELECT cust\_last\_name AS "Name", cust\_credit\_limit + 1000  
AS "New Credit Limit"  
FROM customers;
- D. SELECT INITCAP(cust\_last\_name) "Name", cust\_credit\_limit + 1000 INITCAP("NEW CREDIT LIMIT")  
FROM customers;

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 13

View the Exhibit and examine the structure of the PRODUCTS table.

You need to generate a report in the following format:

CATEGORIES

5MP Digital Photo Camera's category is Photo  
Y Box's category is Electronics

Envoy Ambassador's category is Hardware

Which two queries would give the required output? (Choose two.)

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)

- A. SELECT prod\_name q""s category is ' prod\_category CATEGORIES FROM products;
- B. SELECT prod\_name q'!s ]category is ' prod\_category CATEGORIES FROM products;
- C. SELECT prod\_name q'\s\'' category is ' prod\_category CATEGORIES FROM products;
- D. SELECT prod\_name q'<s >'category is ' prod\_category CATEGORIES FROM products;

**Answer:** CD

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 14

Using the CUSTOMERS table, you need to generate a report that shows 50% of each credit amount in each income level. The report should NOT show any repeated credit amounts in each income level.

Which query would give the required result?

- A. SELECT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50 AS "50% Credit Limit"  
FROM customers;
- B. SELECT DISTINCT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50 AS "50% Credit Limit"  
FROM customers;
- C. SELECT DISTINCT cust\_income\_level '' cust\_credit\_limit \* 0.50 AS "50% Credit Limit"  
FROM customers;
- D. SELECT cust\_income\_level '' cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 15

View the Exhibit and examine the data in the CUSTOMERS table.

Evaluate the following query:

```
SQL> SELECT cust_name AS "NAME", cust_credit_limit/2 AS MIDPOINT,MIDPOINT+100 AS "MAX  
LOWER LIMIT"  
FROM customers;
```

The above query produces an error on execution.

What is the reason for the error?

- A. An alias cannot be used in an expression.
- B. The a alias NAME should not be enclosed with in double quotation marks .
- C. The MIDPOINT+100 expression gives an error because CUST\_CREDIT\_LIMIT contains NULL values.
- D. The a alias MIDPOINT should be enclosed with in double quotation marks for the CUST\_CREDIT\_LIMIT/2 expression .

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 16**

Evaluate the following query:

```
SQL> SELECT promo_name q'{s start date was }' promo_begin_date  
AS "Promotion Launches"  
FROM promotions;
```

What would be the outcome of the above query?

- A. It produces an error because flower braces have been used.
- B. It produces an error because the data types are not matching.
- C. It executes successfully and introduces an 's at the end of each promo\_name in the output.
- D. It executes successfully and displays the literal " {s start date was } " for each row in the output.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 17**

View the Exhibit and examine the data in the EMPLOYEES table.

You want to generate a report showing the total compensation paid to each employee to date.

You issue the following query:

```
SQL>SELECT ename ' joined on ' hiredate  
' , the total compensation paid is '  
TO_CHAR(ROUND(ROUND(SYSDATE-hiredate)/365) * sal + comm)  
"COMPENSATION UNTIL DATE"  
FROM employees;
```

What is the outcome?

**EMPLOYEES**

ENAME	HIREDATE	SAL	COMM
SMITH	17-DEC-00	800	
ALLEN	20-FEB-99	1600	300
WARD	22-FEB-95	1250	500
JONES	02-APR-98	2975	
MARTIN	28-SEP-99	1250	1400
BLAKE	01-MAY-97	2850	

- A. It generates an error because the alias is not valid.
- B. It executes successfully and gives the correct output.
- C. It executes successfully but does not give the correct output.
- D. It generates an error because the usage of the ROUND function in the expression is not valid.
- E. It generates an error because the concatenation operator can be used to combine only two items.

**Answer:** C**Section:** (none)**Explanation/Reference:****QUESTION 18**

Examine the structure of the PROMOTIONS table:

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)

The management wants to see a report of unique promotion costs in each promotion category.

Which query would achieve the required result?

- A. SELECT DISTINCT promo\_cost, promo\_category FROM promotions;
- B. SELECT promo\_category, DISTINCT promo\_cost FROM promotions;
- C. SELECT DISTINCT promo\_cost, DISTINCT promo\_category FROM promotions;
- D. SELECT DISTINCT promo\_category, promo\_cost FROM promotions ORDER BY 1;

**Answer:** D**Section:** (none)**Explanation/Reference:****QUESTION 19**

Evaluate the following query:

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

What is the correct output of the above query?

- A. +25-00 , +54-02, +00 11:12:10.123457
- B. +00-300, +54-02, +00 11:12:10.123457
- C. +25-00 , +00-650, +00 11:12:10.123457
- D. +00-300 , +00-650, +00 11:12:10.123457

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 20**

Which three statements are true regarding the data types in Oracle Database 10g/11g? (Choose three.)

- A. Only one LONG column can be used per table.
- B. A TIMESTAMP data type column stores only time values with fractional seconds.
- C. The BLOB data type column is used to store binary data in an operating system file.
- D. The minimum column width that can be specified for a VARCHAR2 data type column is one.
- E. The value for a CHAR data type column is blank-padded to the maximum defined column width.

**Answer:** ADE

**Section:** (none)

**Explanation/Reference:**

**QUESTION 21**

Examine the description of the EMP\_DETAILS table given below:

```
name NULL TYPE  
EMP_ID NOT NULL NUMBER  
EMP_NAME NOT NULL VARCHAR2 (40)  
EMP_IMAGE LONG
```

Which two statements are true regarding SQL statements that can be executed on the EMP\_DETAIL table?

(Choose two.)

- A. An EMP\_IMAGE column can be included in the GROUP BY clause.
- ` An EMP\_IMAGE column cannot be included in the ORDER BY clause.
- C. You cannot add a new column to the table with LONG as the data type.
- D. You can alter the table to include the NOT NULL constraint on the EMP\_IMAGE column.

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 22**

You need to create a table for a banking application. One of the columns in the table has the following

requirements:

- 1) You want a column in the table to store the duration of the credit period.
- 2) The data in the column should be stored in a format such that it can be easily added and subtracted with DATE data type without using conversion functions.
- 3) The maximum period of the credit provision in the application is 30 days.

4) The interest has to be calculated for the number of days an individual has taken a credit for.

Which data type would you use for such a column in the table?

- A. DATE
- B. NUMBER
- C. TIMESTAMP
- D. INTERVAL DAY TO SECOND
- E. INTERVAL YEAR TO MONTH

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 23**

Examine the structure proposed for the TRANSACTIONS table:

name Null Type

TRANS\_ID NOT NULL NUMBER(6)

CUST\_NAME NOT NULL VARCHAR2(20)

CUST\_STATUS NOT NULL CHAR

TRANS\_DATE NOT NULL DATE

TRANS\_VALIDITY VARCHAR2

CUST\_CREDIT\_LIMIT NUMBER

Which statements are true regarding the creation and storage of data in the above table structure? (Choose all that apply.)

- A. The CUST\_STATUS column would give an error.
- B. The TRANS\_VALIDITY column would give an error.
- C. The CUST\_STATUS column would store exactly one character.
- D. The CUST\_CREDIT\_LIMIT column would not be able to store decimal values.
- E. The TRANS\_VALIDITY column would have a maximum size of one character.
- F. The TRANS\_DATE column would be able to store day, month, century, year, hour, minutes, seconds, and fractions of seconds.

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 24**

Examine the structure proposed for the TRANSACTIONS table:

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)

Which two statements are true regarding the storage of data in the above table structure? (Choose two.)

- A. The TRANS\_DATE column would allow storage of dates only in the dd-mon-yyyy format.
- B. The CUST\_CREDIT\_VALUE column would allow storage of positive and negative integers.
- C. The TRANS\_VALIDITY column would allow storage of a time interval in days, hours, minutes, and seconds.
- D. The CUST\_STATUS column would allow storage of data up to the maximum VARCHAR2 size of 4,000 characters.

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 25

Resume (character large object [CLOB] data type), which contains the resume submitted by the employee

Which is the correct syntax to create this table?

- A. CREATE TABLE EMP\_1  
(emp\_id NUMBER(4),  
emp\_name VARCHAR2(25),  
start\_date DATE,  
e\_status VARCHAR2(10) DEFAULT 'ACTIVE',  
resume CLOB(200));
- B. CREATE TABLE 1\_EMP  
(emp\_id NUMBER(4),  
emp\_name VARCHAR2(25),  
start\_date DATE,  
emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',  
resume CLOB);
- C. CREATE TABLE EMP\_1  
(emp\_id NUMBER(4),  
emp\_name VARCHAR2(25),  
start\_date DATE,  
emp\_status VARCHAR2(10) DEFAULT "ACTIVE",  
resume CLOB);
- D. CREATE TABLE EMP\_1  
(emp\_id NUMBER,  
emp\_name VARCHAR2(25),  
start\_date DATE,  
emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',  
resume CLOB);

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 26

Which is the valid CREATE TABLE statement?

- A. CREATE TABLE emp9\$# (emp\_no NUMBER (4));
- B. CREATE TABLE 9emp\$# (emp\_no NUMBER(4));
- C. CREATE TABLE emp\*123 (emp\_no NUMBER(4));
- D. CREATE TABLE emp9\$# (emp\_no NUMBER(4), date DATE);

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 27**

Which two statements are true regarding tables? (Choose two.)

- A. A table name can be of any length.
- B. A table can have any number of columns.
- C. A column that has a DEFAULT value cannot store null values.
- D. A table and a view can have the same name in the same schema.
- E. A table and a synonym can have the same name in the same schema.
- F. The same table name can be used in different schemas in the same database.

**Answer:** EF

**Section:** (none)

**Explanation/Reference:**

**QUESTION 28**

Which two statements are true regarding constraints? (Choose two.)

- A. A foreign key cannot contain NULL values.
- B. A column with the UNIQUE constraint can contain NULL values.
- C. A constraint is enforced only for the INSERT operation on a table.
- D. A constraint can be disabled even if the constraint column contains data.
- E. All constraints can be defined at the column level as well as the table level.

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 29**

Which two statements are true regarding constraints? (Choose two.)

- A. A foreign key cannot contain NULL values.
- B. The column with a UNIQUE constraint can store NULLS .
- C. A constraint is enforced only for an INSERT operation on a table.
- D. You can have more than one column in a table as part of a primary key.

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 30**

Evaluate the following CREATE TABLE commands:

CREATE TABLE orders

```
(ord_no NUMBER(2) CONSTRAINT ord_pk PRIMARY KEY,  
ord_date DATE,  
cust_id NUMBER(4));
```

CREATE TABLE ord\_items

```
(ord_no NUMBER(2),  
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);
```

The above command fails when executed. What could be the reason?

- A. SYSDATE cannot be used with the CHECK constraint.
- B. The BETWEEN clause cannot be used for the CHECK constraint.
- C. The CHECK constraint cannot be placed on columns having the DATE data type.
- D. ORD\_NO and ITEM\_NO cannot be used as a composite primary key because ORD\_NO is also the FOREIGN KEY.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 31**

Evaluate the following SQL commands:

```
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));
```

The command to create a table fails. Identify the reason for the SQL statement failure? (Choose all that apply.)

- A. You cannot use SYSDATE in the condition of a CHECK constraint.
- B. You cannot use the BETWEEN clause in the condition of a CHECK constraint.
- C. You cannot use the NEXTVAL sequence value as a DEFAULT value for a column.
- D. You cannot use ORD\_NO and ITEM\_NO columns as a composite primary key because ORD\_NO is also the FOREIGN KEY.

**Answer:** AC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 32**

Which CREATE TABLE statement is valid?

- A. CREATE TABLE ord\_details  
(ord\_no NUMBER(2) PRIMARY KEY,  
item\_no NUMBER(3) PRIMARY KEY,  
ord\_date DATE NOT NULL);
- B. CREATE TABLE ord\_details  
(ord\_no NUMBER(2) UNIQUE, NOT NULL,  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT SYSDATE NOT NULL);
- C. CREATE TABLE ord\_details  
(ord\_no NUMBER(2),  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT NOT NULL,  
CONSTRAINT ord\_uq UNIQUE (ord\_no),  
CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));
- D. CREATE TABLE ord\_details  
(ord\_no NUMBER(2),  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT SYSDATE NOT NULL,  
CONSTRAINT ord\_pk PRIMARY KEY (ord\_no, item\_no));

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

### QUESTION 33

You want to create an ORD\_DETAIL table to store details for an order placed having the following

business requirement:

- 1) The order ID will be unique and cannot have null values.
- 2) The order date cannot have null values and the default should be the current date.
- 3) The order amount should not be less than 50.
- 4) The order status will have values either shipped or not shipped.
- 5) The order payment mode should be cheque, credit card, or cash on delivery (COD).

Which is the valid DDL statement for creating the ORD\_DETAIL table?

- A. CREATE TABLE ord\_details  
(ord\_id NUMBER(2) CONSTRAINT ord\_id\_nn NOT NULL,  
ord\_date DATE DEFAULT SYSDATE NOT NULL,  
ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min  
CHECK (ord\_amount > 50),  
ord\_status VARCHAR2(15) CONSTRAINT ord\_status\_chk  
CHECK (ord\_status IN ('Shipped', 'Not Shipped')),  
ord\_pay\_mode VARCHAR2(15) CONSTRAINT ord\_pay\_chk  
CHECK (ord\_pay\_mode IN ('Cheque', 'Credit Card',  
'Cash On Delivery')));
- B. CREATE TABLE ord\_details  
(ord\_id NUMBER(2) CONSTRAINT ord\_id\_uk UNIQUE NOT NULL,  
ord\_date DATE DEFAULT SYSDATE NOT NULL,  
ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min  
CHECK (ord\_amount > 50),  
ord\_status VARCHAR2(15) CONSTRAINT ord\_status\_chk  
CHECK (ord\_status IN ('Shipped', 'Not Shipped')),  
ord\_pay\_mode VARCHAR2(15) CONSTRAINT ord\_pay\_chk  
CHECK (ord\_pay\_mode IN ('Cheque', 'Credit Card',  
'Cash On Delivery')));
- C. CREATE TABLE ord\_details  
(ord\_id NUMBER(2) CONSTRAINT ord\_id\_pk PRIMARY KEY,  
ord\_date DATE DEFAULT SYSDATE NOT NULL,  
ord\_amount NUMBER(5, 2) CONSTRAINT ord\_amount\_min  
CHECK (ord\_amount >= 50),

```

ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));

D. CREATE TABLE ord_details
(ord_id NUMBER(2),
ord_date DATE NOT NULL DEFAULT SYSDATE,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount >= 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));

```

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 34

You created an ORDERS table with the following description:

```

name Null Type
ORD_ID NOT NULL NUMBER(2)
CUST_ID NOT NULL NUMBER(3)
ORD_DATE NOT NULL DATE
ORD_AMOUNT NOT NULL NUMBER (10,2)

```

You inserted some rows in the table. After some time, you want to alter the table by creating the PRIMARY KEY constraint on the ORD\_ID column. Which statement is true in this scenario?

- A. You cannot have two constraints on one column.
- B. You cannot add a primary key constraint if data exists in the column.
- C. The primary key constraint can be created only at the time of table creation .
- D. You can add the primary key constraint even if data exists, provided that there are no duplicate values.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 35

Which two statements are true regarding constraints? (Choose two.)

- A. A table can have only one primary key and one foreign key.
- B. A table can have only one primary key but multiple foreign keys.
- C. Only the primary key can be defined at the column and table levels.
- D. The foreign key and parent table primary key must have the same name.
- E. Both primary key and foreign key constraints can be defined at both column and table levels.

**Answer:** BE

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 36

Examine the following SQL commands:

```
SQL>CREATE TABLE products (
    prod_id NUMBER(3) CONSTRAINT p_ck CHECK (prod_id > 0),
    prod_name CHAR(30),
    prod_qty NUMBER(6),
    CONSTRAINT p_name NOT NULL,
    CONSTRAINT prod_pk PRIMARY KEY (prod_id));
SQL>CREATE TABLE warehouse (
    warehouse_id NUMBER(4),
    roomno NUMBER(10) CONSTRAINT r_id CHECK(roomno BETWEEN 101 AND 200),
    location VARCHAR2(25),
    prod_id NUMBER(3),
    CONSTRAINT wr_pr_pk PRIMARY KEY (warehouse_id,prod_id),
    CONSTRAINT prod_fk FOREIGN KEY (prod_id) REFERENCES products(prod_id);
```

Which statement is true regarding the execution of the above SQL commands?

- A. Both commands execute successfully.
- B. The first CREATE TABLE command generates an error because the NULL constraint is not valid.
- C. The second CREATE TABLE command generates an error because the CHECK constraint is not valid.
- D. The first CREATE TABLE command generates an error because CHECK and PRIMARY KEY constraints cannot be used for the same column.
- E. The first CREATE TABLE command generates an error because the column PROD\_ID cannot be used in the PRIMARY KEY and FOREIGN KEY constraints.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 37**

You issued the following command to drop the PRODUCTS table:

```
SQL> DROP TABLE products;
```

What is the implication of this command? (Choose all that apply.)

- A. All data along with the table structure is deleted.
- B. The pending transaction in the session is committed.
- C. All indexes on the table will remain but they are invalidated.
- D. All views and synonyms will remain but they are invalidated.
- E. All data in the table are deleted but the table structure will remain.

**Answer:** ABD

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 38**

Which two statements are true regarding views? (Choose two.)

- A. A simple view in which column aliases have been used cannot be updated.

- B. Rows cannot be deleted through a view if the view definition contains the DISTINCT keyword.
- C. Rows added through a view are deleted from the table automatically when the view is dropped.
- D. The OR REPLACE option is used to change the definition of an existing view without dropping and re-creating it.
- E. The WITH CHECK OPTION constraint can be used in a view definition to restrict the columns displayed through the view.

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 39**

Evaluate the following command:

```
CREATE TABLE employees
(employee_id NUMBER(2) PRIMARY KEY,
last_name VARCHAR2(25) NOT NULL,
department_id NUMBER(2)NOT NULL,
job_id VARCHAR2(8),
salary NUMBER(10,2));
```

You issue the following command to create a view that displays the IDs and last names of the sales staff in the organization:

```
CREATE OR REPLACE VIEW sales_staff_vu AS4 ? 4;
SELECT employee_id, last_name,job_id4 ? 4;
FROM employees4 ? 4;
WHERE job_id LIKE 'SA_%'4 ? 4;
WITH CHECK OPTION;
```

Which two statements are true regarding the above view? (Choose two.)

- A. It allows you to insert rows into the EMPLOYEES table .
- B. It allows you to delete details of the existing sales staff from the EMPLOYEES table.
- C. It allows you to update job IDs of the existing sales staff to any other job ID in the EMPLOYEES table.
- D. It allows you to insert IDs, last names, and job IDs of the sales staff from the view if it is used in multitable INSERT statements.

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 40**

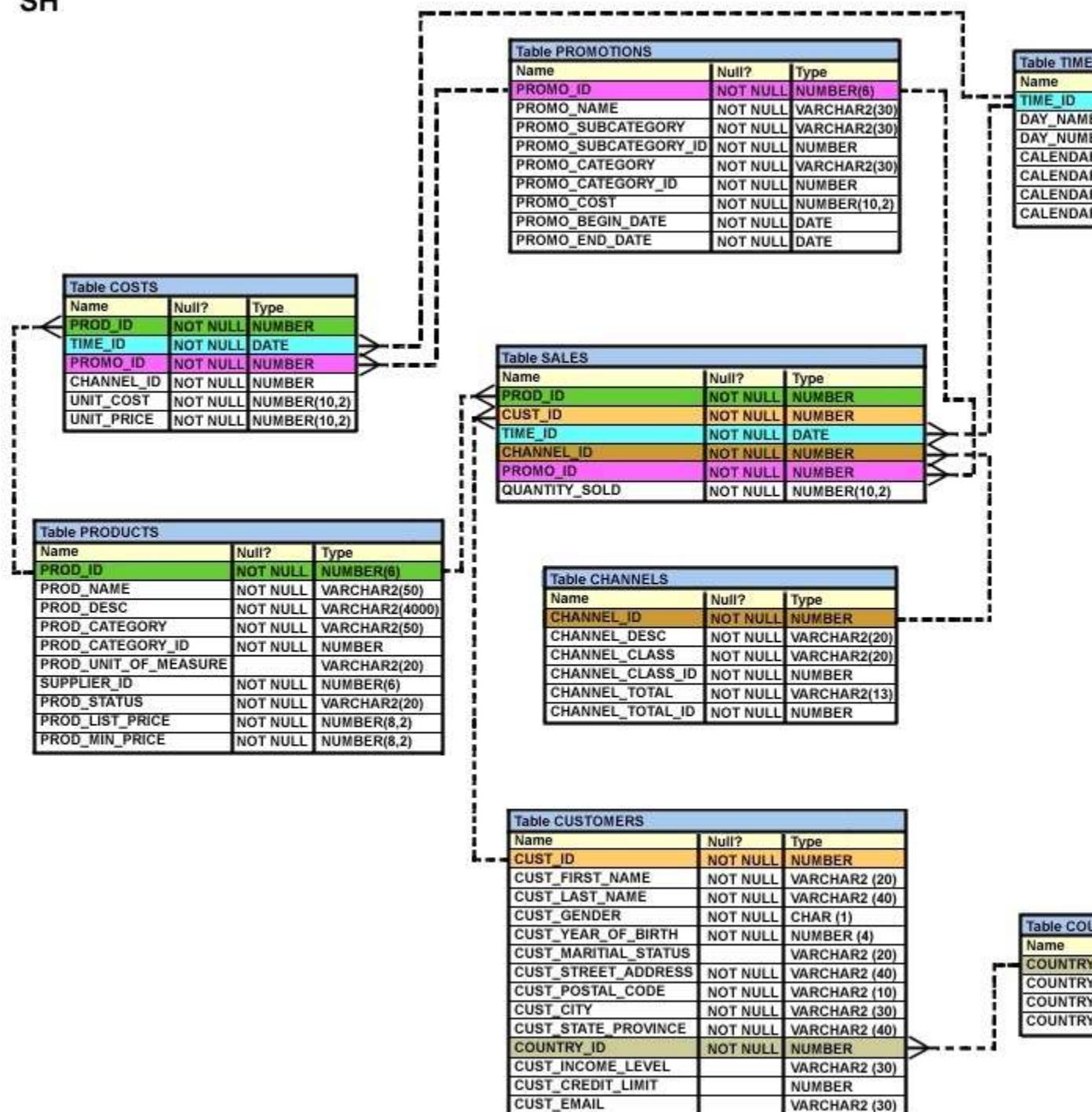
View the Exhibit to examine the description for the SALES and PRODUCTS tables.

You want to create a SALE\_PROD view by executing the following SQL statement:

```
CREATE VIEW sale_prod
AS SELECT p.prod_id, cust_id, SUM(quantity_sold) "Quantity" , SUM(prod_list_price) "Price"
FROM products p, sales s
WHERE p.prod_id=s.prod_id
GROUP BY p.prod_id, cust_id;
```

Which statement is true regarding the execution of the above statement?

**SH**



- The view will be created and you can perform DML operations on the view.
- The view will be created but no DML operations will be allowed on the view.
- The view will not be created because the join statements are not allowed for creating a view.
- The view will not be created because the GROUP BY clause is not allowed for creating a view.

**Answer:** B  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 41**

Which two statements are true regarding views? (Choose two.)

- A. A subquery that defines a view cannot include the GROUP BY clause.
- B. A view that is created with the subquery having the DISTINCT keyword can be updated.
- C. A view that is created with the subquery having the pseudo column ROWNUM keyword cannot be updated.
- D. A data manipulation language ( DML ) operation can be performed on a view that is created with the subquery having all the NOT NULL columns of a table.

**Answer:** CD  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 42**

Which three statements are true regarding views? (Choose three.)

- A. Views can be created only from tables.
- B. Views can be created from tables or other views.
- C. Only simple views can use indexes existing on the underlying tables.
- D. Both simple and complex views can use indexes existing on the underlying tables.
- E. Complex views can be created only on multiple tables that exist in the same schema.
- F. Complex views can be created on multiple tables that exist in the same or different schemas.

**Answer:** BDF  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 43**

Evaluate the following CREATE SEQUENCE statement:

```
CREATE SEQUENCE seq1
START WITH 100
INCREMENT BY 10
MAXVALUE 200
CYCLE
NOCACHE;
```

The SEQ1 sequence has generated numbers up to the maximum limit of 200. You issue the following SQL statement:

```
SELECT seq1.nextval FROM dual;
```

What is displayed by the SELECT statement?

- A. 1
- B. 10
- C. 100
- D. an error

**Answer:** A  
**Section:** (none)

**Explanation/Reference:**  
**QUESTION 44**

View the Exhibit and examine the structure of the ORD table.

Evaluate the following SQL statements that are executed in a user session in the specified order:

```
CREATE SEQUENCE ord_seq;  
SELECT ord_seq.nextval  
FROM dual;  
INSERT INTO ord  
VALUES (ord_seq.CURRVAL, '25-jan-2007',101);  
UPDATE ord  
SET ord_no= ord_seq.NEXTVAL  
WHERE cust_id =101;
```

What would be the outcome of the above statements?

## ORD

Name	Null?	Type
ORD_NO	NOT NULL	NUMBER(2)
ORD_DATE		DATE
CUST_ID		NUMBER(4)

- A. All the statements would execute successfully and the ORD\_NO column would contain the value 2 for the CUST\_ID 101.
- B. The CREATE SEQUENCE command would not execute because the minimum value and maximum value for the sequence have not been specified.
- C. The CREATE SEQUENCE command would not execute because the starting value of the sequence and the increment value have not been specified.
- D. All the statements would execute successfully and the ORD\_NO column would have the value 20 for the CUST\_ID 101 because the default CACHE value is 20.

**Answer:** A  
**Section:** (none)

**Explanation/Reference:**  
**QUESTION 45**

Which two statements are true about sequences created in a single instance database? (Choose two.)

- A. The numbers generated by a sequence can be used only for one table.
- B. DELETE <sequencename> would remove a sequence from the database.
- C. CURRVAL is used to refer to the last sequence number that has been generated.
- D. When the MAXVALUE limit for a sequence is reached, you can increase the MAXVALUE limit by using the ALTER SEQUENCE statement.
- E. When a database instance shuts down abnormally, the sequence numbers that have been cached but not used would be available once again when the database instance

**Answer:** CD  
**Section:** (none)

**Explanation/Reference:**  
**QUESTION 46**

Which statements are correct regarding indexes? (Choose all that apply.)

- A. When a table is dropped, the corresponding indexes are automatically dropped.
- B. A FOREIGN KEY constraint on a column in a table automatically creates a nonunique index.
- C. A nondeferrable PRIMARY KEY or UNIQUE KEY constraint in a table automatically creates a unique index.
- D. For each data manipulation language (DML) operation performed, the corresponding indexes are automatically updated.

**Answer:** ACD  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 47**

View the Exhibit and examine the structure of ORD and ORD\_ITEMS tables.

The ORD\_NO column is PRIMARY KEY in the ORD table and the ORD\_NO and ITEM\_NO columns are composite PRIMARY KEY in the ORD\_ITEMS table.

Which two CREATE INDEX statements are valid? (Choose two.)

ORD

Name	Null?	Type
ORD_NO	NOT NULL	NUMBER(2)
ORD_DATE		DATE
CUST_ID		NUMBER(4)

ORD ITEMS

Name	Null?	Type
ORD_NO	NOT NULL	NUMBER(2)
ITEM_NO	NOT NULL	NUMBER(3)
QTY		NUMBER(8,2)

- A. CREATE INDEX ord\_idx1  
ON ord(ord\_no);
- B. CREATE INDEX ord\_idx2  
ON ord\_items(ord\_no);
- C. CREATE INDEX ord\_idx3  
ON ord\_items(item\_no);
- D. CREATE INDEX ord\_idx4  
ON ord,ord\_items(ord\_no, ord\_date,qty);

**Answer:** BC  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 48**

Which two statements are true regarding indexes? (Choose two.)

- A. They can be created on tables and clusters.
- B. They can be created on tables and simple views.
- C. You can create only one index by using the same columns.
- D. You can create more than one index by using the same columns if you specify distinctly different combinations of the columns.

**Answer:** AD  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 49**

The ORDERS table belongs to the user OE. OE has granted the SELECT privilege on the ORDERS table to the user HR.

Which statement would create a synonym ORD so that HR can execute the following query successfully?

SELECT \* FROM ord;

- A. CREATE SYNONYM ord FOR orders; This command is issued by OE.

- B. CREATE PUBLIC SYNONYM ord FOR orders; This command is issued by OE.
- C. CREATE SYNONYM ord FOR oe.orders; This command is issued by the database administrator.
- D. CREATE PUBLIC SYNONYM ord FOR oe.orders; This command is issued by the database administrator.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 50**

SLS is a private synonym for the SH.SALES table.

The user SH issues the following command:

```
DROP SYNONYM sls;
```

Which statement is true regarding the above SQL statement?

- A. Only the synonym would be dropped.
- B. The synonym would be dropped and the corresponding table would become invalid.
- C. The synonym would be dropped and the packages referring to the synonym would be dropped.
- D. The synonym would be dropped and any PUBLIC synonym with the same name becomes invalid.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 51**

Which statement is true regarding synonyms?

