**Oracle\_CertifyMe\_1Z0-051\_v2011-05-18\_180q\_By-steeve**

Number: 1Z0-051

Passing Score: 600

Time Limit: 120 min

File Version: 2011-05-18



Oracle 1Z0-051

Questions: 180

Version: 2011-05-18

Good luck Friends!

By Steeve

Corrected by vlak

**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?

1. WHERE prod\_id LIKE '%\_D123%' ESCAPE '\_'
2. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\'
3. WHERE prod\_id LIKE '%\_D123%' ESCAPE '%\_'
4. WHERE prod\_id LIKE '%\\_D123%' ESCAPE '\\_'

**Correct Answer:** B

**Section: (none)**

**Explanation**

**Explanation/Reference:**

A naturally occurring underscore character may be escaped (or treated as a regular nonspecial symbol) using the ESCAPE identifier in conjunction with an ESCAPE character. The second example in Figure 3-12 shows the SQL statement that retrieves the JOBS table records with JOB\_ID values equal to SA\_MAN and SA\_REP

and which conforms to the original requirement:

select job\_id from jobs

where job\_id like 'SA\\_%' escape '\';

**QUESTION 4**

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

1. They accept only a single argument.
2. They can be nested only to two levels.
3. Arguments can only be column values or constants.
4. They always return a single result row for every row of a queried table.
5. They can return a data type value different from the one that is referenced.

**Correct Answer:** DE

**Section: (none)**

**Explanation**

**Explanation/Reference:**

A function is a program written to optionally accept input parameters, perform an operation, or return a single value. A function returns only one value per execution.

Three important components form the basis of defining a function. The first is the input parameter list. It specifies zero or more arguments that may be passed to a function as input for processing. These arguments or parameters may be of differing data types, and some are mandatory while others may be optional. The second component is the data type of its resultant value. Upon execution, only one value is returned by the function. The third encapsulates the details of the processing performed by the function and contains the program code that optionally manipulates the input parameters, performs calculations and operations, and generates a return value.

**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?

1. Both execute successfully and give correct results.
2. Only the first query executes successfully but gives a wrong result.
3. Only the first query executes successfully and gives the correct result.
4. Only the second query executes successfully but gives a wrong result.
5. Only the second query executes successfully and gives the correct result.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

The **MONTHS\_BETWEEN**(date 1, date 2) function returns the number of

months between two dates:

months\_between('01-FEB-2008','01-JAN-2008') = 1

**The DECODE Function**

Although its name sounds mysterious, this function is straightforward. The DECODE function implements if-then-else conditional logic by testing its first two terms for equality and returns the third if they are equal and optionally returns another term if they are not.

**DECODE Function**

Facilitates conditional inquiries by doing the work of a CASE expression or an IF-THEN-

ELSE statement:

**DECODE(col|expression, search1, result1**

**[, search2, result2,...,]**

**[, default])**

**DECODE Function**

The DECODE function decodes an expression in a way similar to the IF-THEN-ELSE logic that is used in various languages. The DECODE function decodes expression after comparing it to each search value. If the expression is the same as search, result is returned.

If the default value is omitted, a null value is returned where a search value does not match anyof the result values

**QUESTION 7**

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

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

**Correct Answer:** CD

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Creating Joins with the USING Clause**

If several columns have the same names but the data types do not match, use the USING clause to specify the columns for the equijoin.

Use the USING clause to match only one column when more than one column matches. The NATURAL JOIN and USING clauses are mutually exclusive

**Using Table Aliases with the USING clause**

When joining with the USING clause, you cannot qualify a column that is used in the USING clause itself. Furthermore, if that column is used anywhere in the SQL statement, you cannot alias it. For example, in the query mentioned in the slide, you should not alias the location\_id column in the WHERE clause because the column is used in the USING clause. The columns that are referenced in the USING clause should not have a qualifier (table name oralias) anywhere in the SQL statement.

**Creating Joins with the ON Clause**

The join condition for the natural join is basically an equijoin of all columns with the same name.

Use the ON clause to specify arbitrary conditions or specify columns to join. – ANSWER C The join condition is separated from other search conditions. ANSWER D The ON clause makes code easy to understand.

**QUESTION 9**

Which statement is true regarding the INTERSECT operator?

1. It ignores NULL values.
2. Reversing the order of the intersected tables alters the result.
3. The names of columns in all SELECT statements must be identical.
4. The number of columns and data types must be identical for all SELECT statements in the query.

**Correct Answer:** D

**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?

1. SELECT cust\_last\_name Name, cust\_credit\_limit + 1000 "New Credit Limit"

FROM customers;

1. SELECT cust\_last\_name AS Name, cust\_credit\_limit + 1000 AS New Credit Limit

FROM customers;

1. SELECT cust\_last\_name AS "Name", cust\_credit\_limit + 1000 AS "New Credit Limit"

FROM customers;

1. SELECT INITCAP(cust\_last\_name) "Name", cust\_credit\_limit + 1000 INITCAP("NEW CREDIT LIMIT") FROM customers;

**Correct Answer:** C

**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?

1. 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

1. SELECT DISTINCT cust\_income\_level ' ' cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;
2. SELECT cust\_income\_level ' ' cust\_credit\_limit \* 0.50 AS "50% Credit Limit" FROM customers;

**Correct Answer:** C

**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?

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

**Correct Answer:** A

**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?

1. It produces an error because flower braces have been used.
2. 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.

**Correct Answer:** C

**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?

1. SELECT DISTINCT promo\_cost, promo\_category FROM promotions;
2. SELECT promo\_category, DISTINCT promo\_cost FROM promotions;
3. SELECT DISTINCT promo\_cost, DISTINCT promo\_category FROM promotions;
4. SELECT DISTINCT promo\_category, promo\_cost FROM promotions ORDER BY 1;

**Correct Answer:** D

**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?

1. +25-00 , +54-02, +00 11:12:10.123457
2. +00-300, +54-02, +00 11:12:10.123457
3. +25-00 , +00-650, +00 11:12:10.123457
4. +00-300 , +00-650, +00 11:12:10.123457

**Correct Answer:** A

**QUESTION 20**

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

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

**Correct Answer:** ADE

**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.)

1. An EMP\_IMAGE column can be included in the GROUP BY clause.
2. An EMP\_IMAGE column cannot be included in the ORDER BY clause.
3. You cannot add a new column to the table with LONG as the data type.
4. You can alter the table to include the NOT NULL constraint on the EMP\_IMAGE column.

**Correct Answer:** BC

**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?

1. DATE
2. NUMBER
3. TIMESTAMP
4. INTERVAL DAY TO SECOND
5. INTERVAL YEAR TO MONTH

**Correct Answer:** D

**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.)

1. The CUST\_STATUS column would give an error.
2. The TRANS\_VALIDITY column would give an error.
3. The CUST\_STATUS column would store exactly one character.
4. The CUST\_CREDIT\_LIMIT column would not be able to store decimal values.
5. The TRANS\_VALIDITY column would have a maximum size of one character.
6. The TRANS\_DATE column would be able to store day, month, century, year, hour, minutes, seconds, and fractions of seconds.

**Correct Answer:** BC

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.)

1. The TRANS\_DATE column would allow storage of dates only in the dd-mon-yyyy format
2. The CUST\_CREDIT\_VALUE column would allow storage of positive and negative integers.
3. The TRANS\_VALIDITY column would allow storage of a time interval in days, hours, minutes, and seconds.
4. The CUST\_STATUS column would allow storage of data up to the maximum VARCHAR2 size of 4,000 characters.
5. **Correct Answer:** BC

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.

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.

Correct Answer: BC

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

Correct Answer: D

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.

Correct Answer: BC

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.

Correct Answer: BC

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?

1. CREATE TABLE EMP\_1

(emp\_id NUMBER(4),

emp\_name VARCHAR2(25),

start\_date DATE,

e\_status VARCHAR2(10)

DEFAULT 'ACTIVE',

resume CLOB(200));

1. CREATE TABLE 1\_EMP

(emp\_id NUMBER(4),

emp\_name VARCHAR2(25),

start\_date DATE,

emp\_status VARCHAR2(10)

DEFAULT 'ACTIVE', resume CLOB);

1. CREATE TABLE EMP\_1

(emp\_id NUMBER(4),

emp\_name VARCHAR2(25),

start\_date DATE,

emp\_status VARCHAR2(10)

DEFAULT "ACTIVE",

resume CLOB);

1. CREATE TABLE EMP\_1

(emp\_id NUMBER,

emp\_name VARCHAR2(25),

start\_date DATE,

emp\_status VARCHAR2(10)

DEFAULT 'ACTIVE',

resume CLOB);

Correct Answer: D

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);

Correct Answer: A

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.

Correct Answer: EF

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.

Correct Answer: BD

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.

Correct Answer: BD

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.