- A. Synonyms can be created only for a table.
- B. Synonyms are used to reference only those tables that are owned by another user.
- C. A public synonym and a private synonym can exist with the same name for the same table.
- D. The DROP SYNONYM statement removes the synonym, and the table on which the synonym has been created becomes invalid.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 52**

View the Exhibit and examine the structure of the PRODUCTS table.

Using the PRODUCTS table, you issue the following query to generate the names, current list price, and discounted list price for all those products whose list price falls below \$10 after a discount of 25% is applied on it.

```
SQL>SELECT prod_name, prod_list_price,  
prod_list_price - (prod_list_price * .25) "DISCOUNTED_PRICE"  
FROM products  
WHERE discounted_price < 10;
```

The query generates an error.

What is the reason for the error?

Table PRODUCTS		
Name	Null?	Type
<b>PROD_ID</b>	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)

- A. The parenthesis should be added to enclose the entire expression.
- B. The double quotation marks should be removed from the column alias.
- C. The column alias should be replaced with the expression in the WHERE clause.
- D. The column alias should be put in uppercase and enclosed with in double quotation marks in the WHERE clause.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 53

View the Exhibit and examine the data in the PROMOTIONS table.

PROMO\_BEGIN\_DATE is stored in the default date format, dd-mon-rr.

You need to produce a report that provides the name, cost, and start date of all promos in the POST category that were launched before January 1, 2000.

Which SQL statement would you use?

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

- A. SELECT promo\_name, promo\_cost, promo\_begin\_date  
FROM promotions  
WHERE promo\_category = 'post' AND promo\_begin\_date < '01-01-00';
- B. SELECT promo\_name, promo\_cost, promo\_begin\_date  
FROM promotions  
WHERE promo\_cost LIKE 'post%' AND promo\_begin\_date < '01-01-2000';

- C. SELECT promo\_name, promo\_cost, promo\_begin\_date  
FROM promotions  
WHERE promo\_category LIKE 'P%' AND promo\_begin\_date < '1-JANUARY-00';
- D. SELECT promo\_name, promo\_cost, promo\_begin\_date  
FROM promotions

WHERE promo\_category LIKE '%post%' AND promo\_begin\_date < '1-JAN-00';

**Answer:** D  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 54**

View the Exhibit and examine the structure of the CUSTOMERS table.

Evaluate the query statement:

```
SQL> SELECT cust_last_name, cust_city, cust_credit_limit  
FROM customers
```

```
WHERE cust_last_name BETWEEN 'A' AND 'C' AND cust_credit_limit BETWEEN  
1000 AND 3000;
```

What would be the outcome of the above statement?

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. It executes successfully.
- B. It produces an error because the condition on CUST\_LAST\_NAME is invalid.
- C. It executes successfully only if the CUST\_CREDIT\_LIMIT column does not contain any null values.
- D. It produces an error because the AND operator cannot be used to combine multiple BETWEEN clauses.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 55**

Evaluate the following two queries:

```
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;
```

Which statement is true regarding the above two queries?

- A. Performance would improve in query 2.
- B. Performance would degrade in query 2.
- C. There would be no change in performance.
- D. Performance would improve in query 2 only if there are null values in the CUST\_CREDIT\_LIMIT column.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 56**

View the Exhibit and examine the structure of the PROMOTIONS table.

Using the PROMOTIONS table, you need to find out the names and cost of all the promos done on 'TV' and 'internet' that ended in the time interval 15th March '00 to 15th October '00.

Which two queries would give the required result? (Choose two.)

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\_cost  
FROM promotions  
WHERE promo\_category IN ('TV', 'internet') AND  
promo\_end\_date BETWEEN '15-MAR-00' AND '15-OCT-00';
- B. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE promo\_category = 'TV' OR promo\_category ='internet' AND promo\_end\_date >='15-MAR-00' OR promo\_end\_date <='15-OCT-00';
- C. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE (promo\_category BETWEEN 'TV' AND 'internet') AND  
(promo\_end\_date IN ('15-MAR-00','15-OCT-00'));
- D. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE (promo\_category = 'TV' OR promo\_category ='internet') AND (promo\_end\_date >='15-MAR-00' AND promo\_end\_date <='15-OCT-00')

**Answer:** AD

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 57

Customers whose tax amount is null should not be considered.

Which statement accomplishes all the required tasks?

- A. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers  
WHERE cust\_income\_level IS NOT NULL AND  
tax\_amount IS NOT NULL;
- B. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers  
WHERE cust\_income\_level IS NOT NULL AND  
cust\_credit\_limit IS NOT NULL;
- C. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers  
WHERE cust\_income\_level <> NULL AND  
tax\_amount <> NULL;
- D. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT FROM customers  
WHERE (cust\_income\_level,tax\_amount) IS NOT NULL;

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 58

The PART\_CODE column in the SPARES table contains the following list of values:

PART\_CODE

A%\_WQ123

A%BWQ123

AB\_WQ123

Evaluate the following query:  
SQL> SELECT part\_code

FROM spares

WHERE part\_code LIKE '%\%\_WQ12%' ESCAPE '\';

Which statement is true regarding the outcome of the above query?

- A. It produces an error.
- B. It displays all values.
- C. It displays only the values A%\_WQ123 and AB\_WQ123 .
- D. It displays only the values A%\_WQ123 and A%BWQ123 .
- E. It displays only the values A%BWQ123 and AB\_WQ123.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 59**

View the Exhibit and examine the data in the PRODUCTS table.

You need to display product names from the PRODUCTS table that belong to the 'Software/Other' category with minimum prices as either \$2000 or \$4000 and no unit of measure.

You issue the following query:

SQL>SELECT prod\_name, prod\_category, prod\_min\_price

FROM products

WHERE prod\_category LIKE '%Other%' AND (prod\_min\_price = 2000 OR  
prod\_min\_price = 4000) AND prod\_unit\_of\_measure <> ";

Which statement is true regarding the above query?

## PRODUCTS

PROD_ID	PROD_NAME	PROD_CATEGORY	PROD_MIN_PRICE	PROD_UNIT_OF_MEASURE
101	Envoy 256MB - 40GB	Hardware	6000	Nos.
102	Y Box	Electronics	9000	
103	DVD-R Disc, 4.7 GB	Software/Other	2000	Nos.
104	Documentation Set - Spanish	Software/Other	4000	

- A. It executes successfully but returns no result.
- B. It executes successfully and returns the required result.
- C. It generates an error because the condition specified for PROD\_UNIT\_OF\_MEASURE is not valid.
- D. It generates an error because the condition specified for the PROD\_CATEGORY column is not valid.

**Answer:** A

**Section:** (none)

**Explanation/Reference:****QUESTION 60**

View the Exhibit and examine the structure of CUSTOMERS table.

Evaluate the following query:

```
SQL>SELECT cust_id, cust_city
  FROM customers
 WHERE cust_first_name NOT LIKE 'A_%g_%' AND
       cust_credit_limit BETWEEN 5000 AND 15000 AND
       cust_credit_limit NOT IN (7000, 11000) AND
       cust_city NOT BETWEEN 'A' AND 'B';
```

Which statement is true regarding the above query?

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. It executes successfully.
- B. It produces an error because the condition on the CUST\_CITY column is not valid.
- C. It produces an error because the condition on the CUST\_FIRST\_NAME column is not valid.
- D. It produces an error because conditions on the CUST\_CREDIT\_LIMIT column are not valid.

**Answer:** A

**Section:** (none)

**Explanation/Reference:****QUESTION 61**

The promo should have ended after 1st January 2001.

Which WHERE clause would give the required result?

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. WHERE promo\_name NOT LIKE 'T%' OR promo\_name NOT LIKE 'N%' AND promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01'
- B. WHERE (promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%')OR promo\_cost > 20000 OR promo\_end\_date > '1-JAN-01'
- C. WHERE promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%' AND promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01'
- D. WHERE (promo\_name NOT LIKE '%T%' OR promo\_name NOT LIKE '%N%') AND(promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01')

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 62**

View the Exhibit and examine the structure of the CUSTOMERS table.

You want to generate a report showing the last names and credit limits of all customers whose last names start with A, B, or C, and credit limit is below 10,000.

Evaluate the following two queries:

```
SQL> SELECT cust_last_name, cust_credit_limit FROM customers
      WHERE (UPPER(cust_last_name) LIKE 'A%' OR
             UPPER(cust_last_name) LIKE 'B%' OR UPPER(cust_last_name) LIKE 'C%')
             AND cust_credit_limit < 10000;
```

```
SQL>SELECT cust_last_name, cust_credit_limit FROM customers
      WHERE UPPER(cust_last_name) BETWEEN 'A' AND 'C'
      AND cust_credit_limit < 10000;
```

Which statement is true regarding the execution of the above queries?

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. Only the first query gives the correct result.
- B. Only the second query gives the correct result.
- C. Both execute successfully and give the same result.
- D. Both execute successfully but do not give the required result.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 63**

View the Exhibit and examine the structure of the PRODUCTS table.

You want to display only those product names with their list prices where the list price is at least double the minimum price. The report should start with the product name having the maximum list price satisfying this condition.

Evaluate the following SQL statement:

```

SQL>SELECT prod_name,prod_list_price
  FROM products
 WHERE prod_list_price >= 2 * prod_min_price

```

Which ORDER BY clauses can be added to the above SQL statement to get the correct output?

(Choose all that apply.)

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)

- A. ORDER BY prod\_list\_price DESC, prod\_name;
- B. ORDER BY (2\*prod\_min\_price)DESC, prod\_name;
- C. ORDER BY prod\_name, (2\*prod\_min\_price)DESC;
- D. ORDER BY prod\_name DESC, prod\_list\_price DESC;
- E. ORDER BY prod\_list\_price DESC, prod\_name DESC;

**Answer:** AE

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 64

View the Exhibit and examine the data in the PROMO\_CATEGORY and PROMO\_COST columns of the PROMOTIONS table.

Evaluate the following two queries:

```

SQL>SELECT DISTINCT promo_category TO_CHAR(promo_cost)"code"

```

```

  FROM promotions

```

```

  ORDER BY code;

```

```

SQL>SELECT DISTINCT promo_category promo_cost "code"

```

```

  FROM promotions

```

```

  ORDER BY 1;

```

Which statement is true regarding the execution of the above queries?

## PROMOTIONS

PROMO_CATEGORY	PROMO_COST
radio	97200
newspaper	97800
TV	97600
post	98000
internet	98200
TV	98300
internet	98700
newspaper	98500
magazine	98400
radio	99100
post	99000

- A. Only the first query executes successfully.
- B. Only the second query executes successfully.
- C. Both queries execute successfully but give different results.
- D. Both queries execute successfully and give the same result.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 65**

View the Exhibit and examine the structure of the CUSTOMERS table.

You have been asked to produce a report on the CUSTOMERS table showing the customers details sorted in descending order of the city and in the descending order of their income level in each city.

Which query would accomplish this task?

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 cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_city desc, cust\_income\_level DESC ;
- B. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_income\_level desc, cust\_city DESC;
- C. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY (cust\_city, cust\_income\_level) DESC;
- D. SELECT cust\_city, cust\_income\_level, cust\_last\_name

```
FROM customers
ORDER BY cust_city, cust_income_level DESC;
```

**Answer:** A  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 66**

View the Exhibit and examine the data in the COSTS table.

You need to generate a report that displays the IDs of all products in the COSTS table whose unit price is at least 25% more than the unit cost. The details should be displayed in the descending order of 25% of the unit cost.

You issue the following query:

```
SQL>SELECT prod_id
  FROM costs
 WHERE unit_price >= unit_cost * 1.25
 ORDER BY unit_cost * 0.25 DESC;
```

Which statement is true regarding the above query?

**COSTS**

PROD_ID	PROMO_ID	UNIT_COST	UNIT_PRICE
14	111	900	1129
15	333	875	1075
16	333	700	900
17	444	1000	1150

- A. It executes and produces the required result.
- B. It produces an error because an expression cannot be used in the ORDER BY clause.
- C. It produces an error because the DESC option cannot be used with an expression in the ORDER BY clause.
- D. It produces an error because the expression in the ORDER BY clause should also be specified in the SELECT clause.

**Answer:** A  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 67**

Which two statements are true regarding the ORDER BY clause? (Choose two.)

- A. It is executed first in the query execution.
- B. It must be the last clause in the SELECT statement.
- C. It cannot be used in a SELECT statement containing a HAVING clause.
- D. You cannot specify a column name followed by an expression in this clause.
- E. You can specify a combination of numeric positions and column names in this clause.

**Answer:** BE  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 68**

Which statement is true regarding the default behavior of the ORDER BY clause?

- A. In a character sort, the values are case-sensitive.

- B. NULL values are not considered at all by the sort operation.
- C. Only those columns that are specified in the SELECT list can be used in the ORDER BY clause.
- D. Numeric values are displayed from the maximum to the minimum value if they have decimal positions.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 69**

You need to generate a list of all customer last names with their credit limits from the CUSTOMERS table. Those customers who do not have a credit limit should appear last in the list.

Which two queries would achieve the required result? (Choose two.)

- A. 

```
SELECT cust_last_name, cust_credit_limit
      FROM customers
      ORDER BY cust_credit_limit DESC ;
```
- B. 

```
SELECT cust_last_name, cust_credit_limit
      FROM customers
      ORDER BY cust_credit_limit;
```
- C. 

```
SELECT cust_last_name, cust_credit_limit
      FROM customers
      ORDER BY cust_credit_limit NULLS LAST;
```
- D. 

```
SELECT cust_last_name, cust_credit_limit
      FROM customers
      ORDER BY cust_last_name, cust_credit_limit NULLS LAST;
```

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 70**

View the Exhibit and examine the structure of the PRODUCTS table.

You want to display only those product names with their list prices where the list price is at least double the minimum price. The report should start with the product name having the maximum list price satisfying this condition.

Evaluate the following SQL statement:

```
SQL>SELECT prod_name,prod_list_price
```

```
FROM products
```

```
WHERE prod_list_price >= 2 * prod_min_price
```

Which ORDER BY clauses can be added to the above SQL statement to get the correct output?

(Choose all that apply.)

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)

- A. ORDER BY prod\_list\_price DESC, prod\_name;

- B. ORDER BY (2\*prod\_min\_price)DESC, prod\_name;
- C. ORDER BY prod\_name, (2\*prod\_min\_price)DESC;
- D. ORDER BY prod\_name DESC, prod\_list\_price DESC;
- E. ORDER BY prod\_list\_price DESC, prod\_name DESC;

**Answer:** AE

**Section:** (none)

**Explanation/Reference:**

**QUESTION 71**

Which arithmetic operations can be performed on a column by using a SQL function that is built into

Oracle database ? (Choose three .)

- A. addition
- B. subtraction
- C. raising to a power
- D. finding the quotient
- E. finding the lowest value

**Answer:** ACE

**Section:** (none)

**Explanation/Reference:**

**QUESTION 72**

Which tasks can be performed using SQL functions built into Oracle Database ? (Choose three.)

- A. displaying a date in a nondefault format
- B. finding the number of characters in an expression
- C. substituting a character string in a text expression with a specified string
- D. combining more than two columns or expressions into a single column in the output

**Answer:** ABC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 73**

Which tasks can be performed using SQL functions that are built into Oracle database ? (Choose

three .)

- A. finding the remainder of a division
- B. adding a number to a date for a resultant date value
- C. comparing two expressions to check whether they are equal
- D. checking whether a specified character exists in a given string
- E. removing trailing, leading, and embedded characters from a character string

**Answer:** ACD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 74**

Which statements are true regarding single row functions? (Choose all that apply.)

- A. MOD : returns the quotient of a division
- B. TRUNC : can be used with NUMBER and DATE values
- C. CONCAT : can be used to combine any number of values
- D. SYSDATE : returns the database server current date and time
- E. INSTR : can be used to find only the first occurrence of a character in a string
- F. TRIM : can be used to remove all the occurrences of a character from a string

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 75**

The following data exists in the PRODUCTS table:

PROD_ID	PROD_LIST_PRICE
123456	152525.99

123456 152525.99

You issue the following query:

```
SQL> SELECT RPAD(( ROUND(prod_list_price)), 10,'*')
```

```
FROM products
```

```
WHERE prod_id = 123456;
```

What would be the outcome?

- A. 152526 \*\*\*\*
- B. \*\*152525.99
- C. 152525\*\* \*\*
- D. an error message

**Answer:** A

**Section:** (none)

**Explanation/Reference:****QUESTION 76**

You need to display the first names of all customers from the CUSTOMERS table that contain the character 'e' and have the character 'a' in the second last position.

Which query would give the required output?

- A. 

```
SELECT cust_first_name
      FROM customers
      WHERE INSTR(cust_first_name, 'e')<>0 AND
            SUBSTR(cust_first_name, -2, 1)='a';
```
- B. 

```
SELECT cust_first_name
      FROM customers
      WHERE INSTR(cust_first_name, 'e')<>" AND
            SUBSTR(cust_first_name, -2, 1)='a';
```
- C. 

```
SELECT cust_first_name
      FROM customers
      WHERE INSTR(cust_first_name, 'e')IS NOT NULL AND
            SUBSTR(cust_first_name, 1,-2)='a';
```
- D. 

```
SELECT cust_first_name
      FROM customers
      WHERE INSTR(cust_first_name, 'e')<>0 AND
            SUBSTR(cust_first_name, LENGTH(cust_first_name),-2)='a';
```

**Answer:** A

**Section:** (none)

**Explanation/Reference:****QUESTION 77**

In the CUSTOMERS table, the CUST\_CITY column contains the value 'Paris' for the CUST\_FIRST\_NAME 'ABIGAIL'.

Evaluate the following query:

```
SQL> SELECT INITCAP(cust_first_name ''
                  UPPER(SUBSTR(cust_city,-LENGTH(cust_city),2)))
```

```
FROM customers  
WHERE cust_first_name = 'ABIGAIL';
```

What would be the outcome?

- A. Abigail PA
- B. Abigail Pa
- C. Abigail IS
- D. an error message

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 78**

Evaluate the following query:

```
SQL> SELECT TRUNC(ROUND(156.00,-1),-1)  
FROM DUAL;
```

What would be the outcome?

- A. 16
- B. 100
- C. 160
- D. 200
- E. 150

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 79**

View the Exhibit and examine the structure of the CUSTOMERS table.

In the CUSTOMERS table, the CUST\_LAST\_NAME column contains the values 'Anderson' and 'Ausson'.

You issue the following query:

```
SQL> SELECT LOWER(REPLACE(TRIM('son' FROM cust_last_name),'An','O'))  
FROM CUSTOMERS
```

WHERE LOWER(cust\_last\_name) LIKE 'a%n';

What would be the outcome?

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. 'Oder' and 'Aus'

- B. a n error because the TRIM function specified is not valid
- C. a n error because the LOWER function specified is not valid
- D. a n error because the REPLACE function specified is not valid

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 80**

Which two statements are true regarding working with dates? (Choose two.)

- A. The default internal storage of dates is in the numeric format.
- B. The default internal storage of dates is in the character format.
- C. The RR date format automatically calculates the century from the SYSDATE function and does not allow the user to enter the century.
- D. The RR date format automatically calculates the century from the SYSDATE function but allows the user to enter the century if required.

**Answer:** AD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 81**

You are currently located in Singapore and have connected to a remote database in Chicago.

You issue the following command:

```
SQL> SELECT ROUND(SYSDATE-promo_begin_date,0)
```

```
FROM promotions
```

```
WHERE (SYSDATE-promo_begin_date)/365 > 2;
```

PROMOTIONS is the public synonym for the public database link for the PROMOTIONS table.

What is the outcome?

- A. a n error because the ROUND function specified is invalid
- B. a n error because the WHERE condition specified is invalid
- C. n umber of days since the promo started based on the current Chicago date and time
- D. number of days since the promo started based on the current Singapore date and time

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 82**

Examine the data in the CUST\_NAME column of the CUSTOMERS table.

CUST\_NAME

Renske Ladwig

Jason Mallin

Samuel McCain

Allan MCEwen

Irene Mikkilineni

Julia Nayer

You need to display customers' second names where the second name starts with "Mc" or "MC."

Which query gives the required output?

- A. 

```
SELECT SUBSTR(cust_name, INSTR(cust_name,' ')+1)
      FROM customers
```

- WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1)) = 'Mc';
- B. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1)  
FROM customers  
WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1)) LIKE 'Mc%';
- C. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1)  
FROM customers  
WHERE SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1) LIKE INITCAP('MC%');
- D. SELECT SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1)  
FROM customers  
WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1)) = INITCAP('MC%');

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 83**

Examine the data in the CUST\_NAME column of the CUSTOMERS table.

CUST\_NAME

Lex De Haan

Renske Ladwig

Jose Manuel Urman

Jason Mallin

You want to extract only those customer names that have three names and display the \* symbol in place of the first name as follows:

CUST NAME

\*\*\* De Haan

\*\*\*\* Manuel Urman

Which two queries give the required output? (Choose two.)

- A. SELECT LPAD(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1), LENGTH(cust\_name) - INSTR(cust\_name, ' ') - 1) "CUST NAME" FROM customers WHERE INSTR(cust\_name, ' ', 1, 2) <> 0;
- B. SELECT LPAD(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1), LENGTH(cust\_name) - INSTR(cust\_name, ' ') - 1) "CUST NAME" FROM customers WHERE INSTR(cust\_name, ' ', -1, 2) <> 0;
- C. SELECT LPAD(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1), LENGTH(cust\_name) - INSTR(cust\_name, ' ') - 1) "CUST NAME" FROM customers WHERE INSTR(cust\_name, ' ', -1, -2) <> 0;
- D. SELECT LPAD(SUBSTR(cust\_name, INSTR(cust\_name, ' ') + 1), LENGTH(cust\_name) - INSTR(cust\_name, ' ') - 1) "CUST NAME" FROM customers WHERE INSTR(cust\_name, ' ', 1, 2) <> 0 ;

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

**QUESTION 84**

View the Exhibit and examine the structure of the EMPLOYEES table.

Examine the data in the ENAME and HIREDATE columns of the EMPLOYEES table:

ENAME HIREDATE

SMITH 17-DEC-80

ALLEN 20-FEB-81

WARD 22-FEB-81

You want to generate a list of user IDs as follows:

USERID  
Smi17DEC80  
All20FEB81  
War22FEB81

You issue the following query:

```
SQL>SELECT CONCAT(SUBSTR(INITCAP(ename),1,3), REPLACE(hiredate,'-')) "USERID"  
FROM employees;
```

What is the outcome?

EMPLOYEES		
Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

- A. It executes successfully and gives the correct output.
- B. It executes successfully but does not give the correct output.
- C. It generates an error because the REPLACE function is not valid.
- D. It generates an error because the SUBSTR function cannot be nested in the CONCAT function.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 85**

View the Exhibit and examine the structure and data in the INVOICE table.

Which statements are true regarding data type conversion in expressions used in queries? (Choose all that apply.)

- A. inv\_amt ='0255982' : requires explicit conversion
- B. inv\_date > '01-02-2008' : uses implicit conversion
- C. CONCAT(inv\_amt,inv\_date) : requires explicit conversion
- D. inv\_date = '15-february-2008' : uses implicit conversion
- E. inv\_no BETWEEN '101' AND '110' : uses implicit conversion

**Answer:** DE

**Section:** (none)

**Explanation/Reference:**

**QUESTION 86**

Examine the structure and data of the CUST\_TRANS table:

CUST\_TRANS

Name	Null	Type
CUSTNO	NOT NULL	CHAR(2)

TRANSDATE DATE

TRANSAMT NUMBER(6,2)

CUSTNO TRANSDATE TRANSAMT

11 01-JAN-07 1000

22 01-FEB-07 2000

33 01-MAR-07 3000

Dates are stored in the default date format dd-mon-rr in the CUST\_TRANS table.

Which SQL statements would execute successfully? (Choose three .)

- A. SELECT transdate + '10' FROM cust\_trans;
- B. SELECT \* FROM cust\_trans WHERE transdate = '01-01-07';
- C. SELECT transamt FROM cust\_trans WHERE custno > '11';
- D. SELECT \* FROM cust\_trans WHERE transdate='01-JANUARY-07';
- E. SELECT custno + 'A' FROM cust\_trans WHERE transamt > 2000;

**Answer:** ACD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 87**

You want to display the date for the first Mon day of the next month and issue the following command:

```
SQL>SELECT TO_CHAR(NEXT_DAY(LAST_DAY(SYSDATE),'MON'),  
'dd "is the first Monday for" fmmonth rrrr')  
FROM DUAL;
```

What is the outcome?

- A. It executes successfully and returns the correct result.
- B. It executes successfully but does not return the correct result.
- C. It generates an error because TO\_CHAR should be replaced with TO\_DATE.
- D. It generates an error because rrrr should be replaced by rr in the format string.
- E. It generates an error because fm and double quotation marks should not be used in the format string.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 88**

You need to calculate the number of days from 1st January 2007 till date.

Dates are stored in the default format of dd-mon-rr.

Which SQL statements would give the required output? (Choose two .)

- A. SELECT SYSDATE - '01-JAN-2007' FROM DUAL;
- B. SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;
- C. SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL;
- D. SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;
- E. SELECT TO\_DATE(SYSDATE, 'DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL;

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 89**

You need to display the date 11-oct-2007 in words as 'Eleventh of October, Two Thousand Seven'.

Which SQL statement would give the required result?

- A. SELECT TO\_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year') FROM DUAL;
- B. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdspth of month, year') FROM DUAL;
- C. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdthsp "of" Month, Year') FROM DUAL;
- D. SELECT TO\_DATE(TO\_CHAR('11-oct-2007','fmDdspth "of" Month, Year')) FROM DUAL;

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 90**

Examine the structure and data in the PRICE\_LIST table:

name Null Type

PROD\_ID NOT NULL NUMBER(3)

PROD\_PRICE VARCHAR2(10)

PROD\_ID PROD\_PRICE

100 \$234.55

101 \$6,509.75

102 \$1,234

You plan to give a discount of 25% on the product price and need to display the discount amount in the same format as the PROD\_PRICE.

Which SQL statement would give the required result?

- A. SELECT TO\_CHAR(prod\_price\* .25,'\$99,999.99')  
FROM PRICE\_LIST;
- B. SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\* .25,'\$99,999.00')  
FROM PRICE\_LIST;
- C. SELECT TO\_CHAR(TO\_NUMBER(prod\_price,'\$99,999.99')\* .25,'\$99,999.00') FROM PRICE\_LIST;
- D. SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'\$99,999.99')\* .25,'\$99,999.00') FROM PRICE\_LIST;

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 91**

View the Exhibit and examine the structure of the PROMOTIONS table.

Which two SQL statements would execute successfully? (Choose two.)

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. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE TO\_CHAR(promo\_end\_date, 'yyyy') > '2000';
- B. SELECT promo\_begin\_date  
FROM promotions  
WHERE TO\_CHAR(promo\_begin\_date,'mon dd yy')="jul 01 98";
- C. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8));
- D. SELECT TO\_CHAR(promo\_begin\_date,'dd/month')  
FROM promotions  
WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'));

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 92

View the Exhibit and examine the data in the PROMO\_NAME and PROMO\_END\_DATE columns of the PROMOTIONS table, and the required output format.

Which two queries give the correct result? (Choose two.)

- A. SELECT promo\_name, TO\_CHAR(promo\_end\_date,'Day') ','  
TO\_CHAR(promo\_end\_date,'Month') ''  
TO\_CHAR(promo\_end\_date,'DD, YYYY') AS last\_day  
FROM promotions;
- B. SELECT promo\_name,TO\_CHAR (promo\_end\_date,'fxDay') ','  
TO\_CHAR(promo\_end\_date,'fxMonth') ''  
TO\_CHAR(promo\_end\_date,'fxDD, YYYY') AS last\_day  
FROM promotions;
- C. SELECT promo\_name, TRIM(TO\_CHAR(promo\_end\_date,'Day')) ',' TRIM(TO\_CHAR(promo\_end\_date,'Month')) ''  
TRIM(TO\_CHAR(promo\_end\_date,'DD, YYYY')) AS last\_day  
FROM promotions;
- D. SELECT promo\_name,TO\_CHAR(promo\_end\_date,'fmDay'),' '  
TO\_CHAR(promo\_end\_date,'fmMonth') ''  
TO\_CHAR(promo\_end\_date,'fmDD, YYYY') AS last\_day  
FROM promotions;

**Answer:** CD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 93**

View the Exhibit and examine the structure of the CUSTOMERS table.

Using the CUSTOMERS table, you need to generate a report that shows an increase in the credit limit by

15% for all customers. Customers whose credit limit has not been entered should have the message "Not Available" displayed.

Which SQL statement would produce the required result?

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. SELECT NVL(cust\_credit\_limit,'Not Available')\*.15 "NEW CREDIT" FROM customers;
- B. SELECT NVL(cust\_credit\_limit\*.15,'Not Available') "NEW CREDIT" FROM customers;
- C. SELECT TO\_CHAR(NVL(cust\_credit\_limit\*.15,'Not Available')) "NEW CREDIT" FROM customers;
- D. SELECT NVL(TO\_CHAR(cust\_credit\_limit\*.15),'Not Available') "NEW CREDIT" FROM customers;

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 94**

Examine the structure of the PROGRAMS table:

name Null Type

PROG\_ID NOT NULL NUMBER(3)

PROG\_COST NUMBER(8,2)

START\_DATE NOT NULL DATE

END\_DATE DATE

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT NVL(ADD\_MONTHS(END\_DATE,1),SYSDATE) FROM programs;
- B. SELECT TO\_DATE(NVL(SYSDATE-END\_DATE,SYSDATE)) FROM programs;
- C. SELECT NVL(MONTHS\_BETWEEN(start\_date,end\_date),'Ongoing') FROM programs;
- D. SELECT NVL(TO\_CHAR(MONTHS\_BETWEEN(start\_date,end\_date)),'Ongoing') FROM programs;

**Answer:** AD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 95**

The PRODUCTS table has the following structure:

```
name Null Type  
PROD_ID NOT NULL NUMBER(4)  
PROD_NAME VARCHAR2(25)  
PROD_EXPIRY_DATE DATE
```

Evaluate the following two SQL statements:

```
SQL>SELECT prod_id, NVL2(prod_expiry_date, prod_expiry_date + 15,")  
FROM products;  
  
SQL>SELECT prod_id, NVL(prod_expiry_date, prod_expiry_date + 15)  
FROM products;
```

Which statement is true regarding the outcome?

- A. Both the statements execute and give different results.
- B. Both the statements execute and give the same result.
- C. Only the first SQL statement executes successfully.
- D. Only the second SQL statement executes successfully.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 96**

Examine the structure of the INVOICE table.

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

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT inv\_no,NVL2(inv\_date,'Pending','Incomplete')  
FROM invoice;
- B. SELECT inv\_no,NVL2(inv\_amt,inv\_date,'Not Available')  
FROM invoice;
- C. SELECT inv\_no,NVL2(inv\_date,sysdate-inv\_date,sysdate)  
FROM invoice;
- D. SELECT inv\_no,NVL2(inv\_amt,inv\_amt\*.25,'Not Available')  
FROM invoice;

**Answer:** AC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 97**

View the Exhibit and evaluate the structure and data in the CUST\_STATUS table.

You issue the following SQL statement:

```
SQL> SELECT custno, NVL2(NULLIF(amt_spent, credit_limit), 0, 1000)"BONUS"
```

FROM cust\_status;

Which statement is true regarding the execution of the above query?

- A. It produces an error because the AMT\_SPENT column contains a null value.
- B. It displays a bonus of 1000 for all customers whose AMT\_SPENT is less than CREDIT\_LIMIT.
- C. It displays a bonus of 1000 for all customers whose AMT\_SPENT equals CREDIT\_LIMIT, or AMT\_SPENT is null .
- D. It produces an error because the TO\_NUMBER function must be used to convert the result of the NULLIF function before it can be used by the NVL2 function.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 98**

Which statement is true regarding the COALESCE function?

- A. It can have a maximum of five expressions in a list.
- B. It returns the highest NOT NULL value in the list for all rows.
- C. It requires that all expressions in the list must be of the same data type.
- D. It requires that at least one of the expressions in the list must have a NOT NULL value.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 99**

View the Exhibit and examine the structure of the PROMOTIONS table.

Using the PROMOTIONS table, you need to find out the average cost for all promos in the ranges

\$0-2000 and \$2000-5000 in category A

You issue the following SQL statement:

```
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;  
What would be the outcome?
```

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. It executes successfully and gives the required result.
- B. It generates an error because NULL cannot be specified as a return value.
- C. It generates an error because CASE cannot be used with group functions.
- D. It generates an error because multiple conditions cannot be specified for the WHEN clause.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 100

View the Exhibit and examine the structure of the PROMOTIONS table.

Which SQL statements are valid? (Choose all that apply.)

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\_id, DECODE(NVL(promo\_cost,0), promo\_cost, promo\_cost \* 0.25, 100) "Discount"  
FROM promotions;
- B. SELECT promo\_id, DECODE(promo\_cost, 10000, DECODE(promo\_category, 'G1', promo\_cost \*.25, NULL), NULL) "Catcost"  
FROM promotions;
- C. SELECT promo\_id, DECODE(NULLIF(promo\_cost, 10000), NULL, promo\_cost\*.25, 'N/A') "Catcost"  
FROM promotions;
- D. SELECT promo\_id, DECODE(promo\_cost, >10000, 'High', <10000, 'Low') "Range"  
FROM promotions;

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

**QUESTION 101**

Examine the data in the PROMO\_BEGIN\_DATE column of the PROMOTIONS table:

PROMO\_BEGIN\_DATE

04-jan-00

10-jan-00

15-dec-99

18-oct-98

22-aug-99

You want to display the number of promotions started in 1999 and 2000.

Which query gives the correct output?

- A. SELECT SUM(DECODE(SUBSTR(promo\_begin\_date,8),'00',1,0)) "2000", SUM(DECODE(SUBSTR(promo\_begin\_date,8),'99',1,0)) "1999"  
FROM promotions;
- B. SELECT SUM(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '99' THEN 1 ELSE 0 END) "1999",SUM(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '00' THEN 1 ELSE 0 END) "2000"  
FROM promotions;
- C. SELECT COUNT(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '99' THEN 1 ELSE 0 END) "1999",COUNT(CASE TO\_CHAR(promo\_begin\_date,'yyyy') WHEN '00' THEN 1 ELSE 0 END) "2000"  
FROM promotions;
- D. SELECT COUNT(DECODE(SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8), '1999', 1, 0)) "1999", COUNT(DECODE(SUBSTR(TO\_CHAR(promo\_begin\_date,'yyyy'), 8), '2000', 1, 0)) "2000"  
FROM promotions;

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 102**

Examine the structure of the TRANSACTIONS table:

name Null Type

TRANS\_ID NOT NULL NUMBER(3)

CUST\_NAME VARCHAR2(30)

TRANS\_DATE TIMESTAMP TRANS\_AMT NUMBER(10,2)

You want to display the date, time, and transaction amount of transactions that were done before 12 noon.

The value zero should be displayed for transactions where the transaction amount has not been entered.

Which query gives the required result?

- A. SELECT TO\_CHAR(trans\_date,'dd-mon-yyyy hh24:mi:ss'),  
TO\_CHAR(trans\_amt,'\$99999999D99')  
FROM transactions  
WHERE TO\_NUMBER(TO\_DATE(trans\_date,'hh24')) < 12 AND COALESCE(trans\_amt,NULL)>>NULL;
- B. SELECT TO\_CHAR(trans\_date,'dd-mon-yyyy hh24:mi:ss'),  
NVL(TO\_CHAR(trans\_amt,'\$99999999D99'),0)  
FROM transactions  
WHERE TO\_CHAR(trans\_date,'hh24') < 12;
- C. SELECT TO\_CHAR(trans\_date,'dd-mon-yyyy hh24:mi:ss'),  
COALESCE(TO\_NUMBER(trans\_amt,'\$99999999.99'),0)  
FROM transactions  
WHERE TO\_DATE(trans\_date,'hh24') < 12;

D. SELECT TO\_DATE (trans\_date,'dd-mon-yyyy hh24:mi:ss'),  
NVL2(trans\_amt,TO\_NUMBER(trans\_amt,'\$99999999.99'), 0)  
FROM transactions  
WHERE TO\_DATE(trans\_date,'hh24') < 12;

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 103**

Examine the structure of the TRANSACTIONS table:

name Null Type  
TRANS\_ID NOT NULL NUMBER(3)  
CUST\_NAME VARCHAR2(30)  
TRANS\_DATE DATE  
TRANS\_AMT NUMBER(10,2)

You want to display the transaction date and specify whether it is a weekday or weekend.

Evaluate the following two queries:

SQL>SELECT TRANS\_DATE,CASE  
WHEN TRIM(TO\_CHAR(trans\_date,'DAY')) IN ('SATURDAY','SUNDAY') THEN 'weekend'  
ELSE 'weekday'  
END "Day Type"  
FROM transactions;  
SQL>SELECT TRANS\_DATE, CASE  
WHEN TO\_CHAR(trans\_date,'DAY') BETWEEN 'MONDAY' AND 'FRIDAY' THEN 'weekday'  
ELSE 'weekend'  
END "Day Type"FROM transactions;

Which statement is true regarding the above queries?

- A. Both give wrong results.
- B. Both give the correct result.
- C. Only the first query gives the correct result.
- D. Only the second query gives the correct result.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 104**

Examine the structure of the PROMOS table:

name Null Type  
PROMO\_ID NOT NULL NUMBER(3)  
PROMO\_NAME VARCHAR2(30)

PROMO\_START\_DATE NOT NULL DATE

PROMO\_END\_DATE DATE

You want to generate a report showing promo names and their duration (number of days). If the PROMO\_END\_DATE has not been entered, the message 'ONGOING' should be displayed.

Which queries give the correct output? (Choose all that apply.)

- A. SELECT promo\_name, TO\_CHAR(NVL(promo\_end\_date - promo\_start\_date,'ONGOING')) FROM promos;
- B. SELECT promo\_name, COALESCE(TO\_CHAR(promo\_end\_date - promo\_start\_date),'ONGOING') FROM promos;
- C. SELECT promo\_name, NVL(TO\_CHAR(promo\_end\_date - promo\_start\_date),'ONGOING') FROM promos;
- D. SELECT promo\_name, DECODE(promo\_end\_date - promo\_start\_date,NULL,'ONGOING',promo\_end\_date - promo\_start\_date) FROM promos;
- E. SELECT promo\_name, decode(coalesce(promo\_end\_date,promo\_start\_date),null,'ONGOING', promo\_end\_date - promo\_start\_date) FROM promos;

**Answer:** BCD

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 105

Examine the structure of the PROMOS table:

name Null Type

PROMO\_ID NOT NULL NUMBER(3)

PROMO\_NAME VARCHAR2(30)

PROMO\_START\_DATE NOT NULL DATE

PROMO\_END\_DATE NOT NULL DATE

You want to display the list of promo names with the message 'Same Day' for promos that started and ended on the same day.

Which query gives the correct output?

- A. SELECT promo\_name, NVL(NULLIF(promo\_start\_date, promo\_end\_date), 'Same Day') FROM promos;
- B. SELECT promo\_name, NVL(TRUNC(promo\_end\_date - promo\_start\_date), 'Same Day') FROM promos;
- C. SELECT promo\_name, NVL2(TO\_CHAR(TRUNC(promo\_end\_date-promo\_start\_date)), NULL,'Same Day') FROM promos;
- D. SELECT promo\_name, DECODE((NULLIF(promo\_start\_date, promo\_end\_date)), NULL,'Same day') FROM promos;

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 106

Examine the data in the LIST\_PRICE and MIN\_PRICE columns of the PRODUCTS table:

LIST\_PRICE MIN\_PRICE

10000 8000

20000

30000 30000

Which two expressions give the same output? (Choose two.)

- A. NVL(NULLIF(list\_price, min\_price), 0)
- B. NVL(COALESCE(list\_price, min\_price), 0)
- C. NVL2(COALESCE(list\_price, min\_price), min\_price, 0)
- D. COALESCE(NVL2(list\_price, list\_price, min\_price), 0)

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 107**

View the Exhibit and examine the structure and data in the INVOICE table.

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT AVG(inv\_date )  
FROM invoice;
- B. SELECT MAX(inv\_date),MIN(cust\_id)  
FROM invoice;
- C. SELECT MAX(AVG(SYSDATE - inv\_date))  
FROM invoice;
- D. SELECT AVG( inv\_date - SYSDATE), AVG(inv\_amt)  
FROM invoice;

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 108**

Which two statements are true regarding the COUNT function? (Choose two.)

- A. The COUNT function can be used only for CHAR, VARCHAR2, and NUMBER data types.
- B. COUNT(\*) returns the number of rows including duplicate rows and rows containing NULL value in any of the columns.
- C. COUNT(cust\_id) returns the number of rows including rows with duplicate customer IDs and NULL value in the CUST\_ID column.
- D. COUNT(DISTINCT inv\_amt) returns the number of rows excluding rows containing duplicates and NULL values in the INV\_AMT column.
- E. A SELECT statement using the COUNT function with a DISTINCT keyword cannot have a WHERE clause.

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 109**

Examine the structure of the MARKS table:

name Null Type

STUDENT\_ID NOT NULL VARCHAR2(4)

STUDENT\_NAME VARCHAR2(25)

SUBJECT1 NUMBER(3)

SUBJECT2 NUMBER(3)

SUBJECT3 NUMBER(3)

Which two statements would execute successfully? (Choose two.)

- A. SELECT student\_name,subject1  
FROM marks  
WHERE subject1 > AVG(subject1);
- B. SELECT student\_name,SUM(subject1)  
FROM marks  
WHERE student\_name LIKE 'R%';
- C. SELECT SUM(subject1+subject2+subject3)  
FROM marks  
WHERE student\_name IS NULL;
- D. SELECT SUM(DISTINCT NVL(subject1,0)), MAX(subject1)  
FROM marks  
WHERE subject1 > subject2;

**Answer:** CD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 110**

View the Exhibit and examine the structure of the CUSTOMERS table.

Using the CUSTOMERS table, you need to generate a report that shows the average credit limit for customers in WASHINGTON and NEW YORK.

Which SQL statement would produce the required result?

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 cust\_city, AVG(cust\_credit\_limit)  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK')  
GROUP BY cust\_credit\_limit, cust\_city;
- B. SELECT cust\_city, AVG(cust\_credit\_limit)  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK')  
GROUP BY cust\_city,cust\_credit\_limit;
- C. SELECT cust\_city, AVG(cust\_credit\_limit)  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK')  
GROUP BY cust\_city;
- D. SELECT cust\_city, AVG(NVL(cust\_credit\_limit,0))  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK');

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

**QUESTION 111**

View the Exhibit and examine the structure of the CUSTOMERS table.

Which statement would display the highest credit limit available in each income level in each city in the CUSTOMERS table?

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 cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_city, cust\_income\_level, cust\_credit\_limit;
- B. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_city, cust\_income\_level;
- C. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_credit\_limit, cust\_income\_level, cust\_city ;
- D. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit) FROM customers GROUP BY cust\_city, cust\_income\_level, MAX(cust\_credit\_limit);

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 112**

View the Exhibit and examine the structure of the PROMOTIONS table.

Evaluate the following SQL statement:

```
SQL>SELECT promo_category, AVG(promo_cost) Avg_Cost, AVG(promo_cost)*.25 Avg_Overhead  
FROM promotions  
  
WHERE UPPER(promo_category) IN ('TV', 'INTERNET','POST')  
  
GROUP BY Avg_Cost  
  
ORDER BY Avg_Overhead;
```

The above query generates an error on execution.

Which clause in the above SQL statement causes the error?

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. WHERE
- B. SELECT
- C. GROUP BY
- D. ORDER BY

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 113

Examine the structure of the ORDERS table:

Name Null Type

ORDER\_ID NOT NULL NUMBER(12)

ORDER\_DATE NOT NULL TIMESTAMP(6)

CUSTOMER\_ID NOT NULL NUMBER(6)

ORDER\_STATUS NUMBER(2)

ORDER\_TOTAL NUMBER(8,2)

You want to find the total value of all the orders for each year and issue the following command:

```
SQL>SELECT TO_CHAR(order_date,'rr'), SUM(order_total)
```

FROM orders

```
GROUP BY TO_CHAR(order_date,'yyyy');
```

Which statement is true regarding the outcome?

- A. It executes successfully and gives the correct output.
- B. It gives an error because the TO\_CHAR function is not valid.
- C. It executes successfully but does not give the correct output.
- D. It gives an error because the data type conversion in the SELECT list does not match the data type conversion in the GROUP BY clause.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 114

View the Exhibit and examine the structure of the SALES table.

The following query is written to retrieve all those product ID s from the SALES table that have more than 55000 sold and have been ordered more than 10 times.

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

Which statement is true regarding this SQL statement?

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)

- A. It executes successfully and generates the required result.
- B. It produces an error because COUNT(\*) should be specified in the SELECT clause also.
- C. It produces an error because COUNT(\*) should be only in the HAVING clause and not in the WHERE clause.
- D. It executes successfully but produces no result because COUNT(prod\_id) should be used instead of COUNT(\*) .

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 115

View the Exhibit and examine the structure of the CUSTOMERS table.

Evaluate the following SQL statement:

```
SQL> SELECT cust_city, COUNT(cust_last_name)  
FROM customers  
  
WHERE cust_credit_limit > 1000  
  
GROUP BY cust_city  
  
HAVING AVG(cust_credit_limit) BETWEEN 5000 AND 6000;
```

Which statement is true regarding the outcome of the above query?

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. It executes successfully.
- B. It returns an error because the BETWEEN operator cannot be used in the HAVING clause.
- C. It returns an error because WHERE and HAVING clauses cannot be used in the same SELECT statement.
- D. It returns an error because WHERE and HAVING clauses cannot be used to apply conditions on the same column.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 116

You issue the following query:

```
SQL> SELECT AVG(MAX(qty))
  FROM ord_items
 GROUP BY item_no
 HAVING AVG(MAX(qty))>50;
```

Which statement is true regarding the outcome of this query?

- A. It executes successfully and gives the correct output.
- B. It gives an error because the HAVING clause is not valid.
- C. It executes successfully but does not give the correct output.
- D. It gives an error because the GROUP BY expression is not valid.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 117

Which statements are true regarding the WHERE and HAVING clauses in a SELECT statement?

(Choose all that apply.)

- A. The HAVING clause can be used with aggregate functions in subqueries.
- B. The WHERE clause can be used to exclude rows after dividing them into groups.
- C. The WHERE clause can be used to exclude rows before dividing them into groups.
- D. The aggregate functions and columns used in the HAVING clause must be specified in the SELECT list of the query.
- E. The WHERE and HAVING clauses can be used in the same statement only if they are applied to different columns in the table.

**Answer:** AC

**Section:** (none)

**Explanation/Reference:**

**QUESTION 118**

View the Exhibit and examine the structure of the PROMOTIONS table.

Examine the following two SQL statements:

Statement 1

```
SQL>SELECT promo_category,SUM(promo_cost)  
FROM promotions  
WHERE promo_end_date-promo_begin_date > 30  
GROUP BY promo_category;
```

Statement 2

```
SQL>SELECT promo_category,sum(promo_cost)  
FROM promotions  
GROUP BY promo_category  
HAVING MIN(promo_end_date-promo_begin_date)>30;
```

Which statement is true regarding the above two SQL statements?

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. statement 1 gives an error, statement 2 executes successfully
- B. statement 2 gives an error, statement 1 executes successfully
- C. statement 1 and statement 2 execute successfully and give the same output
- D. statement 1 and statement 2 execute successfully and give a different output

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 119**

Examine the data in the ORD\_ITEMS table:

ORD\_NO ITEM\_NO QTY

1 111 10

1 222 20

1 333 30

2 333 30

2 444 40

3 111 40

Evaluate the following query:

```
SQL>SELECT item_no, AVG(qty)
```

```
FROM ord_items
```

```
HAVING AVG(qty) > MIN(qty) * 2
```

```
GROUP BY item_no;
```

Which statement is true regarding the outcome of the above query?

- A. It gives an error because the HAVING clause should be specified after the GROUP BY clause.
- B. It gives an error because all the aggregate functions used in the HAVING clause must be specified in the SELECT list.
- C. It displays the item nos with their average quantity where the average quantity is more than double the minimum quantity of that item in the table.
- D. It displays the item nos with their average quantity where the average quantity is more than double the overall minimum quantity of all the items.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 120**

View the Exhibits and examine the structures of the PRODUCTS, SALES, and CUSTOMERS

tables.

You issue the following query:

```
SQL>SELECT p.prod_id,prod_name,prod_list_price,  
quantity_sold,cust_last_name  
FROM products p NATURAL JOIN sales s NATURAL JOIN customers c  
WHERE prod_id =148;
```

Which statement is true regarding the outcome of this query?

- A. It executes successfully.
- B. It produces an error because the NATURAL join can be used only with two tables.
- C. It produces an error because a column used in the NATURAL join cannot have a qualifier.
- D. It produces an error because all columns used in the NATURAL join should have a qualifier.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 121**

Which two statements are true regarding the USING clause in table joins? (Choose two .)

- A. It can be used to join a maximum of three tables.
- B. It can be used to restrict the number of columns used in a NATURAL join.
- C. It can be used to access data from tables through equijoins as well as nonequijoins.
- D. It can be used to join tables that have columns with the same name and compatible data types.

**Answer:** BD  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 122**

View the Exhibit for the structure of the STUDENT and FACULTY tables.

You need to display the faculty name followed by the number of students handled by the faculty at the base location.

Examine the following two SQL statements:

Statement 1

```
SQL>SELECT faculty_name,COUNT(student_id)  
FROM student JOIN faculty  
USING (faculty_id, location_id)  
GROUP BY faculty_name;
```

Statement 2

```
SQL>SELECT faculty_name,COUNT(student_id)  
FROM student NATURAL JOIN faculty  
GROUP BY faculty_name;
```

Which statement is true regarding the outcome?

- A. Only statement 1 executes successfully and gives the required result.
- B. Only statement 2 executes successfully and gives the required result.
- C. Both statements 1 and 2 execute successfully and give different results.
- D. Both statements 1 and 2 execute successfully and give the same required result.

**Answer:** D  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 123**

View the Exhibits and examine the structures of the PRODUCTS, SALES, and CUSTOMERS tables.

You need to generate a report that gives details of the customer's last name, name of the product, and the quantity sold for all customers in 'Tokyo' .

Which two queries give the required result? (Choose two.)

- A. 

```
SELECT c.cust_last_name,p.prod_name, s.quantity_sold  
FROM sales s JOIN products p  
USING(prod_id)  
JOIN customers c  
USING(cust_id)  
WHERE c.cust_city='Tokyo';
```
- B. 

```
SELECT c.cust_last_name, p.prod_name, s.quantity_sold  
FROM products p JOIN sales s JOIN customers c  
ON(p.prod_id=s.prod_id)  
ON(s.cust_id=c.cust_id)
```

- WHERE c.cust\_city='Tokyo';
- C. SELECT c.cust\_last\_name, p.prod\_name, s.quantity\_sold  
 FROM products p JOIN sales s  
 ON(p.prod\_id=s.prod\_id)  
 JOIN customers c  
 ON(s.cust\_id=c.cust\_id)  
 AND c.cust\_city='Tokyo';
- D. SELECT c.cust\_id,c.cust\_last\_name,p.prod\_id, p.prod\_name, s.quantity\_sold FROM products p JOIN sales s  
 USING(prod\_id)  
 JOIN customers c  
 USING(cust\_id)  
 WHERE c.cust\_city='Tokyo';

**Answer:** AC

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 124

View the Exhibit and examine the structure of the PROMOTIONS, SALES, and CUSTOMER tables.

You need to generate a report showing the promo name along with the customer name for all products that were sold during their promo campaign and before 30th October 2007.

You issue the following query:

```
SQL> SELECT promo_name,cust_name
  FROM promotions p JOIN sales s
  ON(time_id BETWEEN promo_begin_date AND promo_end_date)
  JOIN customer c
  ON (s.cust_id = c.cust_id) AND time_id < '30-oct-2007';
```

Which statement is true regarding the above query?

- A. It executes successfully and gives the required result.
- B. It executes successfully but does not give the required result.
- C. It produces an error because the join order of the tables is incorrect.
- D. It produces an error because equijoin and nonequijoin conditions cannot be used in the same SELECT statement.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 125

Examine the structure of the CUSTOMERS table:

name Null Type

CUSTNO NOT NULL NUMBER(3)

CUSTNAME NOT NULL VARCHAR2(25)

CUSTADDRESS VARCHAR2(35)

CUST\_CREDIT\_LIMIT NUMBER(5)

CUSTNO is the PRIMARY KEY in the table. You want to find out if any customers' details have been

entered more than once using different CUSTNO, by listing all the duplicate names.

Which two methods can you use to get the required result? (Choose two.)

- A. self-join
- B. subquery
- C. full outer-join with self-join
- D. left outer-join with self-join
- E. right outer-join with self-join

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

**QUESTION 126**

View the Exhibit and examine the data in the PROJ\_TASK\_DETAILS table.

The PROJ\_TASK\_DETAILS table stores information about tasks involved in a project and the relation between them.

The BASED\_ON column indicates dependencies between tasks. Some tasks do not depend on the completion of any other tasks.

You need to generate a report showing all task IDs, the corresponding task ID they are dependent on, and the name of the employee in charge of the task it depends on.

Which query would give the required result?

- A. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p JOIN proj\_task\_details d  
ON (p.based\_on = d.task\_id);
- B. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p LEFT OUTER JOIN proj\_task\_details d ON (p.based\_on = d.task\_id);
- C. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p FULL OUTER JOIN proj\_task\_details d ON (p.based\_on = d.task\_id);
- D. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p JOIN proj\_task\_details d  
ON (p.task\_id = d.task\_id);

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 127**

Examine the data in the CUSTOMERS table:

CUSTNO CUSTNAME CITY

1 KING SEATTLE

2 GREEN BOSTON

3 KOCHAR SEATTLE

4 SMITH NEW YORK

You want to list all cities that have more than one customer along with the customer details.  
Evaluate the following query:

```
SQL>SELECT c1.custname, c1.city  
FROM Customers c1 _____ Customers c2  
ON (c1.city=c2.city AND c1.custname<>c2.custname);
```

Which two JOIN options can be used in the blank in the above query to give the correct output? (Choose two.)

- A. JOIN
- B. NATURAL JOIN
- C. LEFT OUTER JOIN
- D. FULL OUTER JOIN
- E. RIGHT OUTER JOIN

**Answer:** AE

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 128**

View the Exhibits and examine the structures of the CUSTOMERS, SALES, and COUNTRIES

tables.

You need to generate a report that shows all country names, with corresponding customers (if any) and sales details (if any), for all customers.

Which FROM clause gives the required result?

- A. FROM sales JOIN customers USING (cust\_id)  
FULL OUTER JOIN countries USING (country\_id);
- B. FROM sales JOIN customers USING (cust\_id)  
RIGHT OUTER JOIN countries USING (country\_id);
- C. FROM customers LEFT OUTER JOIN sales USING (cust\_id)  
RIGHT OUTER JOIN countries USING (country\_id);
- D. FROM customers LEFT OUTER JOIN sales USING (cust\_id)  
LEFT OUTER JOIN countries USING (country\_id);

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 129**

View the Exhibits and examine the structures of the PROMOTIONS and SALES tables.

Evaluate the following SQL statement:

```
SQL>SELECT p.promo_id, p.promo_name, s.prod_id  
FROM sales s RIGHT OUTER JOIN promotions p  
ON (s.promo_id = p.promo_id);
```

Which statement is true regarding the output of the above query?

- A. It gives the details of promos for which there have been sales.
- B. It gives the details of promos for which there have been no sales.
- C. It gives details of all promos irrespective of whether they have resulted in a sale or not.
- D. It gives details of product ID s that have been sold irrespective of whether they had a promo or not.

**Answer:** C  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 130**

View the Exhibit and examine the data in the EMPLOYEES table:

You want to display all the employee names and their corresponding manager names.

Evaluate the following query:

```
SQL> SELECT e.employee_name "EMP NAME", m.employee_name "MGR NAME"  
FROM employees e _____ employees m  
ON e.manager_id = m.employee_id;
```

Which JOIN option can be used in the blank in the above query to get the required output?

- A. only inner JOIN
- B. only FULL OUTER JOIN
- C. only LEFT OUTER JOIN
- D. only RIGHT OUTER JOIN

**Answer:** C  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 131**

View the Exhibit and examine the structure of the PRODUCT, COMPONENT, and PDT\_COMP tables.

In PRODUCT table, PDTNO is the primary key.

In COMPONENT table, COMPNO is the primary key.

In PDT\_COMP table, (PDTNO,COMPNO) is the primary key, PDTNO is the foreign key referencing PDTNO in PRODUCT table and COMPNO is the foreign key referencing the COMPNO in COMPONENT table.

You want to generate a report listing the product names and their corresponding component names, if the component names and product names exist.

Evaluate the following query:

```
SQL>SELECT pdtno,pdtname, compno,compname  
FROM product _____ pdt_comp  
USING (pdtno) _____ component USING(compno)  
WHERE compname IS NOT NULL;
```

Which combination of joins used in the blanks in the above query gives the correct output?

- A. JOIN; JOIN
- B. FULL OUTER JOIN; FULL OUTER JOIN
- C. RIGHT OUTER JOIN; LEFT OUTER JOIN

D. LEFT OUTER JOIN; RIGHT OUTER JOIN

**Answer:** C  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 132**

View the Exhibit and examine the structure of the SALES and PRODUCTS tables.

In the SALES table, PROD\_ID is the foreign key referencing PROD\_ID in the PRODUCTS table,

You want to list each product ID and the number of times it has been sold.

Evaluate the following query:

```
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;
```

Which two JOIN options can be used in the blank in the above query to get the required output? (Choose two.)

**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)

- A. JOIN
- B. FULL OUTER JOIN
- C. LEFT OUTER JOIN
- D. RIGHT OUTER JOIN

**Answer:** BC  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 133**

Which two statements are true regarding subqueries? (Choose two.)

- A. A subquery can retrieve zero or more rows.