Correct Answer: A

QUESTION 31 Evaluate the following SQL commands:

**SQL>CREATE SEQUENCE ord\_seq**

**INCREMENT BY10**

**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.

Correct Answer: AC

QUESTION 32 Which CREATE TABLE statement is valid?

1. 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));

Correct Answer: D

**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.

Correct Answer: D

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.

Correct Answer: BE

**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.

**Correct Answer:** BE

**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.

**Correct Answer:** ABD

**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 recreating

it.

E. The WITH CHECK OPTION constraint can be used in a view definition to restrict the columns displayed

Through the view.

**Correct Answer:** BD

**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.

**Correct Answer:** CD

**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.

**Correct Answer:** BDF

**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 is restarted.

**Correct Answer:** CD

**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.

**Correct Answer:** ACD

**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.

**Correct Answer:** AD

**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;**

1. CREATE SYNONYM ord FOR orders; This command is issued by OE.
2. CREATE PUBLIC SYNONYM ord FOR orders; This command is issued by OE.
3. CREATE SYNONYM ord FOR oe.orders; This command is issued by the database administrator.
4. CREATE PUBLIC SYNONYM ord FOR oe.orders; This command is issued by the database administrator.

**Correct Answer:** D

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Creating a Synonym for an Object**

To refer to a table that is owned by another user, you need to prefix the table name with the name of the user who created it, followed by a period. Creating a synonym eliminates the need to qualify the object name with the schema and provides you with an alternative name for a table, view, sequence, procedure, or other objects. This method can be especially useful with lengthy object names, such as views. In the syntax:

PUBLIC Creates a synonym that is accessible to all users **synonym** Is the name of the synonym to be created **object** Identifies the object for which the synonym is created

**Guidelines**

The object cannot be contained in a package.

A private synonym name must be distinct from all other objects that are owned by the same user.

If you try to execute the following command (alternative B, issued by OE):

**CREATE PUBLIC SYNONYM ord FOR orders;**

You will get an error.

Error que empieza en la línea 693 del comando:

create public synonym nuly for prueba\_null

Error en la línea de comandos:693 Columna:0

Informe de error:

Error SQL: ORA-01031: privilegios insuficientes

01031. 00000 - "insufficient privileges"

The message gives you the answer: OE doesn't have enough privileges. However, if you give the necessary privileges (issued by DBA):

GRANT **CREATE PUBLIC SYNONYM** TO **OE**;

You won't have problems executing the command in the alternative B (issued by OE): CREATE PUBLIC SYNONYM ord FOR orders;

Finally, if you need to be sure what system privileges you have in your active session, you can execute the following command (issued by OE):

SELECT \* FROM USER\_PRIVS;

(One of the rows must be: **CREATE PUBLIC SYNONYM** ).

**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?

1. Only the synonym would be dropped.
2. The synonym would be dropped and the corresponding table would become invalid.
3. The synonym would be dropped and the packages referring to the synonym would be dropped.
4. The synonym would be dropped and any PUBLIC synonym with the same name becomes invalid.

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

A synonym is an alias for

a table (or a view). Users can execute SQL statements against the synonym, and the database will map them into statements against the object to which the synonym points.

Private synonyms are schema objects. Either they must be in your own schema, or they must be qualified with the schema name. Public synonyms exist independently of a schema. A public synonym can be referred to by any user to whom permission has been granted to see it without the need to qualify it with a schema name.

Private synonyms must be a unique name within their schema**. Public synonyms can have the same name** **as schema objects**. When executing statements that address objects without a schema qualifier, Oracle willfirst look for the object in the local schema, and only if it cannot be found will it look for a public synonym.

**QUESTION 51**

Which statement is true regarding synonyms?

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

**Correct Answer:** C

**Section: (none)**

**Explanation**

**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.

**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?

1. Performance would improve in query 2.
2. Performance would degrade in query 2.
3. There would be no change in performance.
4. Performance would improve in query 2 only if there are null values in the CUST\_CREDIT\_LIMIT column.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Note:** The IN operator is internally evaluated by the Oracle server as a set of OR conditions, such asa=value1 or a=value2 or a=value3. Therefore, using the IN operator has no performance benefits and is used only for logical simplicity.

**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?

1. It produces an error.
2. It displays all values.
3. It displays only the values A%\_WQ123 and AB\_WQ123 .
4. It displays only the values A%\_WQ123 and A%BWQ123 .
5. It displays only the values A%BWQ123 and AB\_WQ123.