- B. Only two subqueries can be placed at one level.
- C. A subquery can be used only in SQL query statements.
- D. A subquery can appear on either side of a comparison operator.
- E. There is no limit on the number of subquery levels in the WHERE clause of a SELECT statement.

**Answer:** AD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 134**

Where can subqueries be used? (Choose all that apply.)

- A. field names in the SELECT statement
- B. the FROM clause in the SELECT statement
- C. the HAVING clause in the SELECT statement
- D. the GROUP BY clause in the SELECT statement
- E. the WHERE clause in only the SELECT statement
- F. the WHERE clause in SELECT as well as all DML statements

**Answer:** ABCF

**Section:** (none)

**Explanation/Reference:**

**QUESTION 135**

Which three statements are true regarding subqueries? (Choose three.)

- A. Subqueries can contain GROUP BY and ORDER BY clauses.
- B. Main query and subquery can get data from different tables.
- C. Main query and subquery must get data from the same tables.
- D. Subqueries can contain ORDER BY but not the GROUP BY clause.
- E. Only one column or expression can be compared between the main query and subquery.
- F. Multiple columns or expressions can be compared between the main query and subquery.

**Answer:** ABF

**Section:** (none)

**Explanation/Reference:**

**QUESTION 136**

View the Exhibit and examine the structure of the PRODUCTS table.

Which two tasks would require subqueries? (Choose two.)

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)

- A. Display the minimum list price for each product status.

- B. Display all suppliers whose list price is less than 1000.
- C. Display the number of products whose list price is more than the average list price.
- D. Display the total number of products supplied by supplier 102 and have product status as 'obsolete'.
- E. Display all products whose minimum list price is more than the average list price of products and have the status 'orderable'.

**Answer:** CE

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 137**

View the Exhibits and examine PRODUCTS and SALES tables.

You issue the following query to display product name and the number of times the product has been sold:

```
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;
```

What happens when the above statement is executed?

- A. The statement executes successfully and produces the required output.
- B. The statement produces an error because ITEM\_CNT cannot be displayed in the outer query.
- C. The statement produces an error because a subquery in the FROM clause and outer-joins cannot be used together.
- D. The statement produces an error because the GROUP BY clause cannot be used in a subquery in the FROM clause.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 138**

Which statement is true regarding subqueries?

- A. The LIKE operator cannot be used with single- row subqueries.
- B. The NOT IN operator is equivalent to IS NULL with single- row subqueries.
- C. =ANY and =ALL operators have the same functionality in multiple- row subqueries.
- D. The NOT operator can be used with IN, ANY, and ALL operators in multiple- row subqueries.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 139**

Which three statements are true about multiple-row subqueries? (Choose three.)

- A. They can contain a subquery within a subquery.
- B. They can return multiple columns as well as rows.
- C. They cannot contain a subquery within a subquery.
- D. They can return only one column but multiple rows.
- E. They can contain group functions and GROUP BY and HAVING clauses.
- F. They can contain group functions and the GROUP BY clause, but not the HAVING clause.

**Answer:** ABE  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 140**

Examine the structure of the PRODUCTS table:

```
name Null Type
PROD_ID NOT NULL NUMBER(4)
PROD_NAME VARCHAR2(20)
PROD_STATUS VARCHAR2(6)
QTY_IN_HAND NUMBER(8,2)
UNIT_PRICE NUMBER(10,2)
```

You want to display the names of the products that have the highest total value for UNIT\_PRICE \* QTY\_IN\_HAND.

Which SQL statement gives the required output?

- A. SELECT prod\_name  
FROM products  
WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products);
- B. SELECT prod\_name  
FROM products  
WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products  
GROUP BY prod\_name);
- C. SELECT prod\_name  
FROM products  
GROUP BY prod\_name  
HAVING MAX(unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products  
GROUP BY prod\_name);
- D. SELECT prod\_name  
FROM products  
WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(SUM(unit\_price \* qty\_in\_hand)) FROM products)  
GROUP BY prod\_name;

**Answer:** A  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 141**

View the Exhibit and examine the structure of CUSTOMERS and GRADES tables.

You need to display names and grades of customers who have the highest credit limit.

Which two SQL statements would accomplish the task? (Choose two.)

- A. SELECT custname, grade  
FROM customers, grades  
WHERE (SELECT MAX(cust\_credit\_limit)  
FROM customers) BETWEEN startval and endval;
- B. SELECT custname, grade  
FROM customers, grades  
WHERE (SELECT MAX(cust\_credit\_limit)  
FROM customers) BETWEEN startval and endval  
AND cust\_credit\_limit BETWEEN startval AND endval;
- C. SELECT custname, grade

```

FROM customers, grades
WHERE cust_credit_limit = (SELECT MAX(cust_credit_limit)
FROM customers)
AND cust_credit_limit BETWEEN startval AND endval;
D. SELECT custname, grade
FROM customers , grades
WHERE cust_credit_limit IN (SELECT MAX(cust_credit_limit)
FROM customers)
AND MAX(cust_credit_limit) BETWEEN startval AND endval;

```

**Answer:** BC  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 142**

View the Exhibit and examine the structure of the PRODUCTS table.

Evaluate the following query:

```

SQL> SELECT prod_name
  FROM products
 WHERE prod_id IN (SELECT prod_id FROM products
 WHERE prod_list_price =
 (SELECT MAX(prod_list_price)FROM products
 WHERE prod_list_price <
 (SELECT MAX(prod_list_price)FROM products)));

```

What would be the outcome of executing the above SQL statement?

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)

- A. It produces an error.
- B. It shows the names of all products in the table.
- C. It shows the names of products whose list price is the second highest in the table.
- D. It shows the names of all products whose list price is less than the maximum list price.

**Answer:** C  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 143**

View the Exhibit and examine the structure of the PROMOTIONS table.

You have to generate a report that displays the promo name and start date for all promos that started after

the last promo in the 'INTERNET' category.

Which query would give you the required output?

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 promotions WHERE promo\_begin\_date > ALL (SELECT MAX(promo\_begin\_date) FROM promotions )AND promo\_category = 'INTERNET';
- B. SELECT promo\_name, promo\_begin\_date FROM promotions WHERE promo\_begin\_date IN (SELECT promo\_begin\_date FROM promotions WHERE promo\_category='INTERNET');
- C. SELECT promo\_name, promo\_begin\_date FROM promotions WHERE promo\_begin\_date > ALL (SELECT promo\_begin\_date FROM promotions WHERE promo\_category = 'INTERNET');
- D. SELECT promo\_name, promo\_begin\_date FROM promotions WHERE promo\_begin\_date > ANY (SELECT promo\_begin\_date FROM promotions WHERE promo\_category = 'INTERNET');

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 144

View the Exhibit and examine the structure of the PRODUCTS table.

You want to display the category with the maximum number of items.

You issue the following query:

```
SQL>SELECT COUNT(*).prod_category_id  
FROM products  
GROUP BY prod_category_id  
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM products);
```

What is the outcome?

Table PRODUCTS		
Name	Null?	Type
<b>PROD_ID</b>	<b>NOT NULL</b>	<b>NUMBER(6)</b>
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)

- A. It executes successfully and gives the correct output.
- B. It executes successfully but does not give the correct output.
- C. It generates an error because the subquery does not have a GROUP BY clause.
- D. It generates an error because = is not valid and should be replaced by the IN operator.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 145

View the Exhibit and examine the structure of the CUSTOMERS table.

You issue the following SQL statement on the CUSTOMERS table to display the customers who are in the same country as customers with the last name 'KING' and whose credit limit is less than the maximum credit limit in countries that have customers with the last name 'KING':

```
SQL> SELECT cust_id,cust_last_name
  FROM customers
 WHERE country_id IN(SELECT country_id
  FROM customers
 WHERE cust_last_name ='King')
 AND cust_credit_limit < (SELECT MAX(cust_credit_limit)
  FROM customers
 WHERE country_id IN(SELECT country_id
  FROM customers
 WHERE cust_last_name='King'));
```

Which statement is true regarding the outcome of the above query?

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. It executes and shows the required result.
- B. It produces an error and the < operator should be replaced by < ALL to get the required output.
- C. It produces an error and the < operator should be replaced by < ANY to get the required output.
- D. It produces an error and the IN operator should be replaced by = in the WHERE clause of the main query to get the required output.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 146

Evaluate the following SQL statement:

```
SQL> SELECT cust_id, cust_last_name
  FROM customers
 WHERE cust_credit_limit IN
    (select cust_credit_limit
      FROM customers
     WHERE cust_city ='Singapore');
```

Which statement is true regarding the above query if one of the values generated by the subquery is NULL?

- A. It produces an error.
- B. It executes but returns no rows.
- C. It generates output for NULL as well as the other values produced by the subquery.
- D. It ignores the NULL value and generates output for the other values produced by the subquery.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 147

View the Exhibit and examine the structure of the PROMOTIONS table.

Evaluate the following SQL statement:

```
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;

```

Which statement is true regarding the outcome of the above query?

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. It shows COST\_REMARK for all the promos in the table.
- B. It produces an error because the subquery gives an error.
- C. It shows COST\_REMARK for all the promos in the promo category 'TV'.
- D. It produces an error because subqueries cannot be used with the CASE expression.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 148

View the Exhibit and examine the structure of the PRODUCTS tables.

You want to generate a report that displays the average list price of product categories where the average list price is less than half the maximum in each category.

Which query would give the correct output?

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)

- A. SELECT prod\_category,avg(prod\_list\_price)
   
FROM products
   
GROUP BY prod\_category
   
HAVING avg(prod\_list\_price) < ALL

- (SELECT max(prod\_list\_price)/2  
   FROM products  
   GROUP BY prod\_category);
- B. SELECT prod\_category,avg(prod\_list\_price)  
   FROM products  
   GROUP BY prod\_category  
   HAVING avg(prod\_list\_price) > ANY  
   (SELECT max(prod\_list\_price)/2  
   FROM products  
   GROUP BY prod\_category);
- C. SELECT prod\_category,avg(prod\_list\_price)  
   FROM products  
   HAVING avg(prod\_list\_price) < ALL  
   (SELECT max(prod\_list\_price)/2  
   FROM products  
   GROUP BY prod\_category);
- D. SELECT prod\_category,avg(prod\_list\_price)  
   FROM products  
   GROUP BY prod\_category  
   HAVING avg(prod\_list\_price) > ANY  
   (SELECT max(prod\_list\_price)/2  
   FROM products);

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 149

View the Exhibits and examine the structures of the COSTS and PROMOTIONS tables.

Evaluate the following SQL statement:

```
SQL> SELECT prod_id FROM costs
WHERE promo_id IN (SELECT promo_id FROM promotions
WHERE promo_cost < ALL
(SELECT MAX(promo_cost) FROM promotions
GROUP BY (promo_end_date-
promo_begin_date)));
```

What would be the outcome of the above SQL statement?

- A. It displays prod IDs in the promo with the lowest cost.
- B. It displays prod IDs in the promos with the lowest cost in the same time interval.
- C. It displays prod IDs in the promos with the highest cost in the same time interval.
- D. It displays prod IDs in the promos with cost less than the highest cost in the same time interval.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 150

View the Exhibit and examine the data in the PROMOTIONS table.

You need to display all promo categories that do not have 'discount' in their subcategory.

Which two SQL statements give the required result? (Choose two.)

- A. SELECT promo\_category  
FROM promotions  
MINUS  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory = 'discount';
- B. SELECT promo\_category  
FROM promotions  
INTERSECT  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory = 'discount';
- C. SELECT promo\_category  
FROM promotions  
MINUS  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory <> 'discount';
- D. SELECT promo\_category  
FROM promotions  
INTERSECT  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory <> 'discount';

**Answer:** AD

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 151**

View the Exhibit and examine the structure of the CUSTOMERS and CUST\_HISTORY tables.

The CUSTOMERS table contains the current location of all currently active customers. The CUST\_HISTORY table stores historical details relating to any changes in the location of all current as well as previous customers who are no longer active with the company.

You need to find those customers who have never changed their address.

Which SET operator would you use to get the required output?

- A. MINUS
- B. UNION
- C. INTERSECT
- D. UNION ALL

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 152**

Which statement is true regarding the UNION operator?

- A. By default, the output is not sorted.
- B. NULL values are not ignored during duplicate checking.
- C. Names of all columns must be identical across all SELECT statements.
- D. The number of columns selected in all SELECT statements need not be the same.

**Answer:** B  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 153**

View the Exhibits and examine the structures of the PRODUCTS and SALES tables.

Which two SQL statements would give the same output? (Choose two.)

- A. SELECT prod\_id FROM products  
INTERSECT  
SELECT prod\_id FROM sales;
- B. SELECT prod\_id FROM products  
MINUS  
SELECT prod\_id FROM sales;
- C. SELECT DISTINCT p.prod\_id  
FROM products p JOIN sales s  
ON p.prod\_id=s.prod\_id;
- D. SELECT DISTINCT p.prod\_id  
FROM products p JOIN sales s  
ON p.prod\_id <> s.prod\_id;

**Answer:** AC  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 154**

View the Exhibit and evaluate structures of the SALES, PRODUCTS, and COSTS tables.

Evaluate the following SQL statement:

```
SQL>SELECT prod_id FROM products  
INTERSECT  
SELECT prod_id FROM sales  
MINUS  
SELECT prod_id FROM costs;
```

Which statement is true regarding the above compound query?

- A. It produces an error.
- B. It shows products that were sold and have a cost recorded.
- C. It shows products that were sold but have no cost recorded.
- D. It shows products that have a cost recorded irrespective of sales.

**Answer:** C  
**Section:** (none)

**Explanation/Reference:**

**QUESTION 155**

Evaluate the following SQL statement:

```
SQL> SELECT promo_id, promo_category  
FROM promotions
```

```
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';
```

Which statement is true regarding the outcome of the above query?

- A. It executes successfully and displays rows in the descending order of PROMO\_CATEGORY.
- B. It produces an error because positional notation cannot be used in the ORDER BY clause with SET operators.
- C. It executes successfully but ignores the ORDER BY clause because it is not located at the end of the compound statement.
- D. It produces an error because the ORDER BY clause should appear only at the end of a compound query—that is, with the last SELECT statement.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 156**

Evaluate the following SQL statement:

```
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;
```

Which ORDER BY clauses are valid for the above query? (Choose all that apply.)

- A. ORDER BY 2,1
- B. ORDER BY CUST\_NO
- C. ORDER BY 2,cust\_id
- D. ORDER BY "CUST\_NO"
- E. ORDER BY "Last Name"

**Answer:** ACE

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 157**

View the Exhibit and examine the structure of the ORDERS and CUSTOMERS tables.

Evaluate the following SQL command:

```

SQL> SELECT o.order_id, c.cust_name, o.order_total, c.credit_limit
  FROM orders o JOIN customers c
    USING (customer_id)
   WHERE o.order_total > c.credit_limit
  FOR UPDATE
 ORDER BY o.order_id;

```

Which two statements are true regarding the outcome of the above query? (Choose two.)

#### **ORDERS**

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

#### **CUSTOMERS**

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(6)
CUST_NAME	NOT NULL	VARCHAR2(20)
CUST_ADDRESS		VARCHAR2(50)
CREDIT_LIMIT		NUMBER(9, 2)

- A. It locks all the rows that satisfy the condition in the statement.
- B. It locks only the columns that satisfy the condition in both the tables.
- C. The locks are released only when a COMMIT or ROLLBACK is issued.
- D. The locks are released after a DML statement is executed on the locked rows.

**Answer:** AC

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 158**

Which statements are true regarding the FOR UPDATE clause in a SELECT statement? (Choose all that apply.)

- A. It locks only the columns specified in the SELECT list.
- B. It locks the rows that satisfy the condition in the SELECT statement.
- C. It can be used only in SELECT statements that are based on a single table.
- D. It can be used in SELECT statements that are based on a single or multiple tables.
- E. After it is enforced by a SELECT statement, no other query can access the same rows until a COMMIT or ROLLBACK is issued.

**Answer:** BD

**Section:** (none)

**Explanation/Reference:**

**QUESTION 159**

View the Exhibit and examine the structure of the CUSTOMERS table.

NEW\_CUSTOMERS is a new table with the columns CUST\_ID, CUST\_NAME and CUST\_CITY that have the same data types and size as the corresponding columns in the CUSTOMERS table.

Evaluate the following INSERT statement:

```
INSERT INTO new_customers (cust_id, cust_name, cust_city)
VALUES(SELECT cust_id,cust_first_name' 'cust_last_name,cust_city
FROM customers
WHERE cust_id > 23004);
```

The INSERT statement fails when executed. What could be the reason?

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. The VALUES clause cannot be used in an INSERT with a subquery.
- B. Column names in the NEW\_CUSTOMERS and CUSTOMERS tables do not match.
- C. The WHERE clause cannot be used in a subquery embedded in an INSERT statement.
- D. The total number of columns in the NEW\_CUSTOMERS table does not match the total number of columns in the CUSTOMERS table.

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 160**

View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables.

There is only one customer with the cust\_last\_name column having value Roberts. Which INSERT statement should be used to add a row into the ORDERS table for the customer whose CUST\_LAST\_NAME is Roberts and CREDIT\_LIMIT is 600?

**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)

- A. INSERT INTO orders  
VALUES (1,'10-mar-2007', 'direct',  
(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600), 1000);
- B. INSERT INTO orders (order\_id,order\_date,order\_mode,  
(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600),order\_total)  
VALUES(1,'10-mar-2007', 'direct', &&customer\_id, 1000);
- C. INSERT INTO(SELECT o.order\_id, o.order\_date,o.order\_mode,c.customer\_id, o.order\_total  
FROM orders o, customers c  
WHERE o.customer\_id = c.customer\_id  
AND c.cust\_last\_name='Roberts' AND c.credit\_limit=600 )  
VALUES (1,'10-mar-2007', 'direct',(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600), 1000);
- D. INSERT INTO orders (order\_id,order\_date,order\_mode,  
(SELECT customer\_id  
FROM customers  
WHERE cust\_last\_name='Roberts' AND  
credit\_limit=600),order\_total)  
VALUES(1,'10-mar-2007', 'direct', &customer\_id, 1000);

**Answer:** A**Section:** (none)**Explanation/Reference:****QUESTION 161**

View the exhibit and examine the description for the SALES and CHANNELS tables.

You issued the following SQL statement to insert a row in the SALES table:

INSERT INTO sales VALUES

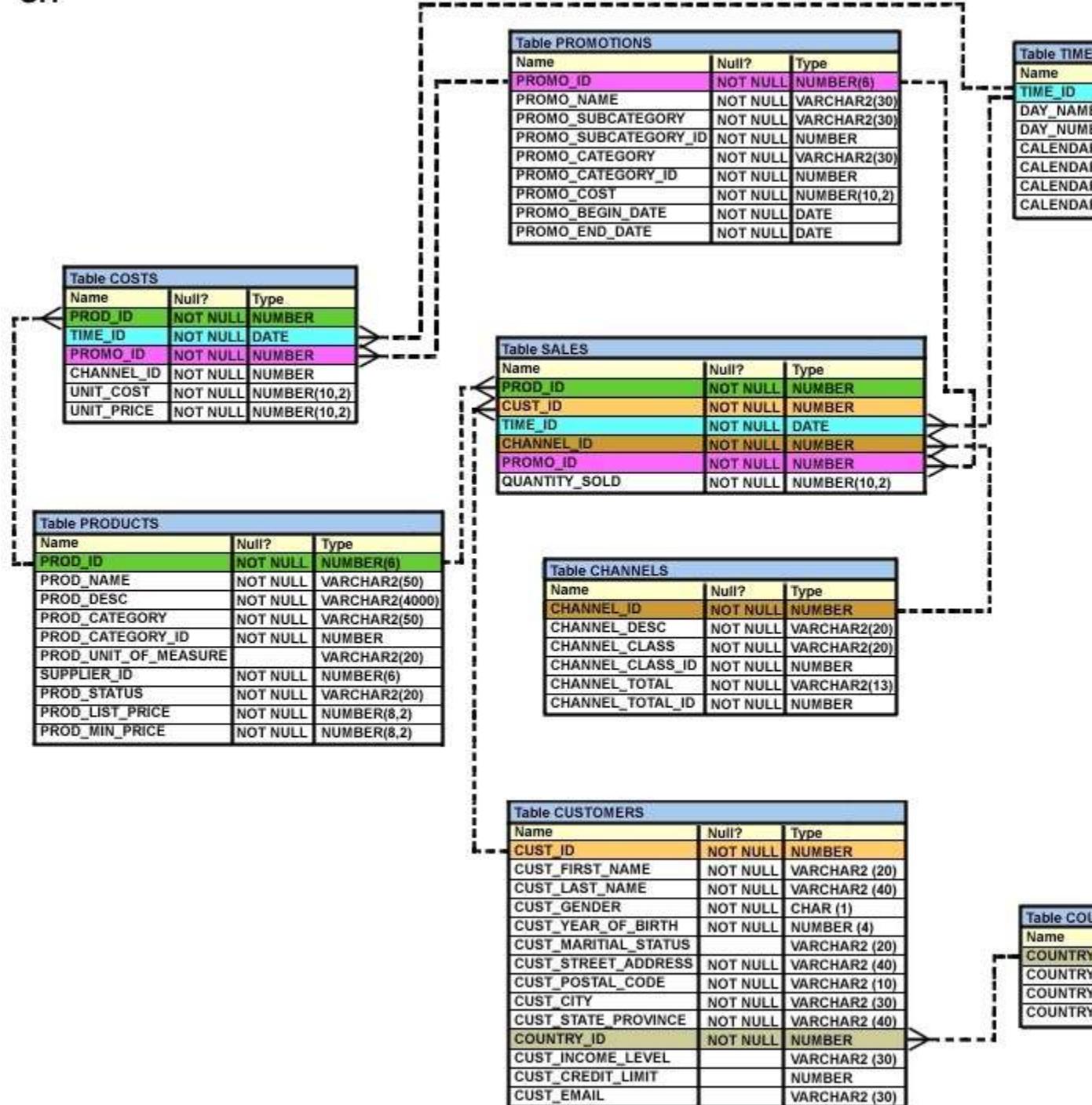
(23, 2300, SYSDATE, (SELECT channel\_id

FROM channels

WHERE channel\_desc='Direct Sales'), 12, 1, 500);

Which statement is true regarding the execution of the above statement?

SH



- A. The statement will execute and the new row will be inserted in the SALES table.
- B. The statement will fail because subquery cannot be used in the VALUES clause.

- C. The statement will fail because the VALUES clause is not required with subquery.
- D. The statement will fail because subquery in the VALUES clause is not enclosed with in single quotation marks .

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 162**

View the Exhibit and examine the structure of the PRODUCTS, SALES, and SALE\_SUMMARY tables.

SALE\_VW is a view created using the following command :

```
SQL>CREATE VIEW sale_vw AS  
SELECT prod_id, SUM(quantity_sold) QTY_SOLD  
FROM sales GROUP BY prod_id;
```

You issue the following command to add a row to the SALE\_SUMMARY table :

```
SQL>INSERT INTO sale_summary  
SELECT prod_id, prod_name, qty_sold FROM sale_vw JOIN products  
USING (prod_id) WHERE prod_id = 16;
```

What is the outcome?

- A. It executes successfully.
- B. It gives an error because a complex view cannot be used to add data into the SALE\_SUMMARY table.
- C. It gives an error because the column names in the subquery and the SALE\_SUMMARY table do not match.
- D. It gives an error because the number of columns to be inserted does not match with the number of columns in the SALE\_SUMMARY table.

**Answer:** D

**Section:** (none)

**Explanation/Reference:**

**QUESTION 163**

View the Exhibit and examine the description for the CUSTOMERS table.

You want to update the CUST\_CREDIT\_LIMIT column to NULL for all the customers, where

CUST\_INCOME\_LEVEL has NULL in the CUSTOMERS table. Which SQL statement will accomplish the task?

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. UPDATE customers  
SET cust\_credit\_limit = NULL  
WHERE CUST\_INCOME\_LEVEL = NULL;
- B. UPDATE customers  
SET cust\_credit\_limit = NULL  
WHERE cust\_income\_level IS NULL;
- C. UPDATE customers  
SET cust\_credit\_limit = TO\_NUMBER(NULL)  
WHERE cust\_income\_level = TO\_NUMBER(NULL);
- D. UPDATE customers  
SET cust\_credit\_limit = TO\_NUMBER(' ',9999)  
WHERE cust\_income\_level IS NULL;

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 164

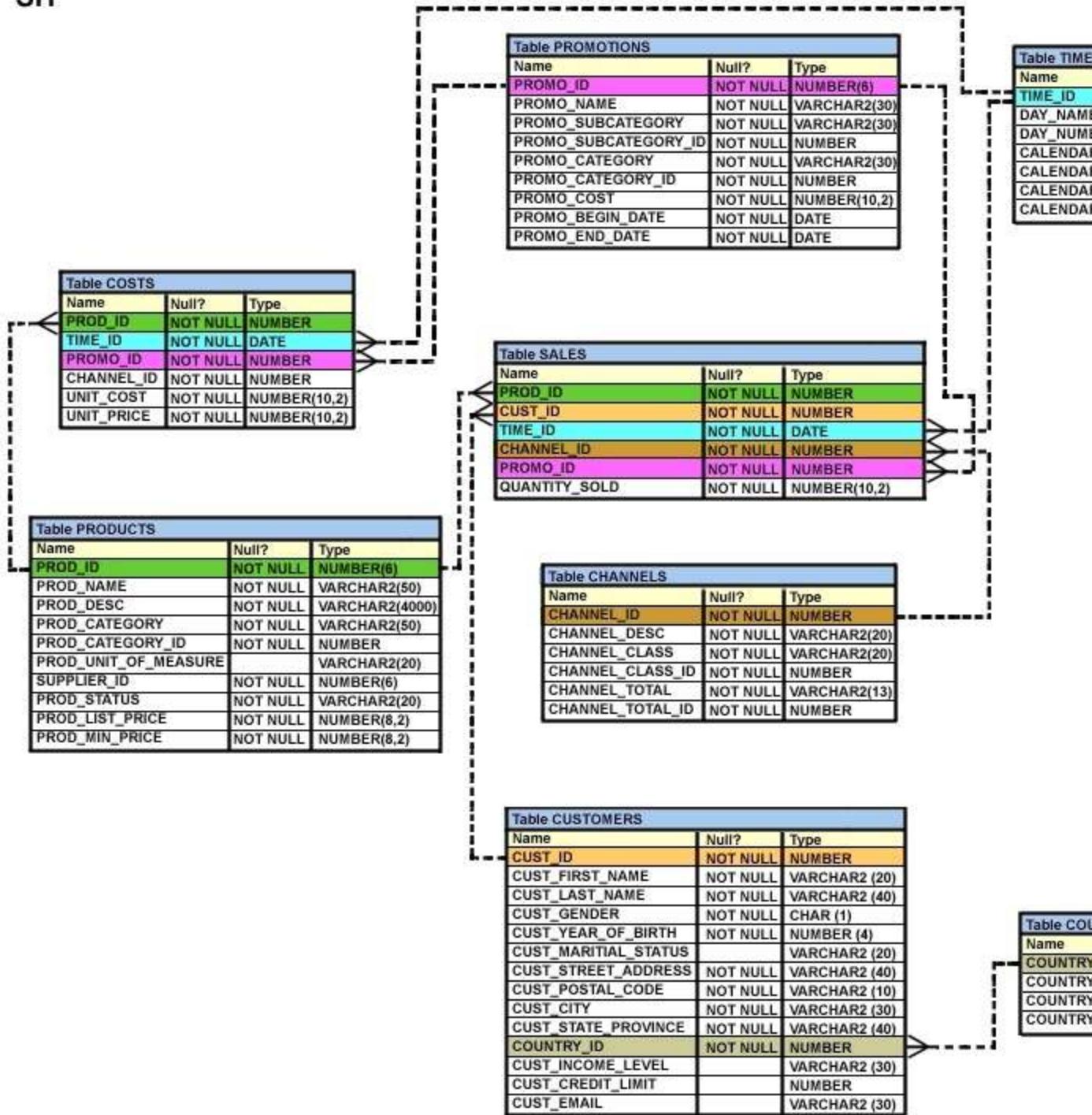
View the Exhibit and examine the structure of CUSTOMERS and SALES tables.

Evaluate the following SQL statement:

```
UPDATE (SELECT prod_id, cust_id, quantity_sold, time_id
FROM sales)
SET time_id = '22-MAR-2007'
WHERE cust_id = (SELECT cust_id
FROM customers
WHERE cust_last_name = 'Roberts' AND
credit_limit = 600);
```

Which statement is true regarding the execution of the above UPDATE statement?

SH



- A. It would not execute because two tables cannot be used in a single UPDATE statement.
- B. It would not execute because the SELECT statement cannot be used in place of the table name.
- C. It would execute and restrict modifications to only the columns specified in the SELECT statement.
- D. It would not execute because a subquery cannot be used in the WHERE clause of an UPDATE statement.

Answer: C  
Section: (none)

**Explanation/Reference:**

**QUESTION 165**

View the Exhibit and examine the description for the CUSTOMERS table.

You want to update the CUST\_INCOME\_LEVEL and CUST\_CREDIT\_LIMIT columns for the customer with the CUST\_ID 2360. You want the value for the CUST\_INCOME\_LEVEL to have the same value as that of the customer with the CUST\_ID 2560 and the CUST\_CREDIT\_LIMIT to have the same value as that of the customer with CUST\_ID 2566.

Which UPDATE statement will accomplish the task?

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. UPDATE customers  
SET cust\_income\_level = (SELECT cust\_income\_level  
FROM customers  
WHERE cust\_id = 2560),  
cust\_credit\_limit = (SELECT cust\_credit\_limit  
FROM customers  
WHERE cust\_id = 2566)  
WHERE cust\_id=2360;
- B. UPDATE customers  
SET (cust\_income\_level,cust\_credit\_limit) = (SELECT  
cust\_income\_level, cust\_credit\_limit  
FROM customers  
WHERE cust\_id=2560 OR cust\_id=2566)  
WHERE cust\_id=2360;
- C. UPDATE customers  
SET (cust\_income\_level,cust\_credit\_limit) = (SELECT  
cust\_income\_level, cust\_credit\_limit  
FROM customers  
WHERE cust\_id IN(2560, 2566)  
WHERE cust\_id=2360;
- D. UPDATE customers  
SET (cust\_income\_level,cust\_credit\_limit) = (SELECT  
cust\_income\_level, cust\_credit\_limit  
FROM customers  
WHERE cust\_id=2560 AND cust\_id=2566)  
WHERE cust\_id=2360;

**Answer:** A

**Section:** (none)

**Explanation/Reference:**

**QUESTION 166**

View the Exhibit and examine the structures of the EMPLOYEES and DEPARTMENTS tables.

You want to update the EMPLOYEES table as follows:  
4 ? 4;

- Update only those employees who work in Boston or Seattle (locations 2900 and 2700).
- Set department\_id for these employees to the department\_id corresponding to London (location\_id 2100).
- Set the employees' salary in location\_id 2100 to 1.1 times the average salary of their department.
- Set the employees' commission in location\_id 2100 to 1.5 times the average commission of their department.

You issue the following command:

```
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)
```

What is the outcome?

- A. It executes successfully and gives the correct result.
- B. It executes successfully but does not give the correct result.
- C. It generates an error because a subquery cannot have a join condition in an UPDATE statement.
- D. It generates an error because multiple columns (SALARY, COMMISION) cannot be specified together in an UPDATE statement.

**Answer:** B

**Section:** (none)

**Explanation/Reference:**

**QUESTION 167**

Evaluate the following DELETE statement:

```
DELETE FROM sales;
```

There are no other uncommitted transactions on the SALES table.

Which statement is true about the DELETE statement?

- A. It would not remove the rows if the table has a primary key.

- B. It removes all the rows as well as the structure of the table.
- C. It removes all the rows in the table and deleted rows can be rolled back.
- D. It removes all the rows in the table and deleted rows cannot be rolled back.

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

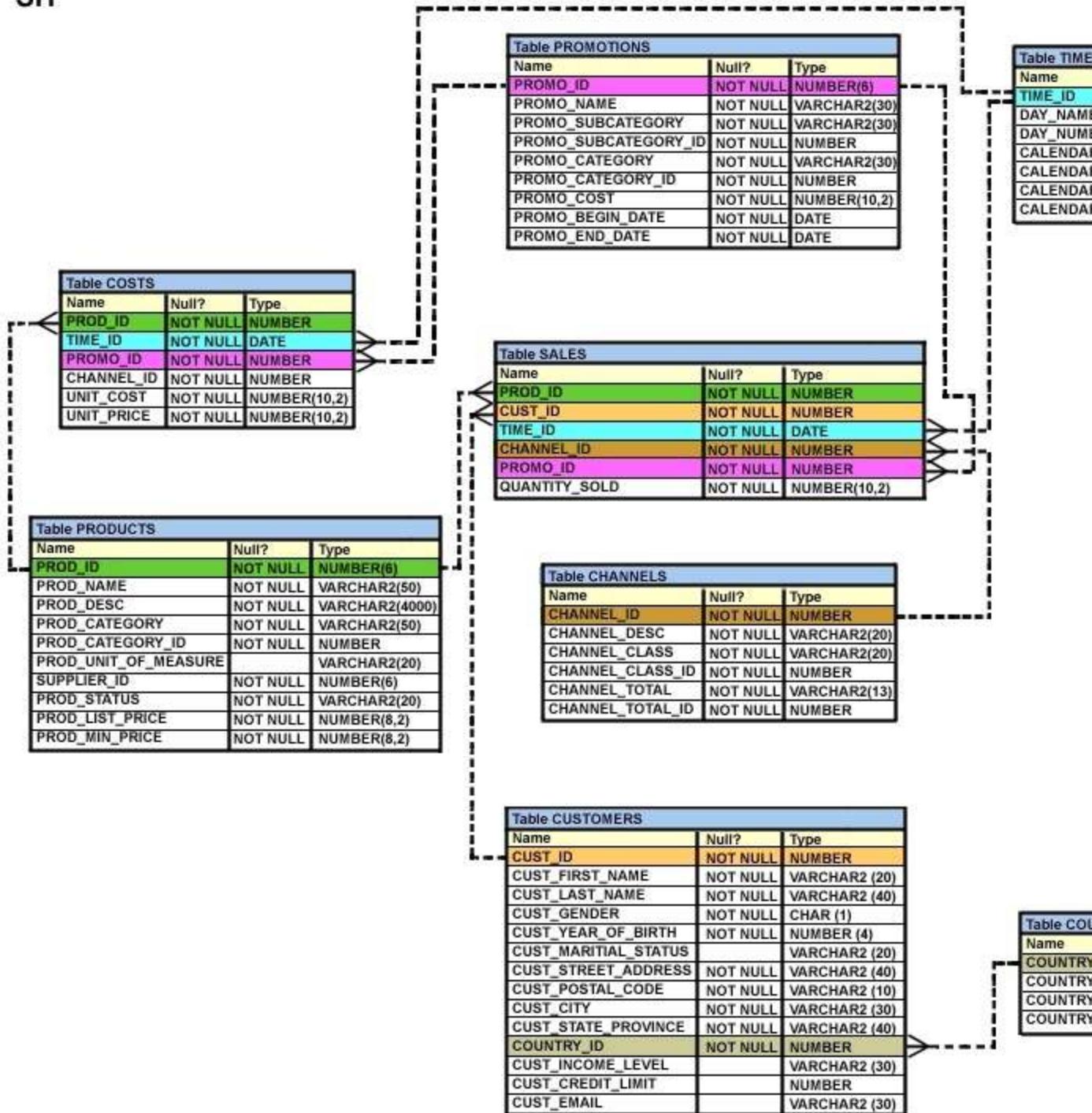
**QUESTION 168**

View the Exhibit and examine the description of SALES and PROMOTIONS tables.

You want to delete rows from the SALES table, where the PROMO\_NAME column in the PROMOTIONS table has either blowout sale or everyday low price as values.

Which DELETE statements are valid? (Choose all that apply.)

SH



A. DELETE  
 FROM sales  
 WHERE promo\_id = (SELECT promo\_id  
 FROM promotions  
 WHERE promo\_name = 'blowout sale')  
 AND promo\_id = (SELECT promo\_id  
 FROM promotions  
 WHERE promo\_name = 'everyday low price');

- B. DELETE  
FROM sales  
WHERE promo\_id = (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'blowout sale')  
OR promo\_id = (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'everyday low price');
- C. DELETE  
FROM sales  
WHERE promo\_id IN (SELECT promo\_id  
FROM promotions  
WHERE promo\_name = 'blowout sale'  
OR promo\_name = 'everyday low price');
- D. DELETE

FROM sales  
WHERE promo\_id IN (SELECT promo\_id  
FROM promotions

```
WHERE promo_name IN ('blowout sale','everyday low price'));
```

**Answer:** BCD

**Section:** (none)

**Explanation/Reference:**

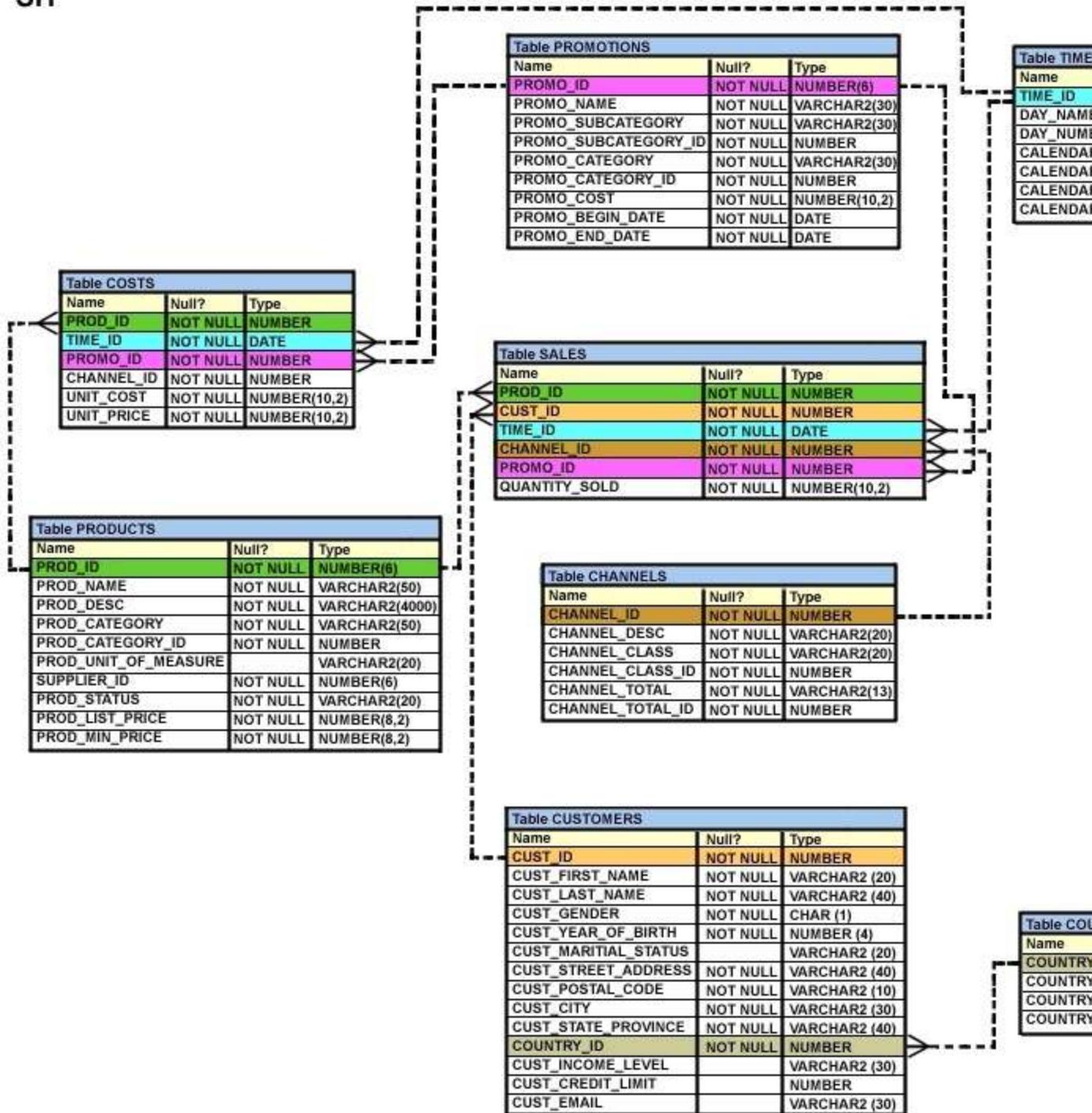
**QUESTION 169**

View the Exhibit and examine the description for the PRODUCTS and SALES table.

PROD\_ID is a primary key in the PRODUCTS table and foreign key in the SALES table. You want to remove all the rows from the PRODUCTS table for which no sale was done for the last three years.

Which is the valid DELETE statement?

SH



- A. DELETE  
FROM products  
WHERE prod\_id = (SELECT prod\_id  
FROM sales  
WHERE time\_id - 3\*365 = SYSDATE );
- B. DELETE  
FROM products  
WHERE prod\_id = (SELECT prod\_id

```

    FROM sales
    WHERE SYSDATE >= time_id - 3*365 );
C. DELETE
    FROM products
    WHERE prod_id IN (SELECT prod_id
    FROM sales
    WHERE SYSDATE - 3*365 >= time_id);
D. DELETE
    FROM products
    WHERE prod_id IN (SELECT prod_id
    FROM sales
    WHERE time_id >= SYSDATE - 3*365 );

```

**Answer:** C

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 170

Which two statements are true regarding the DELETE and TRUNCATE commands? (Choose two.)

- A. DELETE can be used to remove only rows from only one table at a time.
- B. DELETE can be used to remove only rows from multiple tables at a time.
- C. DELETE can be used only on a table that is a parent of a referential integrity constraint.
- D. DELETE can be used to remove data from specific columns as well as complete rows.
- E. DELETE and TRUNCATE can be used on a table that is a parent of a referential integrity constraint having ON DELETE rule .

**Answer:** AE

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 171

Which three statements/commands would cause a transaction to end? (Choose three.)

- A. COMMIT
- B. SELECT
- C. CREATE
- D. ROLLBACK
- E. SAVEPOINT

**Answer:** ACD

**Section:** (none)

**Explanation/Reference:**

#### QUESTION 172

The SQL statements executed in a user session are as follows:

```

SQL> CREATE TABLE product
      (PCODE NUMBER(2),
       PNAME VARCHAR2(10));
SQL> INSERT INTO product VALUES (1, 'pen');
SQL> INSERT INTO product VALUES (2,'pencil');
SQL> SAVEPOINT a;

```

```
SQL> UPDATE product SET pcode = 10 WHERE pcode = 1;
```

```
SQL> SAVEPOINT b;
```

```
SQL> DELETE FROM product WHERE pcode = 2;
```

```
SQL> COMMIT;
```

```
SQL> DELETE FROM product WHERE pcode=10;
```

Which two statements describe the consequences of issuing the ROLLBACK TO SAVE POINT a command in the session? (Choose two.)

- A. The rollback generates an error.
- B. No SQL statements are rolled back.
- C. Only the DELETE statements are rolled back.
- D. Only the second DELETE statement is rolled back.
- E. Both the DELETE statements and the UPDATE statement are rolled back.

**Answer:** AB

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 173**

When does a transaction complete? (Choose all that apply.)

- A. when a DELETE statement is executed
- B. when a ROLLBACK command is executed
- C. when a PL/SQL anonymous block is executed
- D. when a data definition language ( DDL) statement is executed
- E. when a TRUNCATE statement is executed after the pending transaction

**Answer:** BDE

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 174**

Which statement is true regarding transactions? (Choose all that apply.)

- A. A transaction can consist only of a set of DML and DDL statements.
- B. A part or an entire transaction can be undone by using ROLLBACK command .
- C. A transaction consists of a set of DML or DCL statements.
- D. A part or an entire transaction can be made permanent with a COMMIT.
- E. A transaction can consist of only a set of queries or DML or DDL statements.

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

#### **QUESTION 175**

Which two statements are true regarding savepoints? (Choose two.)

- A. Savepoints are effective only for COMMIT.
- B. Savepoints may be used to ROLLBACK.
- C. Savepoints can be used for only DML statements.
- D. Savepoints are effective for both COMMIT and ROLLBACK.

E. Savepoints can be used for both DML and DDL statements.

**Answer:** BC

**Section:** (none)

**Explanation/Reference:**

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

**Question: 1**

Which statement is true regarding the INTERSECT operator?

- A. It ignores NULL values.
- B. Reversing the order of the intersected tables alters the result.
- C. The names of columns in all SELECT statements must be identical.
- D. The number of columns and data types must be identical for all SELECT statements in the query.

**Answer: D**

**Question: 2**

Which three statements are true regarding the data types in Oracle Database 0g/11g? (Choose three.)

- A. Only one LONG column can be used per table.
- B. A TIMESTAMP data type column stores only time values with fractional seconds.
- C. The BLOB data type column is used to store binary data in an operating system file.
- D. The minimum column width that can be specified for a VARCHAR2 data type column is one.
- E. The value for a CHAR data type column is blank-padded to the maximum defined column width.

**Answer: A, D, E**

**Question: 3**

Examine the structure of the PROGRAMS table:

Name Null? Type

---

```
PROG_ID NOT NULL NUMBER(3)
PROG_COST NUMBER(8,2)
START_DATE NOT NULL DATE
END_DATE DATE
```

Which two SQL statements would execute successfully? (Choose two.)

- A. SELECT NVL(ADD\_MONTHS(END\_DATE,1),SYSDATE) FROM programs;
- B. SELECT TO\_DATE(NVL(SYSDATE-END\_DATE,SYSDATE)) FROM programs;
- C. SELECT NVL(MONTHS\_BETWEEN(start\_date,end\_date),'Ongoing') FROM programs;
- D. SELECT NVL(TO\_CHAR(MONTHS\_BETWEEN(start\_date,end\_date)),'Ongoing') FROM programs;

**Answer: A, D**

**Question: 4**

View the Exhibit and examine the structure of the PROMOTIONS table.

Which two SQL statements would execute successfully? (Choose two.)

Exam Name:	Oracle Database: SQL Fundamentals		
Exam Type:	Oracle		
Exam Code:	1Z0-051	Total Questions	114

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. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE TO\_CHAR(promo\_end\_date, 'yyyy') > '2000';
- B. SELECT promo\_begin\_date  
FROM promotions  
WHERE TO\_CHAR(promo\_begin\_date,'mon dd yy')='jul 01 98';
- C. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8));
- D. SELECT TO\_CHAR(promo\_begin\_date,'dd/month')  
FROM promotions  
WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'));

**Answer: A, B**

**Question: 5**

View the Exhibit and evaluate structures of the SALES, PRODUCTS, and COSTS tables.

Evaluate the following SQL statement:

SQL>SELECT prod\_id FROM products

INTERSECT

SELECT prod\_id FROM sales

MINUS

SELECT prod\_id FROM costs;

Which statement is true regarding the above compound query?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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)

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)

Table COSTS		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
PROMO_ID	NOT NULL	NUMBER
CHANNEL_ID	NOT NULL	NUMBER
UNIT_COST	NOT NULL	NUMBER(10,2)
UNIT_PRICE	NOT NULL	NUMBER(10,2)

- A. It produces an error.
- B. It shows products that were sold and have a cost recorded.
- C. It shows products that were sold but have no cost recorded.
- D. It shows products that have a cost recorded irrespective of sales.

**Answer: C**

#### **Question: 6**

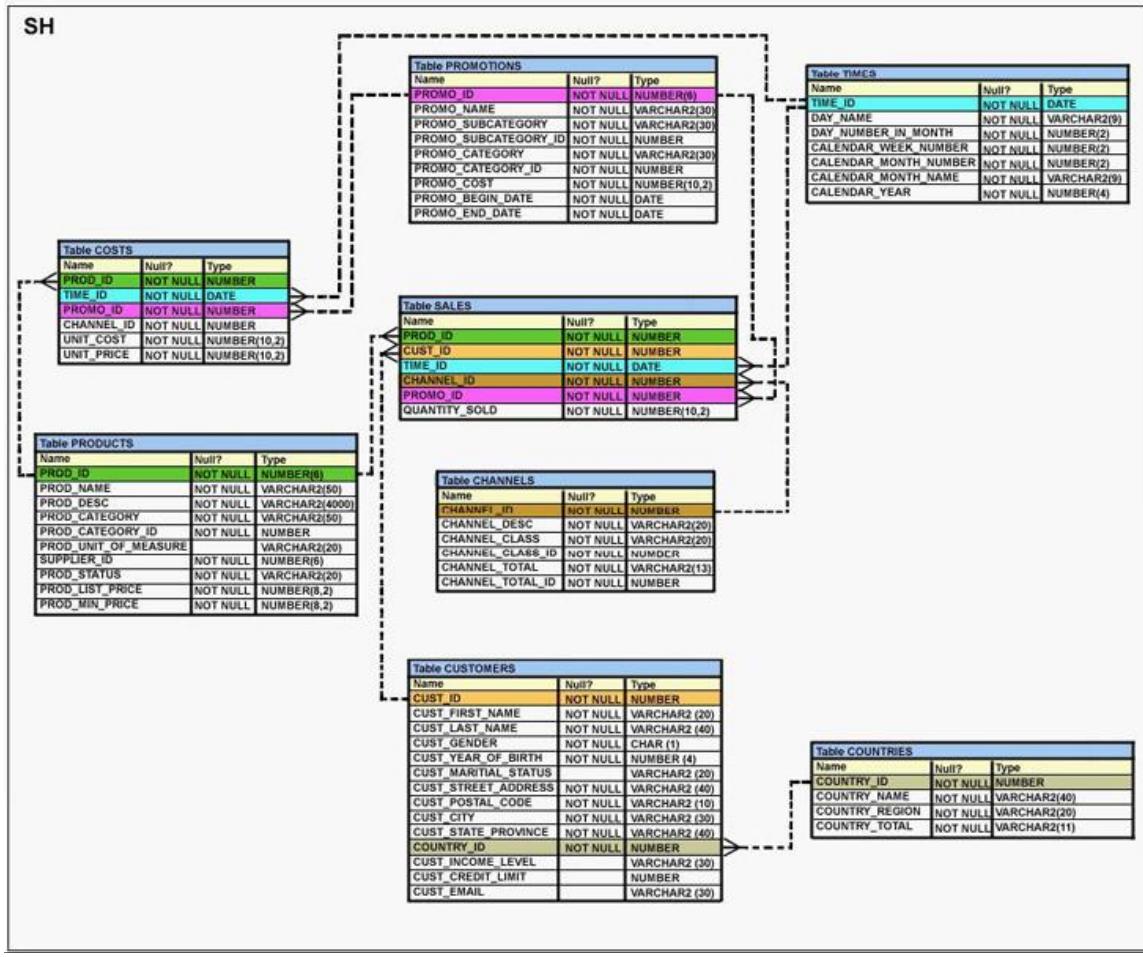
View the Exhibit and examine the structure of CUSTOMERS and SALES tables.

Evaluate the following SQL statement:

```
UPDATE (SELECT prod_id, cust_id, quantity_sold, time_id
FROM sales)
SET time_id = '22-MAR-2007'
WHERE cust_id = (SELECT cust_id
FROM customers
WHERE cust_last_name = 'Roberts' AND
credit_limit = 600);
```

Which statement is true regarding the execution of the above UPDATE statement?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>	
<b>Exam Type:</b>	<b>Oracle</b>	
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b> <b>114</b>



- A. It would not execute because two tables cannot be used in a single UPDATE statement.
- B. It would not execute because the SELECT statement cannot be used in place of the table name.
- C. It would execute and restrict modifications to only the columns specified in the SELECT statement.
- D. It would not execute because a subquery cannot be used in the WHERE clause of an UPDATE statement.

**Answer: C**

### Question: 7

You are currently located in Singapore and have connected to a remote database in Chicago.

You issue the following command:

```
SQL> SELECT ROUND(SYSDATE-promo_begin_date,0)
```

FROM promotions

WHERE (SYSDATE-promo\_begin\_date)/365 > 2;

PROMOTIONS is the public synonym for the public database link for the PROMOTIONS table. What is the outcome?

- A. an error because the ROUND function specified is invalid
- B. an error because the WHERE condition specified is invalid
- C. number of days since the promo started based on the current Chicago date and time
- D. number of days since the promo started based on the current Singapore date and time

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

**Answer: C**

**Question: 8**

You need to display the first names of all customers from the CUSTOMERS table that contain the character 'e' and have the character 'a' in the second last position. Which query would give the required output?

- A. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')<>0 AND  
SUBSTR(cust\_first\_name, -2, 1)='a';
- B. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')<>" AND  
SUBSTR(cust\_first\_name, -2, 1)='a';
- C. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')IS NOT NULL AND  
SUBSTR(cust\_first\_name, 1,-2)='a';
- D. SELECT cust\_first\_name  
FROM customers  
WHERE INSTR(cust\_first\_name, 'e')<>0 AND  
SUBSTR(cust\_first\_name, LENGTH(cust\_first\_name),-2)='a';

**Answer: A**

**Question: 9**

Evaluate the following query:

```
SQL> SELECT TRUNC(ROUND(156.00,-1),-1)
  FROM DUAL;
```

What would be the outcome?

- A. 16
- B. 100
- C. 160
- D. 200
- E. 150

**Answer: C**

**Question: 10**

Which two statements are true regarding the COUNT function? (Choose two.)

- A. The COUNT function can be used only for CHAR, VARCHAR2, and NUMBER data types.
- B. COUNT(\*) returns the number of rows including duplicate rows and rows containing NULL value in any of the columns.
- C. COUNT(cust\_id) returns the number of rows including rows with duplicate customer IDs and NULL value in the CUST\_ID column
- D. COUNT(DISTINCT inv\_amt) returns the number of rows excluding rows containing duplicates and NULL values in the INV\_AMT column.
- E. A SELECT statement using the COUNT function with a DISTINCT keyword cannot have a WHERE clause.

**Answer: B, D**

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

### Question: 11

View the Exhibits and examine the structures of the PROMOTIONS and SALES tables.  
Evaluate the following SQL statement:

```
SQL>SELECT p.promo_id, p.promo_name, s.prod_id
  FROM sales s RIGHT OUTER JOIN promotions p
  ON (s.promo_id = p.promo_id);
```

Which statement is true regarding the output of the above query?

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

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)

- A. It gives the details of promos for which there have been sales.
- B. It gives the details of promos for which there have been no sales.
- C. It gives details of all promos irrespective of whether they have resulted in a sale or not.
- D. It gives details of product IDs that have been sold irrespective of whether they had a promo or not.

### Answer: C

### Question: 12

View the Exhibit and examine the structure of the CUSTOMERS table.

Evaluate the query statement:

```
SQL> SELECT cust_last_name, cust_city, cust_credit_limit
  FROM customers
 WHERE cust_last_name BETWEEN 'A' AND 'C' AND cust_credit_limit BETWEEN
 1000 AND 3000;
```

What would be the outcome of the above statement?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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. It executes successfully.
- B. It produces an error because the condition on CUST\_LAST\_NAME is invalid.
- C. It executes successfully only if the CUST\_CREDIT\_LIMIT column does not contain any null values.
- D. It produces an error because the AND operator cannot be used to combine multiple BETWEEN clauses.

**Answer: A**

**Question: 13**

Which two statements are true regarding the USING and ON clauses in table joins? (Choose two.)

- A. Both USING and ON clauses can be used for equijoins and nonequijoins.
- B. A maximum of one pair of columns can be joined between two tables using the ON clause.
- C. The ON clause can be used to join tables on columns that have different names but compatible data types.
- D. The WHERE clause can be used to apply additional conditions in SELECT statements containing the ON or the USING clause.

**Answer: C, D**

**Question: 14**

Where can subqueries be used? (Choose all that apply.)

- A. field names in the SELECT statement
- B. the FROM clause in the SELECT statement
- C. the HAVING clause in the SELECT statement
- D. the GROUP BY clause in the SELECT statement
- E. the WHERE clause in only the SELECT statement
- F. the WHERE clause in SELECT as well as all DML statements

**Answer: A, B, C, F**

**Question: 15**

Using the CUSTOMERS table, you need to generate a report that shows 50% of each credit amount in each income level. The report should NOT show any repeated credit amounts in each income level. Which query would give the required result?

- A. SELECT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- AS "50% Credit Limit"  
 FROM customers;
- B. SELECT DISTINCT cust\_income\_level, DISTINCT cust\_credit\_limit \* 0.50  
 AS "50% Credit Limit"  
 FROM customers;
- C. SELECT DISTINCT cust\_income\_level || ' ' || cust\_credit\_limit \* 0.50  
 AS "50% Credit Limit"  
 FROM customers;
- D. SELECT cust\_income\_level ||' '| cust\_credit\_limit \* 0.50 AS "50% Credit Limit"  
 FROM customers;

**Answer: C**

**Question: 16**

Which statement is true regarding the UNION operator?

- A. By default, the output is not sorted.
- B. NULL values are not ignored during duplicate checking.
- C. Names of all columns must be identical across all SELECT statements.
- D. The number of columns selected in all SELECT statements need not be the same.

**Answer: B**

**Question: 17**

Which two statements are true regarding working with dates? (Choose two.)

- A. The default internal storage of dates is in the numeric format.
- B. The default internal storage of dates is in the character format.
- C. The RR date format automatically calculates the century from the SYSDATE function and does not allow the user to enter the century.
- D. The RR date format automatically calculates the century from the SYSDATE function but allows the user to enter the century if required.

**Answer: A, D**

**Question: 18**

The ORDERS table belongs to the user OE. OE has granted the SELECT privilege on the ORDERS table to the user HR. Which statement would create a synonym ORD so that HR can execute the following query successfully?

SELECT \* FROM ord;

- A. CREATE SYNONYM ord FOR orders; This command is issued by OE.
- B. CREATE PUBLIC SYNONYM ord FOR orders; This command is issued by OE.
- C. CREATE SYNONYM ord FOR oe.orders; This command is issued by the database administrator.
- D. CREATE PUBLIC SYNONYM ord FOR oe.orders; This command is issued by the database administrator.

**Answer: D**

**Question: 19**

View the Exhibit and examine the structure of the PROMOTIONS table.

Which SQL statements are valid? (Choose all that apply.)

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

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\_id, DECODE(NVL(promo\_cost,0), promo\_cost, promo\_cost \* 0.25, 100) "Discount"  
FROM promotions;
- B. SELECT promo\_id, DECODE(promo\_cost, 10000, DECODE(promo\_category, 'G1', promo\_cost \*.25, NULL), NULL) "Catcost"  
FROM promotions;
- C. SELECT promo\_id, DECODE(NULLIF(promo\_cost, 10000), NULL, promo\_cost\*.25, 'N/A') "Catcost"  
FROM promotions;
- D. SELECT promo\_id, DECODE(promo\_cost, >10000, 'High', <10000, 'Low') "Range"  
FROM promotions;

**Answer: A, B**

**Question: 20**

Evaluate the following SQL statement:  
SQL> SELECT cust\_id, cust\_last\_name  
FROM customers  
WHERE cust\_credit\_limit IN  
(select cust\_credit\_limit  
FROM customers  
WHERE cust\_city ='Singapore');

Which statement is true regarding the above query if one of the values generated by the subquery is NULL?

- A. It produces an error.
- B. It executes but returns no rows.
- C. It generates output for NULL as well as the other values produced by the subquery.
- D. It ignores the NULL value and generates output for the other values produced by the subquery.

**Answer: C**

**Question: 21**

View the Exhibit and examine the structure of the PRODUCTS table.

You need to generate a report in the following format:

CATEGORIES

---

5MP Digital Photo Camera's category is Photo

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

Y Box's category is Electronics

Envoy Ambassador's category is Hardware

Which two queries would give the required output? (Choose two.)

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)

- A. SELECT prod\_name || q'"s category is ' || prod\_category CATEGORIES FROM products;
- B. SELECT prod\_name || q'[s ]'category is ' || prod\_category CATEGORIES FROM products;
- C. SELECT prod\_name || q'\s\ ' category is ' || prod\_category CATEGORIES FROM products;
- D. SELECT prod\_name || q'<s >' 'category is ' || prod\_category CATEGORIES FROM products;

**Answer: C, D**

### Question: 22

The PART\_CODE column in the SPARES table contains the following list of values:

PART\_CODE

-----  
A%\_WQ123  
A%BWQ123  
AB\_WQ123

Evaluate the following query:

SQL> SELECT part\_code  
FROM spares

WHERE part\_code LIKE '%\%\_WQ12%' ESCAPE '\';

Which statement is true regarding the outcome of the above query?

- A. It produces an error.
- B. It displays all values.
- C. It displays only the values A%\_WQ123 and AB\_WQ123.
- D. It displays only the values A%\_WQ123 and A%BWQ123.
- E. It displays only the values A%BWQ123 and AB\_WQ123.

**Answer: D**

### Question: 23

When does a transaction complete? (Choose all that apply.)

- A. when a DELETE statement is executed
- B. when a ROLLBACK command is executed
- C. when a PL/SQL anonymous block is executed
- D. when a data definition language (DDL) statement is executed

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

E. when a TRUNCATE statement is executed after the pending transaction

**Answer: B, D, E**

**Question: 24**

Which two statements are true regarding single row functions? (Choose two.)

- A. They accept only a single argument.
- B. They can be nested only to two levels.
- C. Arguments can only be column values or constants.
- D. They always return a single result row for every row of a queried table.
- E. They can return a data type value different from the one that is referenced.

**Answer: D, E**

**Question: 25**

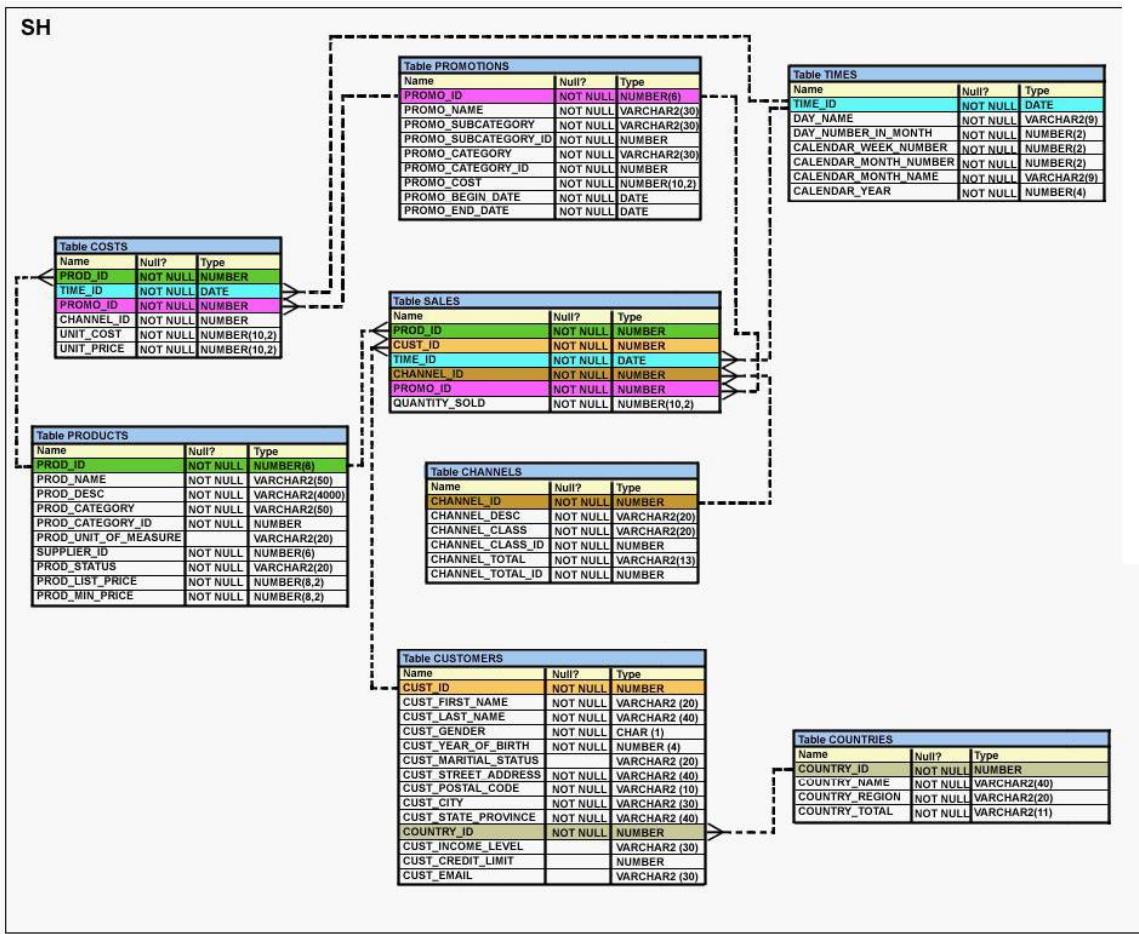
View the Exhibit to examine the description for the SALES and PRODUCTS tables.

You want to create a SALE\_PROD view by executing the following SQL statement:

```
CREATE VIEW sale_prod
AS SELECT p.prod_id, cust_id, SUM(quantity_sold) "Quantity", SUM(prod_list_price) "Price"
FROM products p, sales s
WHERE p.prod_id=s.prod_id
GROUP BY p.prod_id, cust_id;
```

Which statement is true regarding the execution of the above statement?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>	
<b>Exam Type:</b>	<b>Oracle</b>	
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b> <b>114</b>



- A. The view will be created and you can perform DML operations on the view.
- B. The view will be created but no DML operations will be allowed on the view.
- C. The view will not be created because the join statements are not allowed for creating a view.
- D. The view will not be created because the GROUP BY clause is not allowed for creating a view.

**Answer: B**

### Question: 26

View the Exhibit and examine the description for the CUSTOMERS table.

You want to update the CUST\_INCOME\_LEVEL and CUST\_CREDIT\_LIMIT columns for the customer with the CUST\_ID 2360. You want the value for the CUST\_INCOME\_LEVEL to have the same value as that of the customer with the CUST\_ID 2560 and the CUST\_CREDIT\_LIMIT to have the same value as that of the customer with CUST\_ID 2566.

Which UPDATE statement will accomplish the task?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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. UPDATE customers

```
SET cust_income_level = (SELECT cust_income_level
FROM customers
WHERE cust_id = 2560),
cust_credit_limit = (SELECT cust_credit_limit
FROM customers
WHERE cust_id = 2566)
WHERE cust_id=2360;
```

B. UPDATE customers

```
SET (cust_income_level,cust_credit_limit) = (SELECT
cust_income_level, cust_credit_limit
FROM customers
WHERE cust_id=2560 OR cust_id=2566)
WHERE cust_id=2360;
```

C. UPDATE customers

```
SET (cust_income_level,cust_credit_limit) = (SELECT
cust_income_level, cust_credit_limit
FROM customers
WHERE cust_id IN(2560, 2566)
WHERE cust_id=2360;
```

D. UPDATE customers

```
SET (cust_income_level,cust_credit_limit) = (SELECT
cust_income_level, cust_credit_limit
FROM customers
WHERE cust_id=2560 AND cust_id=2566)
WHERE cust_id=2360;
```

**Answer: A**

**Question: 27**

View the Exhibit and examine the structure of the PROMOTIONS table.

Evaluate the following SQL statement:

```
SQL>SELECT promo_category, AVG(promo_cost) Avg_Cost, AVG(promo_cost)*.25
Avg_Overhead
FROM promotions
WHERE UPPER(promo_category) IN ('TV', 'INTERNET','POST') GROUP BY Avg_Cost ORDER
BY
Avg_Overhead; The above query generates an error on execution. Which clause in the above
SQL statement causes the error?
```

Exam Name:	Oracle Database: SQL Fundamentals		
Exam Type:	Oracle		
Exam Code:	1Z0-051	Total Questions	114

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. WHERE
- B. SELECT
- C. GROUP BY
- D. ORDER BY

**Answer: C**

**Question: 28**

Which statement is true regarding subqueries?

- A. The LIKE operator cannot be used with single-row subqueries.
- B. The NOT IN operator is equivalent to IS NULL with single-row subqueries.
- C. =ANY and =ALL operators have the same functionality in multiple-row subqueries.
- D. The NOT operator can be used with IN, ANY, and ALL operators in multiple-row subqueries.

**Answer: D**

**Question: 29**

View the Exhibits and examine the structures of the PRODUCTS, SALES, and CUSTOMERS tables.

You need to generate a report that gives details of the customer's last name, name of the product, and the quantity sold for all customers in 'Tokyo'.

Which two queries give the required result? (Choose two.)

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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)

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)

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 c.cust\_last\_name,p.prod\_name, s.quantity\_sold  
 FROM sales s JOIN products p  
 USING(prod\_id)  
 JOIN customers c  
 USING(cust\_id)  
 WHERE c.cust\_city='Tokyo';
- B. SELECT c.cust\_last\_name, p.prod\_name, s.quantity\_sold  
 FROM products p JOIN sales s JOIN customers c  
 ON(p.prod\_id=s.prod\_id)  
 ON(s.cust\_id=c.cust\_id)  
 WHERE c.cust\_city='Tokyo';
- C. SELECT c.cust\_last\_name, p.prod\_name, s.quantity\_sold  
 FROM products p JOIN sales s  
 ON(p.prod\_id=s.prod\_id)  
 JOIN customers c  
 ON(s.cust\_id=c.cust\_id)  
 AND c.cust\_city='Tokyo';
- D. SELECT c.cust\_id,c.cust\_last\_name,p.prod\_id, p.prod\_name, s.quantity\_sold  
 FROM products p JOIN sales s  
 USING(prod\_id)  
 JOIN customers c

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

USING(cust\_id)  
WHERE c.cust\_city='Tokyo';

**Answer: A, C**

**Question: 30**

Which three SQL statements would display the value 1890.55 as \$1,890.55?  
(Choose three.)

- A. SELECT TO\_CHAR(1890.55,'\$0G000D00')  
FROM DUAL;
- B. SELECT TO\_CHAR(1890.55,'\$9,999V99')  
FROM DUAL;
- C. SELECT TO\_CHAR(1890.55,'\$99,999D99')  
FROM DUAL;
- D. SELECT TO\_CHAR(1890.55,'\$99G999D00')  
FROM DUAL;
- E. SELECT TO\_CHAR(1890.55,'\$99G999D99')  
FROM DUAL;

**Answer: A, D, E**

**Question: 31**

View the Exhibit and examine the structure of the CUSTOMERS table.  
Which statement would display the highest credit limit available in each income level in each city in the CUSTOMERS table?

Table CUSTOMERS		
Name	Null?	Type
<b>CUST_ID</b>	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)
<b>COUNTRY_ID</b>	NOT NULL	NUMBER
CUST_INCOME_LEVEL		VARCHAR2 (30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2 (30)

- A. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)  
FROM customers  
GROUP BY cust\_city, cust\_income\_level, cust\_credit\_limit;
- B. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)  
FROM customers  
GROUP BY cust\_city, cust\_income\_level;
- C. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)  
FROM customers  
GROUP BY cust\_credit\_limit, cust\_income\_level, cust\_city;
- D. SELECT cust\_city, cust\_income\_level, MAX(cust\_credit\_limit)  
FROM customers  
GROUP BY cust\_city, cust\_income\_level, MAX (cust\_credit\_limit);

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

**Answer: B**

**Question: 32**

Which CREATE TABLE statement is valid?

- A. CREATE TABLE ord\_details  
(ord\_no NUMBER(2) PRIMARY KEY,  
item\_no NUMBER(3) PRIMARY KEY,  
ord\_date DATE NOT NULL);
- B. CREATE TABLE ord\_details  
(ord\_no NUMBER(2) UNIQUE, NOT NULL,  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT SYSDATE NOT NULL);
- C. CREATE TABLE ord\_details  
(ord\_no NUMBER(2),  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT NOT NULL,  
CONSTRAINT ord\_uq UNIQUE (ord\_no),  
CONSTRAINT ord\_pk PRIMARY KEY (ord\_no));
- D. CREATE TABLE ord\_details  
(ord\_no NUMBER(2),  
item\_no NUMBER(3),  
ord\_date DATE DEFAULT SYSDATE NOT NULL,  
CONSTRAINT ord\_pk PRIMARY KEY (ord\_no, item\_no));

**Answer: D**

**Question: 33**

Evaluate the following SQL statement:

```
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;
```

Which ORDER BY clauses are valid for the above query? (Choose all that apply.)

- A. ORDER BY 2,1
- B. ORDER BY CUST\_NO
- C. ORDER BY 2,cust\_id
- D. ORDER BY "CUST\_NO"
- E. ORDER BY "Last Name"

**Answer: A, C, E**

**Question: 34**

Which is the valid CREATE TABLE statement?

- A. CREATE TABLE emp9## (emp\_no NUMBER(4));
- B. CREATE TABLE 9emp## (emp\_no NUMBER(4));
- C. CREATE TABLE emp\*123 (emp\_no NUMBER(4));
- D. CREATE TABLE emp9## (emp\_no NUMBER(4), date DATE);

**Answer: A**

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

**Question: 35**

The SQL statements executed in a user session are as follows:

```
SQL> CREATE TABLE product
  (PCODE NUMBER(2),
  PNAME VARCHAR2(10));

SQL> INSERT INTO product VALUES (1, 'pen');
SQL> INSERT INTO product VALUES (2,'pencil');
SQL> SAVEPOINT a;
SQL> UPDATE product SET PCODE = 10 WHERE PCODE = 1;
SQL> SAVEPOINT b;
SQL> DELETE FROM product WHERE PCODE = 2;
SQL> COMMIT; SQL> DELETE FROM product WHERE PCODE=10;
```

Which two statements describe the consequences of issuing the ROLLBACK TO SAVE POINT a command in the session? (Choose two.)

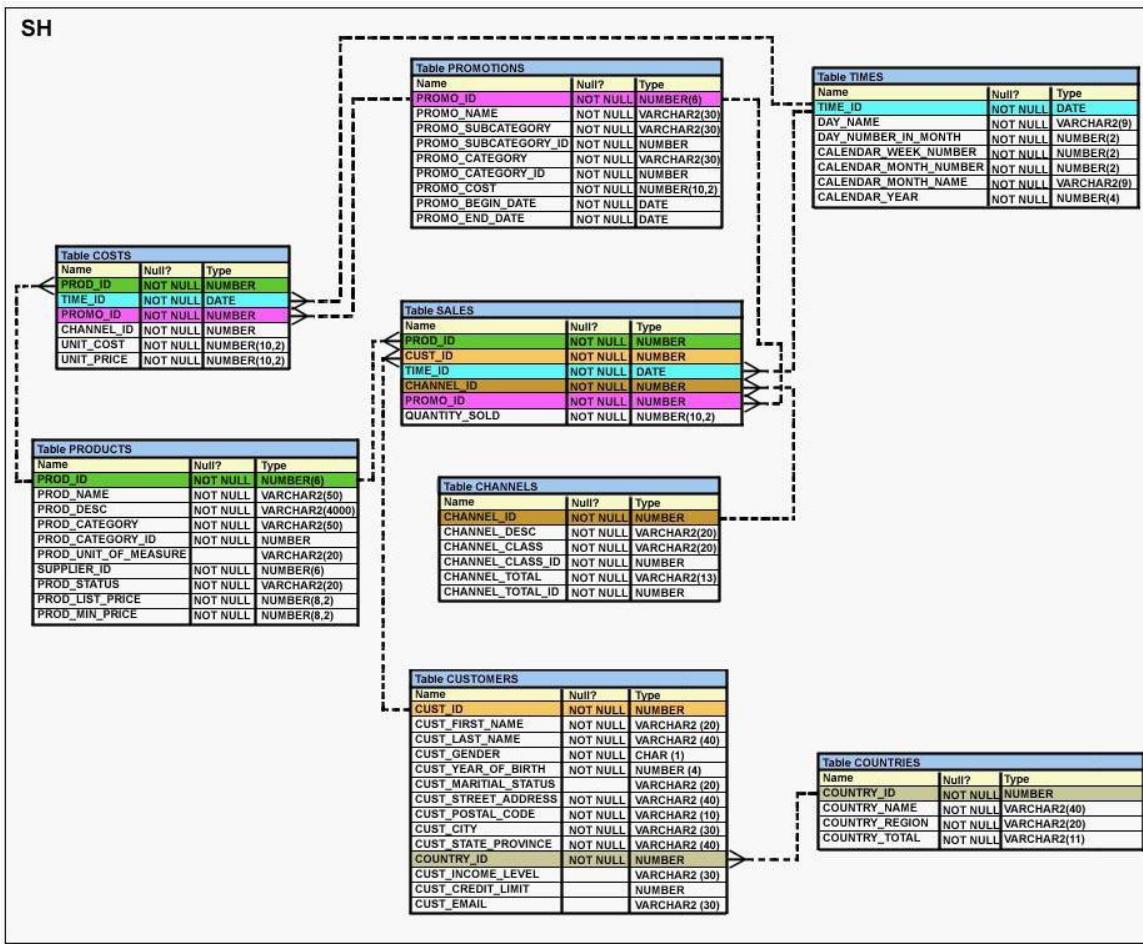
- A. The rollback generates an error.
- B. No SQL statements are rolled back.
- C. Only the DELETE statements are rolled back.
- D. Only the second DELETE statement is rolled back.
- E. Both the DELETE statements and the UPDATE statement are rolled back.

**Answer: A, B**

**Question: 36**

View the Exhibit and examine the description of SALES and PROMOTIONS tables. You want to delete rows from the SALES table, where the PROMO\_NAME column in the PROMOTIONS table has either blowout sale or everyday low price as values. Which DELETE statements are valid? (Choose all that apply.)

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>



#### A. DELETE

```

FROM sales
WHERE promo_id = (SELECT promo_id
                  FROM promotions
                  WHERE promo_name = 'blowout sale')
                  AND promo_id = (SELECT promo_id
                  FROM promotions
                  WHERE promo_name = 'everyday low price');
  
```

#### B. DELETE

```

FROM sales
WHERE promo_id = (SELECT promo_id
                  FROM promotions
                  WHERE promo_name = 'blowout sale')
                  OR promo_id = (SELECT promo_id
                  FROM promotions
                  WHERE promo_name = 'everyday low price');
  
```

#### C. DELETE

```

FROM sales
WHERE promo_id IN (SELECT promo_id
                  FROM promotions
                  WHERE promo_name = 'blowout sale'
                  OR promo_name = 'everyday low price');
  
```

#### D. DELETE

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

```

FROM sales
WHERE promo_id IN (SELECT promo_id
FROM promotions
WHERE promo_name IN ('blowout sale', 'everyday low price'));

```

**Answer: B, C, D**

**Question: 37**

View the Exhibit to examine the description for the SALES table.  
Which views can have all DML operations performed on it? (Choose all that apply.)

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)

- A. CREATE VIEW v3  
AS SELECT \* FROM SALES  
WHERE cust\_id = 2034  
WITH CHECK OPTION;
- B. CREATE VIEW v1  
AS SELECT \* FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
WITH CHECK OPTION;
- C. CREATE VIEW v2  
AS SELECT prod\_id, cust\_id, time\_id FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
WITH CHECK OPTION;
- D. CREATE VIEW v4  
AS SELECT prod\_id, cust\_id, SUM(quantity\_sold) FROM SALES  
WHERE time\_id <= SYSDATE - 2\*365  
GROUP BY prod\_id, cust\_id  
WITH CHECK OPTION;

**Answer: A, B**

**Question: 38**

View the Exhibit and examine the structure of ORDERS and CUSTOMERS tables.  
There is only one customer with the cust\_last\_name column having value Roberts. Which  
INSERT statement should be used to add a row into the ORDERS table for the customer whose  
CUST\_LAST\_NAME is Roberts and CREDIT\_LIMIT is 600?

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

<b>ORDERS</b>		
<b>Name</b>	<b>Null?</b>	<b>Type</b>
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)

<b>CUSTOMERS</b>		
<b>Name</b>	<b>Null?</b>	<b>Type</b>
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)

A. INSERT INTO orders

```
VALUES (1,'10-mar-2007', 'direct',
(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600), 1000);
```

B. INSERT INTO orders (order\_id,order\_date,order\_mode,

```
(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &customer_id, 1000);
```

C. INSERT INTO(SELECT o.order\_id, o.order\_date,o.order\_mode,c.customer\_id, o.order\_total

```
FROM orders o, customers c
WHERE o.customer_id = c.customer_id
AND c.cust_last_name='Roberts' ANDc.credit_limit=600 )
VALUES (1,'10-mar-2007', 'direct',(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600), 1000);
```

D. INSERT INTO orders (order\_id,order\_date,order\_mode,

```
(SELECT customer_id
FROM customers
WHERE cust_last_name='Roberts' AND
credit_limit=600),order_total)
VALUES(1,'10-mar-2007', 'direct', &customer_id, 1000);
```

**Answer: A**

**Question: 39**

Evaluate the following command:

```
CREATE TABLE employees
(employee_id NUMBER(2) PRIMARY KEY,
```

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

last\_name VARCHAR2(25) NOT NULL,  
 department\_id NUMBER(2)NOT NULL,  
 job\_id VARCHAR2(8),  
 salary NUMBER(10,2));

You issue the following command to create a view that displays the IDs and last names of the sales staff in the organization:

```
CREATE OR REPLACE VIEW sales_staff_vu AS
SELECT employee_id,
       last_name,job_id
  FROM employees
 WHERE job_id LIKE 'SA_%' WITH CHECK OPTION;
```

Which two statements are true regarding the above view? (Choose two.)

- A. It allows you to insert rows into the EMPLOYEES table.
- B. It allows you to delete details of the existing sales staff from the EMPLOYEES table.
- C. It allows you to update job IDs of the existing sales staff to any other job ID in the EMPLOYEES table.
- D. It allows you to insert IDs, last names, and job IDs of the sales staff from the view if it is used in multitable INSERT statements.

**Answer: B, D**

#### **Question: 40**

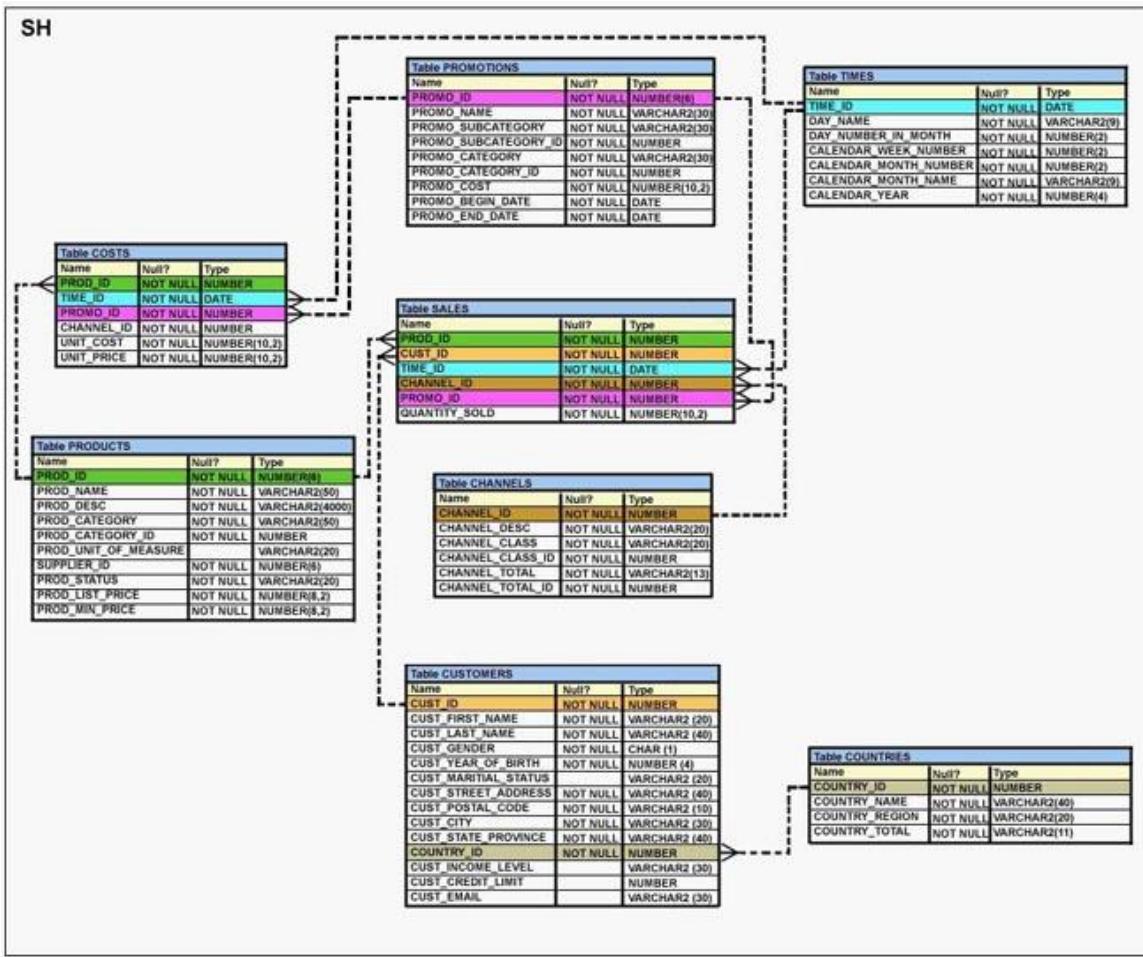
View the Exhibit and examine the structure of the SALES, CUSTOMERS, PRODUCTS, and TIMES tables. The PROD\_ID column is the foreign key in the SALES table, which references the PRODUCTS table. Similarly, the CUST\_ID and TIME\_ID columns are also foreign keys in the SALES table referencing the CUSTOMERS and TIMES tables, respectively.

Evaluate the following CREATE TABLE command:

```
CREATE TABLE new_sales(prod_id, cust_id, order_date DEFAULT SYSDATE)
AS SELECT prod_id, cust_id, time_id
  FROM sales;
```

Which statement is true regarding the above command?

Exam Name:	Oracle Database: SQL Fundamentals	
Exam Type:	Oracle	
Exam Code:	1Z0-051	Total Questions 114



- A. The NEW\_SALES table would not get created because the DEFAULT value cannot be specified in the column definition.
- B. The NEW\_SALES table would get created and all the NOT NULL constraints defined on the specified columns would be passed to the new table.
- C. The NEW\_SALES table would not get created because the column names in the CREATE TABLE command and the SELECT clause do not match.
- D. The NEW\_SALES table would get created and all the FOREIGN KEY constraints defined on the specified columns would be passed to the new table.

**Answer: B**

#### Question: 41

View the Exhibit and examine the structure of the CUSTOMERS table.

Evaluate the following SQL statement:

```
SQL> SELECT cust_city, COUNT(cust_last_name)
```

FROM customers

WHERE cust\_credit\_limit > 1000

GROUP BY cust\_city

HAVING AVG(cust\_credit\_limit) BETWEEN 5000 AND 6000;

Which statement is true regarding the outcome of the above query?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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. It executes successfully.
- B. It returns an error because the BETWEEN operator cannot be used in the HAVING clause.
- C. It returns an error because WHERE and HAVING clauses cannot be used in the same SELECT statement.
- D. It returns an error because WHERE and HAVING clauses cannot be used to apply conditions on the same column.

**Answer: A**

**Question: 42**

The following data exists in the PRODUCTS table:

PROD\_ID PROD\_LIST\_PRICE

-----  
123456 152525.99

You issue the following query:

```
SQL> SELECT RPAD((ROUND(prod_list_price)), 10,'*')
FROM products
```

WHERE prod\_id = 123456;

What would be the outcome?

- A. 152526\*\*\*\*
- B. \*\*152525.99
- C. 152525\*\*
- D. an error message

**Answer: A**

**Question: 43**

View the Exhibit and examine the structure of the PROMOTIONS table.

Examine the following two SQL statements:

Statement 1

```
SQL>SELECT promo_category,SUM(promo_cost)
FROM promotions
WHERE promo_end_date-promo_begin_date > 30
GROUP BY promo_category;
```

Statement 2

```
SQL>SELECT promo_category,sum(promo_cost)
FROM promotions
GROUP BY promo_category
HAVING MIN(promo_end_date-promo_begin_date)>30;
```

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

Which statement is true regarding the above two SQL statements?

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. statement 1 gives an error, statement 2 executes successfully
- B. statement 2 gives an error, statement 1 executes successfully
- C. statement 1 and statement 2 execute successfully and give the same output
- D. statement 1 and statement 2 execute successfully and give a different output

**Answer: D**

#### Question: 44

Examine the structure and data of the CUST\_TRANS table:

CUST\_TRANS

Name Null? Type

-----

CUSTNO NOT NULL CHAR(2)

TRANSDATE DATE

TRANSAMT NUMBER(6,2)

-----

CUSTNO TRANSDATE TRANSAMT

-----

11 01-JAN-07 1000

22 01-FEB-07 2000

33 01-MAR-07 3000

Dates are stored in the default date format dd-mon-rr in the CUST\_TRANS table.

Which three SQL statements would execute successfully? (Choose three.)

- A. SELECT transdate + '10' FROM cust\_trans;
- B. SELECT \* FROM cust\_trans WHERE transdate = '01-01-07';
- C. SELECT transamt FROM cust\_trans WHERE custno > '11';
- D. SELECT \* FROM cust\_trans WHERE transdate='01-JANUARY-07';
- E. SELECT custno + 'A' FROM cust\_trans WHERE transamt > 2000;

**Answer: A, C, D**

#### Question: 45

View the Exhibit and examine the structure of the PROMOTIONS table.

You have to generate a report that displays the promo name and start date for all promos that started after the last promo in the 'INTERNET' category.

Which query would give you the required output?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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 promotions  
WHERE promo\_begin\_date > ALL (SELECT MAX(promo\_begin\_date)  
FROM promotions )AND  
promo\_category = 'INTERNET';
- B. SELECT promo\_name, promo\_begin\_date FROM promotions  
WHERE promo\_begin\_date IN (SELECT promo\_begin\_date  
FROM promotions  
WHERE promo\_category='INTERNET');
- C. SELECT promo\_name, promo\_begin\_date FROM promotions  
WHERE promo\_begin\_date > ALL (SELECT promo\_begin\_date  
FROM promotions  
WHERE promo\_category = 'INTERNET');
- D. SELECT promo\_name, promo\_begin\_date FROM promotions  
WHERE promo\_begin\_date > ANY (SELECT promo\_begin\_date  
FROM promotions  
WHERE promo\_category = 'INTERNET');

**Answer: C**

**Question: 46**

View the Exhibit and examine the structure of the PRODUCTS table.

Evaluate the following query:

```
SQL> SELECT prod_name
  FROM products
 WHERE prod_id IN (SELECT prod_id FROM products
 WHERE prod_list_price =
 (SELECT MAX(prod_list_price)FROM products
 WHERE prod_list_price <
 (SELECT MAX(prod_list_price)FROM products)));
```

What would be the outcome of executing the above SQL statement?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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)

- A. It produces an error.
- B. It shows the names of all products in the table.
- C. It shows the names of products whose list price is the second highest in the table.
- D. It shows the names of all products whose list price is less than the maximum list price.

**Answer: C**

**Question: 47**

View the Exhibit and examine the data in the COSTS table.

You need to generate a report that displays the IDs of all products in the COSTS table whose unit price is at least 25% more than the unit cost. The details should be displayed in the descending order of 25% of the unit cost.

You issue the following query:

```
SQL>SELECT prod_id
  FROM costs
 WHERE unit_price >= unit_cost * 1.25
 ORDER BY unit_cost * 0.25 DESC;
```

Which statement is true regarding the above query?

COSTS			
PROD_ID	PROMO_ID	UNIT_COST	UNIT_PRICE
14	111	900	1129
15	333	875	1075
16	333	700	900
17	444	1000	1150

- A. It executes and produces the required result.
- B. It produces an error because an expression cannot be used in the ORDER BY clause.
- C. It produces an error because the DESC option cannot be used with an expression in the ORDER BY clause.
- D. It produces an error because the expression in the ORDER BY clause should also be specified in the SELECT clause.

**Answer: A**

**Question: 48**

Which three tasks can be performed using SQL functions built into Oracle Database? (Choose three.)

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- A. displaying a date in a nondefault format
- B. finding the number of characters in an expression
- C. substituting a character string in a text expression with a specified string
- D. combining more than two columns or expressions into a single column in the output

**Answer: A, B, C**

**Question: 49**

View the Exhibit and evaluate the structure and data in the CUST\_STATUS table.

You issue the following SQL statement:

```
SQL> SELECT custno, NVL2(NULLIF(amt_spent, credit_limit), 0, 1000)"BONUS"
FROM cust_status;
```

Which statement is true regarding the execution of the above query?

CUST_STATUS		
Name	Null?	Type
CUSTNO	NOT NULL	NUMBER(2)
AMT_SPENT		NUMBER(10,2)
CREDIT_LIMIT		NUMBER(10,2)
CUSTNO	AMT_SPENT	CREDIT_LIMIT
1	1000	1000
2	2000	2500
3		3000
4	3000	2800

- A. It produces an error because the AMT\_SPENT column contains a null value.
- B. It displays a bonus of 1000 for all customers whose AMT\_SPENT is less than CREDIT\_LIMIT.
- C. It displays a bonus of 1000 for all customers whose AMT\_SPENT equals CREDIT\_LIMIT, or AMT\_SPENT is null.
- D. It produces an error because the TO\_NUMBER function must be used to convert the result of the NULLIF function before it can be used by the NVL2 function.

**Answer: C**

**Question: 50**

Which two statements are true regarding subqueries? (Choose two.)

- A. A subquery can retrieve zero or more rows.
- B. Only two subqueries can be placed at one level.
- C. A subquery can be used only in SQL query statements.
- D. A subquery can appear on either side of a comparison operator.
- E. There is no limit on the number of subquery levels in the WHERE clause of a SELECT statement.

**Answer: A, D**

**Question: 51**

View the Exhibit and examine the data in the PRODUCTS table.

You need to display product names from the PRODUCTS table that belong to the 'Software/Other' category with minimum prices as either \$2000 or \$4000 and no unit of measure. You issue the following query:

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

SQL>SELECT prod\_name, prod\_category, prod\_min\_price  
 FROM products  
 WHERE prod\_category LIKE '%Other%' AND (prod\_min\_price = 2000 OR  
 prod\_min\_price = 4000) AND prod\_unit\_of\_measure <> " ";  
 Which statement is true regarding the above query?

<b>PRODUCTS</b>				
PROD_ID	PROD_NAME	PROD_CATEGORY	PROD_MIN_PRICE	PROD_UNIT_OF_MEASURE
101	Envoy 256MB - 40GB	Hardware	6000	Nos.
102	Y Box	Electronics	9000	
103	DVD-R Disc, 4.7 GB	Software/Other	2000	Nos.
104	Documentation Set - Spanish	Software/Other	4000	

- A. It executes successfully but returns no result.
- B. It executes successfully and returns the required result.
- C. It generates an error because the condition specified for PROD\_UNIT\_OF\_MEASURE is not valid.
- D. It generates an error because the condition specified for the PROD\_CATEGORY column is not valid.

**Answer: A**

#### **Question: 52**

You need to calculate the number of days from 1st January 2007 till date .

Dates are stored in the default format of dd-mon-rr.

Which two SQL statements would give the required output? (Choose two.)

- A. SELECT SYSDATE - '01-JAN-2007' FROM DUAL;
- B. SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;
- C. SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL;
- D. SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;
- E. SELECT TO\_DATE(SYSDATE, 'DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL;

**Answer: B, C**

#### **Question: 53**

You need to generate a list of all customer last names with their credit limits from the CUSTOMERS table. Those customers who do not have a credit limit should appear last in the list. Which two queries would achieve the required result? (Choose two.)

- A. SELECT cust\_last\_name, cust\_credit\_limit  
     FROM customers  
     ORDER BY cust\_credit\_limit DESC;
- B. SELECT cust\_last\_name, cust\_credit\_limit  
     FROM customers  
     ORDER BY cust\_credit\_limit;
- C. SELECT cust\_last\_name, cust\_credit\_limit  
     FROM customers

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

ORDER BY cust\_credit\_limit NULLS LAST;  
D. SELECT cust\_last\_name, cust\_credit\_limit  
FROM customers  
ORDER BY cust\_last\_name, cust\_credit\_limit NULLS LAST;

**Answer: B, C**

**Question: 54**

View the Exhibit and examine the structure of the ORD table.

Evaluate the following SQL statements that are executed in a user session in the specified order:

```
CREATE SEQUENCE ord_seq;
SELECT ord_seq.nextval
FROM dual;
INSERT INTO ord
VALUES (ord_seq.CURRVAL, '25-jan-2007',101);
UPDATE ord
SET ord_no= ord_seq.NEXTVAL
WHERE cust_id =101;
```

What would be the outcome of the above statements?

ORD		
Name	Null?	Type
ORD_NO	NOT NULL	NUMBER(2)
ORD_DATE		DATE
CUST_ID		NUMBER(4)

- A. All the statements would execute successfully and the ORD\_NO column would contain the value 2 for the CUST\_ID 101.
- B. The CREATE SEQUENCE command would not execute because the minimum value and maximum value for the sequence have not been specified.
- C. The CREATE SEQUENCE command would not execute because the starting value of the sequence and the increment value have not been specified.
- D. All the statements would execute successfully and the ORD\_NO column would have the value 20 for the CUST\_ID 101 because the default CACHE value is 20.

**Answer: A**

**Question: 55**

Examine the structure and data in the PRICE\_LIST table:

Name Null? Type

---

```
PROD_ID NOT NULL NUMBER(3)
PROD_PRICE VARCHAR2(10)
PROD_ID PROD_PRICE
```

---

```
100 $234.55
101 $6,509.75
102 $1,234
```

'in the same format as the PROD\_PRICE.

Which SQL statement would give the required result?

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- A. SELECT TO\_CHAR(prod\_price\* .25,'\$99,999.99') FROM PRICE\_LIST;
- B. SELECT TO\_CHAR(TO\_NUMBER(prod\_price)\* .25,'\$99,999.00') FROM PRICE\_LIST;
- C. SELECT TO\_CHAR(TO\_NUMBER(prod\_price,'\$99,999.99')\* .25,'\$99,999.00') FROM PRICE\_LIST;
- D. SELECT TO\_NUMBER(TO\_NUMBER(prod\_price,'\$99,999.99')\* .25,'\$99,999.00') FROM PRICE\_LIST;

**Answer: C**

**Question: 56**

View the Exhibit for the structure of the STUDENT and FACULTY tables.

You need to display the faculty name followed by the number of students handled by the faculty at the base location. Examine the following two SQL statements:

Statement 1

```
SQL>SELECT faculty_name,COUNT(student_id)
  FROM student JOIN faculty
    USING (faculty_id, location_id)
   GROUP BY faculty_name;
```

Statement 2

```
SQL>SELECT faculty_name,COUNT(student_id)
  FROM student NATURAL JOIN faculty
   GROUP BY faculty_name;
```

Which statement is true regarding the outcome?

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)

- A. Only statement 1 executes successfully and gives the required result.
- B. Only statement 2 executes successfully and gives the required result.
- C. Both statements 1 and 2 execute successfully and give different results.
- D. Both statements 1 and 2 execute successfully and give the same required result.

**Answer: D**

**Question: 57**

Which statement is true regarding the COALESCE function?

- A. It can have a maximum of five expressions in a list.
- B. It returns the highest NOT NULL value in the list for all rows.
- C. It requires that all expressions in the list must be of the same data type.
- D. It requires that at least one of the expressions in the list must have a NOT NULL value.

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

**Answer: C**

**Question: 58**

View the Exhibit and examine the structure of the PRODUCTS table.

Using the PRODUCTS table, you issue the following query to generate the names, current list price, and discounted list price for all those products whose list price falls below \$10 after a discount of 25% is applied on it.

```
SQL>SELECT prod_name, prod_list_price,
prod_list_price - (prod_list_price * .25) "DISCOUNTED_PRICE"
FROM products
WHERE discounted_price < 10;
```

The query generates an error.  
What is the reason for the error?

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)

- A. The parenthesis should be added to enclose the entire expression.
- B. The double quotation marks should be removed from the column alias.
- C. The column alias should be replaced with the expression in the WHERE clause.
- D. The column alias should be put in uppercase and enclosed within double quotation marks in the WHERE clause.

**Answer: C**

**Question: 59**

Evaluate the following CREATE TABLE commands:

```
CREATE TABLE orders
(ord_no NUMBER(2) CONSTRAINT ord_pk PRIMARY KEY,
ord_date DATE,
cust_id NUMBER(4));
CREATE TABLE ord_items
(ord_no NUMBER(2),
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));
The above command fails when executed. What could be the reason?
```

- A. SYSDATE cannot be used with the CHECK constraint.
- B. The BETWEEN clause cannot be used for the CHECK constraint.
- C. The CHECK constraint cannot be placed on columns having the DATE data type.
- D. ORD\_NO and ITEM\_NO cannot be used as a composite primary key because ORD\_NO is also the FOREIGN KEY.

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

**Answer: A**

**Question: 60**

You created an ORDERS table with the following description:

Name Null Type

---

ORD\_ID NOT NULL NUMBER(2)  
 CUST\_ID NOT NULL NUMBER(3)  
 ORD\_DATE NOT NULL DATE  
 ORD\_AMOUNT NOT NULL NUMBER (10,2)

You inserted some rows in the table. After some time, you want to alter the table by creating the PRIMARY KEY constraint on the ORD\_ID column.

Which statement is true in this scenario?

- A. You cannot have two constraints on one column.
- B. You cannot add a primary key constraint if data exists in the column.
- C. The primary key constraint can be created only at the time of table creation.
- D. You can add the primary key constraint even if data exists, provided that there are no duplicate values.

**Answer: D**

**Question: 61**

View the Exhibit and examine the structure of ORD and ORD\_ITEMS tables.

The ORD\_NO column is PRIMARY KEY in the ORD table and the ORD\_NO and ITEM\_NO columns are composite PRIMARY KEY in the ORD\_ITEMS table.

Which two CREATE INDEX statements are valid? (Choose two.)

ORD		
Name	Null?	Type
ORD_NO	NOT NULL	NUMBER (2)
ORD_DATE		DATE
CUST_ID		NUMBER (4)

ORD_ITEMS		
Name	Null?	Type
ORD_NO	NOT NULL	NUMBER (2)
ITEM_NO	NOT NULL	NUMBER (3)
QTY		NUMBER (8,2)

- A. CREATE INDEX ord\_idx1  
ON ord(ord\_no);
- B. CREATE INDEX ord\_idx2  
ON ord\_items(ord\_no);
- C. CREATE INDEX ord\_idx3  
ON ord\_items(item\_no);
- D. CREATE INDEX ord\_idx4  
ON ord,ord\_items(ord\_no, ord\_date,qty);

**Answer: B, C**

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

**Question: 62**

View the Exhibit and examine the structure of the PROMOTIONS table. Using the PROMOTIONS table, you need to find out the names and cost of all the promos done on 'TV' and 'internet' that ended in the time interval 15th March '00 to 15th October '00.

Which two queries would give the required result? (Choose two.)

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\_cost  
FROM promotions  
WHERE promo\_category IN ('TV', 'internet') AND  
promo\_end\_date BETWEEN '15-MAR-00' AND '15-OCT-00';
- B. SELECT promo\_name, promo\_cost  
FROM promotions  
WHERE promo\_category = 'TV' OR promo\_category ='internet' AND  
promo\_end\_date >='15-MAR-00' OR promo\_end\_date <='15-OCT-00';
- C. SELECT promo\_name, promo\_cost  
FROM promotions WHERE (promo\_category BETWEEN 'TV' AND 'internet') AND  
(promo\_end\_date IN ('15-MAR-00','15-OCT-00'));
- D. SELECT promo\_name, promo\_cost  
FROM promotions WHERE (promo\_category = 'TV' OR promo\_category ='internet') AND  
(promo\_end\_date >='15-MAR-00' AND promo\_end\_date <='15-OCT-00');

**Answer: A, D**

**Question: 63**

You need to display the date 11-oct-2007 in words as 'Eleventh of October, Two Thousand Seven'. Which SQL statement would give the required result?

- A. SELECT TO\_CHAR('11-oct-2007', 'fmDdsph "of" Month, Year')  
FROM DUAL;
- B. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdsph of month, year')  
FROM DUAL;
- C. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdthsp "of" Month, Year')  
FROM DUAL;
- D. SELECT TO\_DATE(TO\_CHAR('11-oct-2007','fmDdsph "of" Month, Year'))  
FROM DUAL;

**Answer: C**

**Question: 64**

You need to create a table for a banking application. One of the columns in the table has the following requirements:

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

- 1) You want a column in the table to store the duration of the credit period.
- 2) The data in the column should be stored in a format such that it can be easily added and subtracted with DATE data type without using conversion functions.
- 3) The maximum period of the credit provision in the application is 30 days.
- 4) The interest has to be calculated for the number of days an individual has taken a credit for.

Which data type would you use for such a column in the table?

- A. DATE
- B. NUMBER
- C. TIMESTAMP
- D. INTERVAL DAY TO SECOND
- E. INTERVAL YEAR TO MONTH

**Answer: D**

**Question: 65**

View the Exhibit and examine the data in the CUSTOMERS table.

Evaluate the following query:

```
SQL> SELECT cust_name AS "NAME", cust_credit_limit/2 AS MIDPOINT,
MIDPOINT+100 AS "MAX LOWER LIMIT"
FROM customers;
```

The above query produces an error on execution.

What is the reason for the error?

CUSTOMERS			
CUST_NO	CUST_NAME	CUST_CITY	CUST_CREDIT_LIMIT
101	KING	NEW YORK	100000
102	GREEN	BOSTON	150000
103	SCOTT	LONDON	
104	SMITH	BOSTON	

- A. An alias cannot be used in an expression.
- B. The alias NAME should not be enclosed within double quotation marks.
- C. The MIDPOINT+100 expression gives an error because CUST\_CREDIT\_LIMIT contains NULL values.
- D. The alias MIDPOINT should be enclosed within double quotation marks for the CUST\_CREDIT\_LIMIT/2 expression.

**Answer: A**

**Question: 66**

Which statements are correct regarding indexes? (Choose all that apply.)

- A. When a table is dropped, the corresponding indexes are automatically dropped.
- B. A FOREIGN KEY constraint on a column in a table automatically creates a nonunique index.
- C. A nondeferrable PRIMARY KEY or UNIQUE KEY constraint in a table automatically creates a unique index.
- D. For each data manipulation language (DML) operation performed, the corresponding indexes are automatically updated.

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

**Answer: A, C, D**

**Question: 67**

View the Exhibit and examine the structure of the PRODUCTS table.

All products have a list price.

You issue the following command to display the total price of each product after a discount of 25% and a

tax of 15% are applied on it. Freight charges of \$100 have to be applied to all the products.

```
SQL>SELECT prod_name, prod_list_price -(prod_list_price*(25/100))
```

```
+ (prod_list_price -(prod_list_price*(25/100)) * (15/100)) + 100
```

```
AS "TOTAL PRICE"
```

```
FROM products;
```

What would be the outcome if all the parentheses are removed from the above statement?

Table PRODUCTS		
Name	Null?	Type
<b>PROD_ID</b>	<b>NOT NULL</b>	<b>NUMBER(6)</b>
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)

- A. It produces a syntax error.
- B. The result remains unchanged.
- C. The total price value would be lower than the correct value.
- D. The total price value would be higher than the correct value.

**Answer: B**

**Question: 68**

View the Exhibit and examine the structure of the CUSTOMERS table.

Using the CUSTOMERS table, you need to generate a report that shows an increase in the credit limit by 15% for all customers. Customers whose credit limit has not been entered should have the message "Not Available" displayed.

Which SQL statement would produce the required result?

Table CUSTOMERS		
Name	Null?	Type
<b>CUST_ID</b>	<b>NOT NULL</b>	<b>NUMBER</b>
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)
<b>COUNTRY_ID</b>	<b>NOT NULL</b>	<b>NUMBER</b>
CUST_INCOME_LEVEL		VARCHAR2(30)
CUST_CREDIT_LIMIT		NUMBER
CUST_EMAIL		VARCHAR2(30)

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- A. SELECT NVL(cust\_credit\_limit,'Not Available')\*.15 "NEW CREDIT"  
FROM customers;
- B. SELECT NVL(cust\_credit\_limit\*.15,'Not Available') "NEW CREDIT"  
FROM customers;
- C. SELECT TO\_CHAR(NVL(cust\_credit\_limit\*.15,'Not Available')) "NEW CREDIT"  
FROM customers;
- D. SELECT NVL(TO\_CHAR(cust\_credit\_limit\*.15),'Not Available') "NEW CREDIT"  
FROM customers;

**Answer: D**

**Question: 69**

Which two statements are true about sequences created in a single instance database? (Choose two.)

- A. The numbers generated by a sequence can be used only for one table.
- B. DELETE <sequencename> would remove a sequence from the database.
- C. CURRVAL is used to refer to the last sequence number that has been generated.
- D. When the MAXVALUE limit for a sequence is reached, you can increase the MAXVALUE limit by using the ALTER SEQUENCE statement.
- E. When a database instance shuts down abnormally, the sequence numbers that have been cached but not used would be available once again when the database instance is restarted.

**Answer: C, D**

**Question: 70**

Which two statements are true regarding single row functions? (Choose two.)

- A. They accept only a single argument.
- B. They can be nested only to two levels.
- C. Arguments can only be column values or constants.
- D. They always return a single result row for every row of a queried table.
- E. They can return a data type value different from the one that is referenced.

**Answer: D, E**

**Question: 71**

View the Exhibit and examine the structure of the PROMOTIONS table.

Using the PROMOTIONS table, you need to find out the average cost for all promos in the ranges \$0-2000 and \$2000-5000 in category A.

You issue the following SQL statement:

```
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;
```

What would be the outcome?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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. It executes successfully and gives the required result.
- B. It generates an error because NULL cannot be specified as a return value.
- C. It generates an error because CASE cannot be used with group functions.
- D. It generates an error because multiple conditions cannot be specified for the WHEN clause.

**Answer: A**

**Question: 72**

Which two statements are true regarding constraints? (Choose two.)

- A. A foreign key cannot contain NULL values.
- B. A column with the UNIQUE constraint can contain NULL values.
- C. A constraint is enforced only for the INSERT operation on a table.
- D. A constraint can be disabled even if the constraint column contains data.
- E. All constraints can be defined at the column level as well as the table level.

**Answer: B, D**

**Question: 73**

Evaluate the following query:

```
SQL> SELECT promo_name || q'{s start date was }' || promo_begin_date
AS "Promotion Launches"
FROM promotions;
```

What would be the outcome of the above query?

- A. It produces an error because flower braces have been used.
- B. It produces an error because the data types are not matching.
- C. It executes successfully and introduces an 's at the end of each promo\_name in the output.
- D. It executes successfully and displays the literal " {s start date was } " for each row in the output.

**Answer: C**

**Question: 74**

View the Exhibit and examine the structure of the PROMOTIONS table.

Examine the following two SQL statements:

Statement 1

```
SQL>SELECT promo_category,SUM(promo_cost)
FROM promotions
WHERE promo_end_date-promo_begin_date > 30
GROUP BY promo_category;
```

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

Statement 2

```
SQL>SELECT promo_category,sum(promo_cost)
  FROM promotions
 GROUP BY promo_category
 HAVING MIN(promo_end_date-promo_begin_date)>30;
Which statement is true regarding the above two SQL statements?
```

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. statement 1 gives an error, statement 2 executes successfully
- B. statement 2 gives an error, statement 1 executes successfully
- C. statement 1 and statement 2 execute successfully and give the same output
- D. statement 1 and statement 2 execute successfully and give a different output

**Answer: D**

**Question: 75**

Evaluate the following query:

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

What is the correct output of the above query?

- A. +25-00 , +54-02, +00 11:12:10.123457
- B. +00-300, +54-02, +00 11:12:10.123457
- C. +25-00 , +00-650, +00 11:12:10.123457
- D. +00-300 , +00-650, +00 11:12:10.123457

**Answer: A**

**Question: 76**

You issued the following command to drop the PRODUCTS table:

```
SQL> DROP TABLE products;
```

What is the implication of this command? (Choose all that apply.)

- A. All data along with the table structure is deleted.
- B. The pending transaction in the session is committed.
- C. All indexes on the table will remain but they are invalidated.
- D. All views and synonyms will remain but they are invalidated.
- E. All data in the table are deleted but the table structure will remain.

**Answer: A, B, D**

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

### Question: 77

View the Exhibit and examine the structure of the CUSTOMERS table.

NEW\_CUSTOMERS is a new table with the columns CUST\_ID, CUST\_NAME and CUST\_CITY that have the same data types and size as the corresponding columns in the CUSTOMERS table.

Evaluate the following INSERT statement:

```
INSERT INTO new_customers (cust_id, cust_name, cust_city) VALUES(SELECT
cust_id,cust_first_name||' '||cust_last_name,cust_city
FROM customers
WHERE cust_id > 23004);
```

The INSERT statement fails when executed. What could be the reason?

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. The VALUES clause cannot be used in an INSERT with a subquery.
- B. Column names in the NEW\_CUSTOMERS and CUSTOMERS tables do not match.
- C. The WHERE clause cannot be used in a subquery embedded in an INSERT statement.
- D. The total number of columns in the NEW\_CUSTOMERS table does not match the total number of columns in the CUSTOMERS table.

### Answer: A

### Question: 78

View the Exhibit and examine the structure of the PROMOTIONS table.

Using the PROMOTIONS table, you need to display the names of all promos done after January 1, 2001, starting with the latest promo.

Which query would give the required result? (Choose all that apply.)

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

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- FROM promotions  
WHERE promo\_begin\_date > '01-JAN-01'  
ORDER BY 2 DESC;
- B. SELECT promo\_name, promo\_begin\_date  
FROM promotions  
WHERE promo\_begin\_date > '01-JAN-01'  
ORDER BY promo\_name DESC;
- C. SELECT promo\_name, promo\_begin\_date  
FROM promotions  
WHERE promo\_begin\_date > '01-JAN-01'  
ORDER BY 1 DESC;
- D. SELECT promo\_name, promo\_begin\_date "START DATE"  
FROM promotions  
WHERE promo\_begin\_date > '01-JAN-01'  
ORDER BY "START DATE" DESC;

**Answer: A, D**

**Question: 79**

Evaluate the following SQL commands:

```
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));
The command to create a table fails. Identify the reason for the SQL statement failure? (Choose all that apply.)
```

- A. You cannot use SYSDATE in the condition of a CHECK constraint.
- B. You cannot use the BETWEEN clause in the condition of a CHECK constraint.
- C. You cannot use the NEXTVAL sequence value as a DEFAULT value for a column.
- D. You cannot use ORD\_NO and ITEM\_NO columns as a composite primary key because ORD\_NO is also the FOREIGN KEY.

**Answer: A, C**

**Question: 80**

View the Exhibit and examine the structure and data in the INVOICE table.

Which two SQL statements would execute successfully? (Choose two.)

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

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	A1Q	1000
2	01-OCT-07	B1R	2000
3	01-FEB-07		3000

- A. SELECT AVG(inv\_date)  
FROM invoice;
- B. SELECT MAX(inv\_date),MIN(cust\_id)  
FROM invoice;
- C. SELECT MAX(AVG(SYSDATE - inv\_date))  
FROM invoice;
- D. SELECT AVG( inv\_date - SYSDATE), AVG(inv\_amt)  
FROM invoice;

**Answer: B, D**

#### **Question: 81**

Which arithmetic operations can be performed on a column by using a SQL function that is built into Oracle database? (Choose three.)

- A. addition
- B. subtraction
- C. raising to a power
- D. finding the quotient
- E. finding the lowest value

**Answer: A, C, E**

#### **Question: 82**

View the Exhibit and examine the structure of the PROMOTIONS table.

You need to generate a report of all promos from the PROMOTIONS table based on the following conditions:

1. The promo name should not begin with 'T' or 'N'.
2. The promo should cost more than \$20000.
3. The promo should have ended after 1st January 2001.

Which WHERE clause would give the required result?

Exam Name:	Oracle Database: SQL Fundamentals		
Exam Type:	Oracle		
Exam Code:	1Z0-051	Total Questions	114

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. WHERE promo\_name NOT LIKE 'T%' OR promo\_name NOT LIKE 'N%' AND promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01'
- B. WHERE (promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%')OR promo\_cost > 20000 OR promo\_end\_date > '1-JAN-01'
- C. WHERE promo\_name NOT LIKE 'T%' AND promo\_name NOT LIKE 'N%' AND promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01'
- D. WHERE (promo\_name NOT LIKE '%T%' OR promo\_name NOT LIKE '%N%') AND(promo\_cost > 20000 AND promo\_end\_date > '1-JAN-01')

**Answer: C**

### Question: 83

View the Exhibit and examine the structure of the SALES table.

The following query is written to retrieve all those product IDs from the SALES table that have more than 55000 sold and have been ordered more than 10 times.

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

Which statement is true regarding this SQL statement?

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)

- A. It executes successfully and generates the required result.
- B. It produces an error because COUNT(\*) should be specified in the SELECT clause also.
- C. It produces an error because COUNT(\*) should be only in the HAVING clause and not in the WHERE clause.
- D. It executes successfully but produces no result because COUNT(prod\_id) should be used instead of COUNT(\*) .

**Answer: C**

### Question: 84

You need to create a table with the following column specifications:

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

1. Employee ID (numeric data type) for each employee
2. Employee Name (character data type) that stores the employee name
3. Hire date, which stores the date of joining the organization for each employee
4. Status (character data type), that contains the value 'ACTIVE' if no data is entered
5. Resume (character large object [CLOB] data type), which contains the resume submitted by the employee

Which is the correct syntax to create this table?

- A. CREATE TABLE EMP\_1  
(emp\_id NUMBER(4),  
emp\_name VARCHAR2(25),  
start\_date DATE,  
e\_status VARCHAR2(10) DEFAULT 'ACTIVE',  
resume CLOB(200));
- B. CREATE TABLE 1\_EMP  
(emp\_id NUMBER(4),  
emp\_name VARCHAR2(25),  
start\_date DATE,  
emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',  
resume CLOB);
- C. CREATE TABLE EMP\_1  
(emp\_id NUMBER(4),  
emp\_name VARCHAR2(25),  
start\_date DATE,  
emp\_status VARCHAR2(10) DEFAULT "ACTIVE",  
resume CLOB);
- D. CREATE TABLE EMP\_1  
(emp\_id NUMBER,  
emp\_name VARCHAR2(25),  
start\_date DATE,  
emp\_status VARCHAR2(10) DEFAULT 'ACTIVE',  
resume CLOB);

**Answer: D**

**Question: 85**

Which statement is true regarding the default behavior of the ORDER BY clause?

- A. In a character sort, the values are case-sensitive.
- B. NULL values are not considered at all by the sort operation.
- C. Only those columns that are specified in the SELECT list can be used in the ORDER BY clause.
- D. Numeric values are displayed from the maximum to the minimum value if they have decimal positions.

**Answer: A**

**Question: 86**

You need to extract details of those products in the SALES table where the PROD\_ID column contains the string '\_D123'. Which WHERE clause could be used in the SELECT statement to get the required output?

- A. WHERE prod\_id LIKE '%\_D123%' ESCAPE '\_'
- B. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\'

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- C. WHERE prod\_id LIKE '%\_D123%' ESCAPE '%\_'  
 D. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\\_'

**Answer: B**

**Question: 87**

Evaluate the following DELETE statement:

DELETE FROM sales;

There are no other uncommitted transactions on the SALES table.

Which statement is true about the DELETE statement?

- A. It would not remove the rows if the table has a primary key.  
 B. It removes all the rows as well as the structure of the table.  
 C. It removes all the rows in the table and deleted rows can be rolled back.  
 D. It removes all the rows in the table and deleted rows cannot be rolled back.

**Answer: C**

**Question: 88**

Which three statements are true about multiple-row subqueries? (Choose three.)

- A. They can contain a subquery within a subquery.  
 B. They can return multiple columns as well as rows.  
 C. They cannot contain a subquery within a subquery.  
 D. They can return only one column but multiple rows.  
 E. They can contain group functions and GROUP BY and HAVING clauses.  
 F. They can contain group functions and the GROUP BY clause, but not the HAVING clause.

**Answer: A, B, E**

**Question: 89**

Examine the description of the EMP\_DETAILS table given below:

NAME NULL TYPE

---

EMP\_ID NOT NULL NUMBER  
 EMP\_NAME NOT NULL VARCHAR2 (40)  
 EMP\_IMAGE LONG

Which two statements are true regarding SQL statements that can be executed on the EMP\_DETAIL table? (Choose two.)

- A. An EMP\_IMAGE column can be included in the GROUP BY clause.  
 B. An EMP\_IMAGE column cannot be included in the ORDER BY clause.  
 C. You cannot add a new column to the table with LONG as the data type.  
 D. You can alter the table to include the NOT NULL constraint on the EMP\_IMAGE column.

**Answer: B, C**

**Question: 90**

View the Exhibits and examine PRODUCTS and SALES tables.

You issue the following query to display product name and the number of times the product has been sold:

```
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
```

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

ON i.prod\_id = p.prod\_id;  
 What happens when the above statement is executed?



The image shows two separate windows side-by-side. The top window is titled "Exhibit\_Products" and displays the structure of the "PRODUCTS" table. The bottom window is titled "Exhibit\_Sales" and displays the structure of the "SALES" table. Both windows have standard window controls (minimize, maximize, close) at the top right.

**Table PRODUCTS**

Name	Null?	Type
<b>PROD_ID</b>	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)

**Table SALES**

Name	Null?	Type
<b>PROD_ID</b>	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)

- A. The statement executes successfully and produces the required output.
- B. The statement produces an error because ITEM\_CNT cannot be displayed in the outer query.
- C. The statement produces an error because a subquery in the FROM clause and outer-joins cannot be used together.
- D. The statement produces an error because the GROUP BY clause cannot be used in a subquery in the FROM clause.

**Answer: A**

#### Question: 91

Which two statements are true regarding the ORDER BY clause? (Choose two.)

- A. It is executed first in the query execution.
- B. It must be the last clause in the SELECT statement.
- C. It cannot be used in a SELECT statement containing a HAVING clause.
- D. You cannot specify a column name followed by an expression in this clause.
- E. You can specify a combination of numeric positions and column names in this clause.

**Answer: B, E**

#### Question: 92

Evaluate the following SQL statement:

```
SQL> SELECT promo_id, promo_category
  FROM promotions
 WHERE promo_category = 'Internet' ORDER BY 2 DESC
 UNION
  SELECT promo_id, promo_category
  FROM promotions
```

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

```

WHERE promo_category = 'TV'
UNION
SELECT promo_id, promo_category
FROM promotions
WHERE promo_category ='Radio';
Which statement is true regarding the outcome of the above query?

```

- A. It executes successfully and displays rows in the descending order of PROMO\_CATEGORY.
- B. It produces an error because positional notation cannot be used in the ORDER BY clause with SET operators.
- C. It executes successfully but ignores the ORDER BY clause because it is not located at the end of the compound statement.
- D. It produces an error because the ORDER BY clause should appear only at the end of a compound query—that is, with the last SELECT statement.

**Answer: D**

**Question: 93**

View the Exhibit; examine the structure of the PROMOTIONS table. Each promotion has a duration of at least seven days.

Your manager has asked you to generate a report, which provides the weekly cost for each promotion done to date.

Which query would achieve the required result?

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_cost/promo_end_date-promo_begin_date/7  
FROM promotions;`
- B. `SELECT promo_name,(promo_cost/promo_end_date-promo_begin_date)/7  
FROM promotions;`
- C. `SELECT promo_name, promo_cost/(promo_end_date-promo_begin_date/7)  
FROM promotions;`
- D. `SELECT promo_name, promo_cost/((promo_end_date-promo_begin_date)/7)  
FROM promotions;`

**Answer: D**

**Question: 94**

View the Exhibit and examine the structure of the CUSTOMERS and CUST\_HISTORY tables.

The CUSTOMERS table contains the current location of all currently active customers. The CUST\_HISTORY table stores historical details relating to any changes in the location of all current as well as previous customers who are no longer active with the company. You need to find those customers who have never changed their address.

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

Which SET operator would you use to get the required output?

CUSTOMERS		
Name	Null?	Type
CUST_ID	NOT NULL	NUMBER (4)
CUST_NAME		VARCHAR2 (20)
CUST_ADDRESS		VARCHAR2 (30)
CUST_CITY		VARCHAR2 (20)

CUST_HISTORY		
Name	Null?	Type
CUST_ID	NOT NULL	NUMBER (4)
CUST_NAME		VARCHAR2 (20)
CUST_CITY		VARCHAR2 (20)
CHANGE_DATE		DATE

- A. MINUS
- B. UNION
- C. INTERSECT
- D. UNION ALL

**Answer: A**

**Question: 95**

View the Exhibit and examine the description for the CUSTOMERS table.

You want to update the CUST\_CREDIT\_LIMIT column to NULL for all the customers, where CUST\_INCOME\_LEVEL has NULL in the CUSTOMERS table. Which SQL statement will accomplish the task?

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. UPDATE customers  
SET cust\_credit\_limit = NULL  
WHERE CUST\_INCOME\_LEVEL = NULL;

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- B. UPDATE customers  
   SET cust\_credit\_limit = NULL  
   WHERE cust\_income\_level IS NULL;
- C. UPDATE customers  
   SET cust\_credit\_limit = TO\_NUMBER(NULL)  
   WHERE cust\_income\_level = TO\_NUMBER(NULL);
- D. UPDATE customers  
   SET cust\_credit\_limit = TO\_NUMBER(' ',9999)  
   WHERE cust\_income\_level IS NULL;

**Answer: B**

**Question: 96**

The CUSTOMERS table has the following structure:  
 Name Null? Type

---

```
CUST_ID NOT NULL NUMBER
CUST_FIRST_NAME NOT NULL VARCHAR2(20)
CUST_LAST_NAME NOT NULL VARCHAR2(30)
CUST_INCOME_LEVEL VARCHAR2(30)
CUST_CREDIT_LIMIT NUMBER
```

You need to write a query that does the following tasks:

1. Display the first name and tax amount of the customers. Tax is 5% of their credit limit.
2. Only those customers whose income level has a value should be considered.
3. Customers whose tax amount is null should not be considered.

Which statement accomplishes all the required tasks?

- A. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT  
   FROM customers  
   WHERE cust\_income\_level IS NOT NULL AND  
   tax\_amount IS NOT NULL;
- B. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT  
   FROM customers  
   WHERE cust\_income\_level IS NOT NULL AND  
   cust\_credit\_limit IS NOT NULL;
- C. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT  
   FROM customers  
   WHERE cust\_income\_level <> NULL AND  
   tax\_amount <> NULL;
- D. SELECT cust\_first\_name, cust\_credit\_limit \* .05 AS TAX\_AMOUNT  
   FROM customers  
   WHERE (cust\_income\_level,tax\_amount) IS NOT NULL;

**Answer: B**

**Question: 97**

Which two statements are true regarding the USING clause in table joins?  
 (Choose two.)

- A. It can be used to join a maximum of three tables.  
 B. It can be used to restrict the number of columns used in a NATURAL join.  
 C. It can be used to access data from tables through equijoins as well as nonequijoins.

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

D. It can be used to join tables that have columns with the same name and compatible data types.

**Answer: B, D**

**Question: 98**

View the Exhibit and examine the data in the PROMOTIONS table.

You need to display all promo categories that do not have 'discount' in their subcategory. Which two SQL statements give the required result? (Choose two.)

PROMOTIONS		
PROMO_ID	PROMO_CATEGORY	PROMO_SUBCATEGORY
506	magazine	discount
507	TV	general advt
508	newspaper	discount
509	post	general advt
510	post	discount
511	radio	general advt
512	newspaper	general advt
513	newspaper	discount
514	magazine	general advt
515	newspaper	discount
516	newspaper	general advt

- A. SELECT promo\_category  
FROM promotions  
MINUS  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory = 'discount';
- B. SELECT promo\_category  
FROM promotions  
INTERSECT  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory = 'discount';
- C. SELECT promo\_category  
FROM promotions  
MINUS  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory <> 'discount';
- D. SELECT promo\_category  
FROM promotions  
INTERSECT  
SELECT promo\_category  
FROM promotions  
WHERE promo\_subcategory <> 'discount';

**Answer: A, D**

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

**Question: 99**

View the Exhibit and examine the data in the PROJ\_TASK\_DETAILS table.

The PROJ\_TASK\_DETAILS table stores information about tasks involved in a project and the relation between them.

The BASED\_ON column indicates dependencies between tasks. Some tasks do not depend on the completion of any other tasks.

You need to generate a report showing all task IDs, the corresponding task ID they are dependent on, and the name of the employee in charge of the task it depends on.

Which query would give the required result?

**PROJ\_TASK\_DETAILS**

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

- A. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p JOIN proj\_task\_details d  
ON (p.based\_on = d.task\_id);
- B. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p LEFT OUTER JOIN proj\_task\_details d  
ON (p.based\_on = d.task\_id);
- C. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p FULL OUTER JOIN proj\_task\_details d  
ON (p.based\_on = d.task\_id);
- D. SELECT p.task\_id, p.based\_on, d.task\_in\_charge  
FROM proj\_task\_details p JOIN proj\_task\_details d  
ON (p.task\_id = d.task\_id);

**Answer: B**

**Question: 100**

Which three statements/commands would cause a transaction to end? (Choose three.)

- A. COMMIT
- B. SELECT
- C. CREATE
- D. ROLLBACK
- E. SAVEPOINT

**Answer: A, C, D**

**Question: 101**

Which two statements are true regarding views? (Choose two.)

- A. A simple view in which column aliases have been used cannot be updated.
- B. Rows cannot be deleted through a view if the view definition contains the DISTINCT keyword.
- C. Rows added through a view are deleted from the table automatically when the view is dropped.

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- D. The OR REPLACE option is used to change the definition of an existing view without dropping and re-creating it.  
E. The WITH CHECK OPTION constraint can be used in a view definition to restrict the columns displayed through the view.

**Answer: B, D**

**Question: 102**

View the Exhibit and examine the structure of the PROMOTIONS table. Which two SQL statements would execute successfully? (Choose two.)

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. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE TO\_CHAR(promo\_end\_date, 'yyyy') > '2000';  
B. SELECT promo\_begin\_date  
FROM promotions  
WHERE TO\_CHAR(promo\_begin\_date,'mon dd yy')='JUL 01 98';  
C. UPDATE promotions  
SET promo\_cost = promo\_cost+ 100  
WHERE promo\_end\_date > TO\_DATE(SUBSTR('01-JAN-2000',8));  
D. SELECT TO\_CHAR(promo\_begin\_date,'dd/month')  
FROM promotions  
WHERE promo\_begin\_date IN (TO\_DATE('JUN 01 98'), TO\_DATE('JUL 01 98'));

**Answer: A, B**

**Question: 103**

Examine the structure of the MARKS table:

Name Null? Type

---

STUDENT\_ID NOT NULL VARCHAR2(4)  
STUDENT\_NAME VARCHAR2(25)  
SUBJECT1 NUMBER(3)  
SUBJECT2 NUMBER(3)  
SUBJECT3 NUMBER(3)

Which two statements would execute successfully? (Choose two.)

- A. SELECT student\_name,subject1  
FROM marks  
WHERE subject1 > AVG(subject1);  
B. SELECT student\_name,SUM(subject1)

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

FROM marks  
WHERE student\_name LIKE 'R%';  
C. SELECT SUM(subject1+subject2+subject3)  
FROM marks  
WHERE student\_name IS NULL;  
D. SELECT SUM(DISTINCT NVL(subject1,0)), MAX(subject1)  
FROM marks  
WHERE subject1 > subject2;

**Answer: C, D**

**Question: 104**

View the Exhibit and examine the structure of the CUSTOMERS table.

You have been asked to produce a report on the CUSTOMERS table showing the customers details sorted in descending order of the city and in the descending order of their income level in each city.

Which query would accomplish this task?

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 cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_city desc, cust\_income\_level DESC;  
B. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_income\_level desc, cust\_city DESC;  
C. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY (cust\_city, cust\_income\_level) DESC;  
D. SELECT cust\_city, cust\_income\_level, cust\_last\_name  
FROM customers  
ORDER BY cust\_city, cust\_income\_level DESC;

**Answer: A**

**Question: 105**

View the Exhibit and examine the structure of CUSTOMERS and GRADES tables.

You need to display names and grades of customers who have the highest credit limit.

Which two SQL statements would accomplish the task? (Choose two.)

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

#### CUSTOMERS

Name	Null?	Type
CUSTNO	NOT NULL	NUMBER (2)
CUSTNAME		VARCHAR2 (10)
CUSTADDRESS		VARCHAR2 (20)
CUST_CREDIT_LIMIT		NUMBER (5)

#### GRADES

Name	Null?	Type
GRADE	NOT NULL	VARCHAR2 (1)
STARTVAL		NUMBER (5)
ENDVAL		NUMBER (5)

- A. SELECT custname, grade  
FROM customers, grades  
WHERE (SELECT MAX(cust\_credit\_limit)  
FROM customers) BETWEEN startval and endval;
- B. SELECT custname, grade  
FROM customers, grades  
WHERE (SELECT MAX(cust\_credit\_limit)  
FROM customers) BETWEEN startval and endval  
AND cust\_credit\_limit BETWEEN startval AND endval;
- C. SELECT custname, grade  
FROM customers, grades  
WHERE cust\_credit\_limit = (SELECT MAX(cust\_credit\_limit)  
FROM customers)  
AND cust\_credit\_limit BETWEEN startval AND endval;
- D. SELECT custname, grade  
FROM customers, grades  
WHERE cust\_credit\_limit IN (SELECT MAX(cust\_credit\_limit)  
FROM customers)  
AND MAX(cust\_credit\_limit) BETWEEN startval AND endval;

**Answer: B, C**

#### Question: 106

View the Exhibit and examine the structure of the PROMOTIONS, SALES, and CUSTOMER tables.

You need to generate a report showing the promo name along with the customer name for all products that were sold during their promo campaign and before 30th October 2007.

You issue the following query:

```
SQL> SELECT promo_name,cust_name
  FROM promotions p JOIN sales s
  ON(time_id BETWEEN promo_begin_date AND promo_end_date)
  JOIN customer c
```

ON (s.cust\_id = c.cust\_id) AND time\_id < '30-oct-2007';

Which statement is true regarding the above query?

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

PROMOTIONS		
Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(2)
PROMO_NAME		VARCHAR2(10)
PROMO_CAT		VARCHAR2(10)
PROMO_COST		NUMBER(8,2)
PROMO_BEGIN_DATE		DATE
PROMO_END_DATE		DATE
SALES		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER(3)
PROMO_ID	NOT NULL	NUMBER(3)
TIME_ID		DATE
QTY_SOLD		NUMBER(6,2)
CUST_ID	NOT NULL	NUMBER(2)
CUSTOMER		
Name	Null?	Type
CUST_ID	NOT NULL	NUMBER(3)
CUST_NAME		VARCHAR2(20)
CUST_ADDRESS		VARCHAR2(30)

- A. It executes successfully and gives the required result.
- B. It executes successfully but does not give the required result.
- C. It produces an error because the join order of the tables is incorrect.
- D. It produces an error because equijoin and nonequijoin conditions cannot be used in the same SELECT statement.

**Answer: B**

**Question: 107**

Examine the structure of the INVOICE table.

Name Null? Type

```
-----
INV_NO NOT NULL NUMBER(3)
INV_DATE DATE
INV_AMT NUMBER(10,2)
```

Which two SQL statements would execute successfully? (Choose two.)

- A. `SELECT inv_no,NVL2(inv_date,'Pending','Incomplete')`  
FROM invoice;
- B. `SELECT inv_no,NVL2(inv_amt,inv_date,'Not Available')`  
FROM invoice;
- C. `SELECT inv_no,NVL2(inv_date,sysdate-inv_date,sysdate)`  
FROM invoice;
- D. `SELECT inv_no,NVL2(inv_amt,inv_amt*.25,'Not Available')`  
FROM invoice;

**Answer: A, C**

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>		
<b>Exam Type:</b>	<b>Oracle</b>		
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b>	<b>114</b>

**Question: 108**

Which statement is true regarding synonyms?

- A. Synonyms can be created only for a table.
- B. Synonyms are used to reference only those tables that are owned by another user.
- C. A public synonym and a private synonym can exist with the same name for the same table.
- D. The DROP SYNONYM statement removes the synonym, and the table on which the synonym has been created becomes invalid.

**Answer: C**

**Question: 109**

View the Exhibit and examine the structure of the CUSTOMERS table.

Using the CUSTOMERS table, you need to generate a report that shows the average credit limit for customers in WASHINGTON and NEW YORK.

Which SQL statement would produce the required result?

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 cust\_city, AVG(cust\_credit\_limit)  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK')  
GROUP BY cust\_credit\_limit, cust\_city;
- B. SELECT cust\_city, AVG(cust\_credit\_limit)  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK')  
GROUP BY cust\_city,cust\_credit\_limit;
- C. SELECT cust\_city, AVG(cust\_credit\_limit)  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK')  
GROUP BY cust\_city;
- D. SELECT cust\_city, AVG(NVL(cust\_credit\_limit,0))  
FROM customers  
WHERE cust\_city IN ('WASHINGTON','NEW YORK');

**Answer: C**

**Question: 110**

You want to create an ORD\_DETAIL table to store details for an order placed having the following business requirement:

- 1) The order ID will be unique and cannot have null values.

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

- 2) The order date cannot have null values and the default should be the current date.
- 3) The order amount should not be less than 50.
- 4) The order status will have values either shipped or not shipped.
- 5) The order payment mode should be cheque, credit card, or cash on delivery (COD).

Which is the valid DDL statement for creating the ORD\_DETAIL table?

A. CREATE TABLE ord\_details

```
(ord_id NUMBER(2) CONSTRAINT ord_id_nn NOT NULL,
ord_date DATE DEFAULT SYSDATE NOT NULL,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount > 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```

B. CREATE TABLE ord\_details

```
(ord_id NUMBER(2) CONSTRAINT ord_id_uk UNIQUE NOT NULL,
ord_date DATE DEFAULT SYSDATE NOT NULL,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount > 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```

C. CREATE TABLE ord\_details

```
(ord_id NUMBER(2) CONSTRAINT ord_id_pk PRIMARY KEY,
ord_date DATE DEFAULT SYSDATE NOT NULL,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount >= 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk ord_status VARCHAR2(15)
CONSTRAINT
ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk ord_pay_mode VARCHAR2(15)
CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```

D. CREATE TABLE ord\_details

```
(ord_id NUMBER(2),
ord_date DATE NOT NULL DEFAULT SYSDATE,
ord_amount NUMBER(5, 2) CONSTRAINT ord_amount_min
CHECK (ord_amount >= 50),
ord_status VARCHAR2(15) CONSTRAINT ord_status_chk
CHECK (ord_status IN ('Shipped', 'Not Shipped')),
ord_pay_mode VARCHAR2(15) CONSTRAINT ord_pay_chk
CHECK (ord_pay_mode IN ('Cheque', 'Credit Card',
'Cash On Delivery')));
```

**Answer: C**

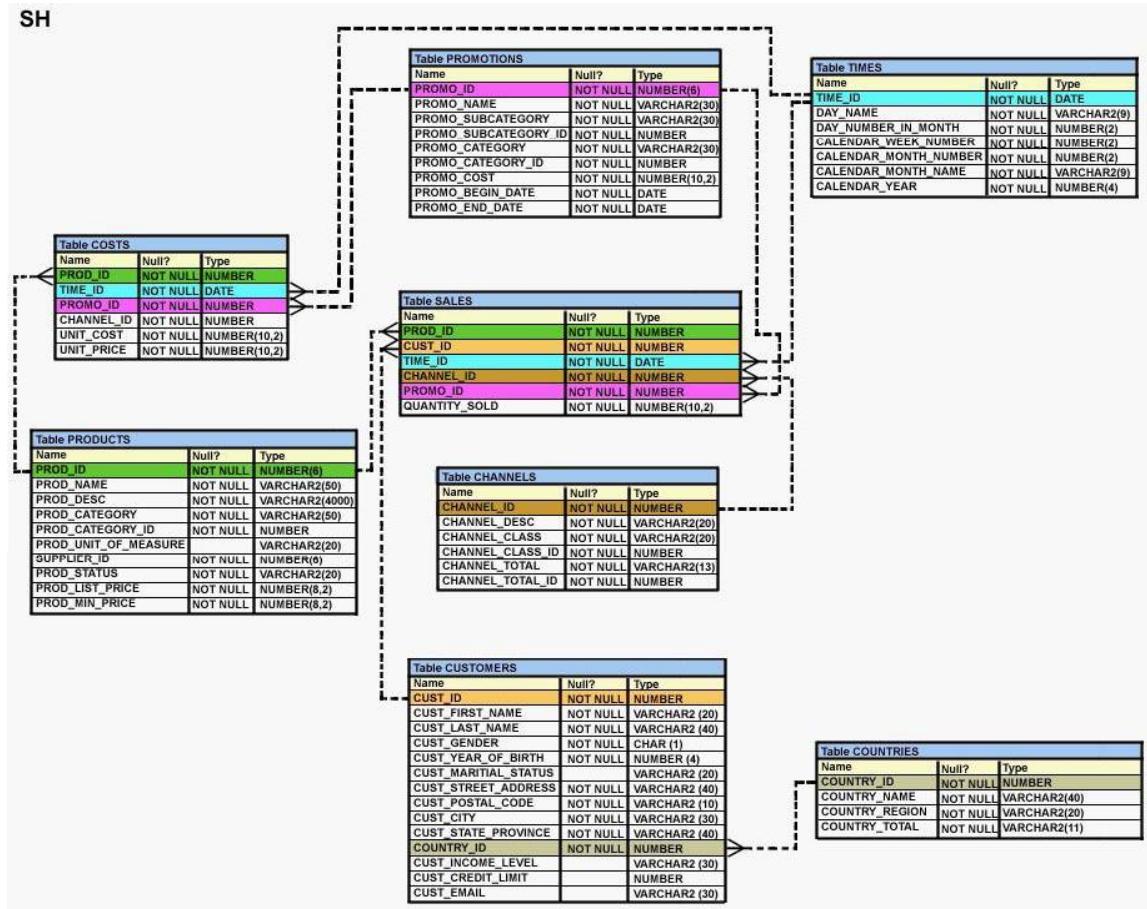
**Question: 111**

View the Exhibit and examine the description for the PRODUCTS and SALES table.

<b>Exam Name:</b>	<b>Oracle Database: SQL Fundamentals</b>	
<b>Exam Type:</b>	<b>Oracle</b>	
<b>Exam Code:</b>	<b>1Z0-051</b>	<b>Total Questions</b> <b>114</b>

PROD\_ID is a primary key in the PRODUCTS table and foreign key in the SALES table. You want to remove all the rows from the PRODUCTS table for which no sale was done for the last three years.

Which is the valid DELETE statement?



#### A. DELETE

```
FROM products
WHERE prod_id = (SELECT prod_id
FROM sales
WHERE time_id - 3*365 = SYSDATE );
```

#### B. DELETE

```
FROM products
WHERE prod_id = (SELECT prod_id
FROM sales
WHERE SYSDATE >= time_id - 3*365 );
```

#### C. DELETE

```
FROM products
WHERE prod_id IN (SELECT prod_id
FROM sales
WHERE SYSDATE - 3*365 >= time_id);
```

#### D. DELETE

```
FROM products
WHERE prod_id IN (SELECT prod_id
FROM sales
```

<b>Exam Name:</b>	<a href="#">Oracle Database: SQL Fundamentals</a>		
<b>Exam Type:</b>	<a href="#">Oracle</a>		
<b>Exam Code:</b>	<a href="#">1Z0-051</a>	<b>Total Questions</b>	<a href="#">114</a>

WHERE time\_id >= SYSDATE - 3\*365 );

**Answer: C**

**Question: 112**

Evaluate the following two queries:

```
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;
Which statement is true regarding the above two queries?
```

- A. Performance would improve in query 2.
- B. Performance would degrade in query 2.
- C. There would be no change in performance.
- D. Performance would improve in query 2 only if there are null values in the CUST\_CREDIT\_LIMIT column.

**Answer: C**

**Question: 113**

Which two statements are true regarding views? (Choose two.)

- A. A subquery that defines a view cannot include the GROUP BY clause.
- B. A view that is created with the subquery having the DISTINCT keyword can be updated.
- C. A view that is created with the subquery having the pseudo column ROWNUM keyword cannot be updated.
- D. A data manipulation language (DML) operation can be performed on a view that is created with the subquery having all the NOT NULL columns of a table.

**Answer: C, D**

**Question: 114**

Which three statements are true regarding subqueries? (Choose three.)

- A. Subqueries can contain GROUP BY and ORDER BY clauses.
- B. Main query and subquery can get data from different tables.
- C. Main query and subquery must get data from the same tables.
- D. Subqueries can contain ORDER BY but not the GROUP BY clause.
- E. Only one column or expression can be compared between the main query and subquery.
- F. Multiple columns or expressions can be compared between the main query and subquery.

**Answer: A, B, F**

**End of Document**