**Correct Answer:** D

**QUESTION 67**

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

1. It is executed first in the query execution.
2. It must be the last clause in the SELECT statement.
3. It cannot be used in a SELECT statement containin g a HAVING clause.
4. You cannot specify a column name followed by an expression in this clause.
5. You can specify a combination of numeric positions and column names in this clause.

**Correct Answer:** BE

**Section: (none)**

**Explanation**

**QUESTION 68**

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

1. In a character sort, the values are case- sensitive.
2. NULL values are not considered at all by the sort operation.
3. Only those columns that are specified in the SELECT list can be used in the ORDER BY clause.
4. Numeric values are displayed from the maximum to the minimum value if they have decimal positions.

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Character Strings and Dates

Character strings and date values are enclosed with single quotation marks.

Character values **are case-sensitive** and date values are format-sensitive.

The default date display format is DD-MON-RR.

**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.)

1. SELECT cust\_last\_name, cust\_credit\_limit FROM customers

ORDER BY cust\_credit\_limit DESC ;

1. SELECT cust\_last\_name, cust\_credit\_limit FROM customers

ORDER BY cust\_credit\_limit;

1. SELECT cust\_last\_name, cust\_credit\_limit FROM customers

ORDER BY cust\_credit\_limit NULLS LAST;

1. SELECT cust\_last\_name, cust\_credit\_limit FROM customers

ORDER BY cust\_last\_name, cust\_credit\_limit NULLS LAST;

**Correct Answer:** BC

**Section: (none)**

**Explanation**

**Explanation/Reference:**

If the ORDER BY clause is not used, the sort order is undefined, and the Oracle server may not fetch rows in the same order for the same query twice. Use the ORDER BY clause to display the rows in a specific order.

**Note:** Use the keywords NULLS FIRST or NULLS LAST to specify whether returned rowscontaining null values should appear first or last in the ordering sequence. ANSWER C

**Sorting**

The default sort order is ascending:

* Numeric values are displayed with the lowest values first (for example, 1 to 999).
* Date values are displayed with the earliest value first (for example, 01-JAN-92 before 01-JAN-95).
* Character values are displayed in the alphabetical order (for example, “A” first and “Z” last).
* Null values are displayed last for ascending sequences and first for descending

sequences. - ANSWER B

• You can also sort by a column that is not in the SELECT list.

**QUESTION 71**

Which arithmetic operations can be performed on a column by using a SQL function that is built into Oracle database ? (Choose three .)

1. addition
2. subtraction
3. raising to a power
4. finding the quotient
5. finding the lowest value

**Correct Answer:** ACE

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 72**

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

1. displaying a date in a nondefault format
2. finding the number of characters in an expression
3. substituting a character string in a text expression with a specified string
4. combining more than two columns or expressions into a single column in the output

**Correct Answer:** ABC

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 73**

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

1. finding the remainder of a division
2. adding a number to a date for a resultant date value
3. comparing two expressions to check whether they are equal
4. checking whether a specified character exists in a given string
5. removing trailing, leading, and embedded characters from a character string

**Correct Answer:** ACD

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 74**

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

1. MOD : returns the quotient of a division
2. TRUNC : can be used with NUMBER and DATE values
3. CONCAT : can be used to combine any number of values
4. SYSDATE : returns the database server current date and time
5. INSTR : can be used to find only the first occurrence of a character in a string
6. TRIM : can be used to remove all the occurrences of a character from a string

**Correct Answer:** BD

**Section: (none)**

**Explanation**

**Explanation/Reference:**

ROUND: Rounds value to a specified decimal

TRUNC: Truncates value to a specified decimal

MOD: Returns remainder of division

**SYSDATE i**s a date function that returns the current database server date and time.

**Date-Manipulation Functions**

Date functions operate on Oracle dates. All date functions return a value of the DATE data type except MONTHS\_BETWEEN, which returns a numeric value.

MONTHS\_BETWEEN(**date1, date2**): Finds the number of months between **date1** and **date2**. The result can be positive or negative. If **date1** is later than **date2**, the result is positive; if **date1** is earlier than **date2**, the result is negative. The noninteger part of the result represents a portion of the month.

ADD\_MONTHS(**date, n**): Adds **n** number of calendar months to **date**. The value of **n** must be an integer and can be negative.

NEXT\_DAY(**date,** '**char**'): Finds the date of the next specified day of the week ('**char**') following **date**. The

value of **char** may be a number representing a day or a character string.

LAST\_DAY(**date**): Finds the date of the last day of the month that contains **date**

The above list is a subset of the available date functions. ROUND and TRUNC number functions can also be

used to manipulate the date values as shown below:

ROUND(**date**[,'**fmt**']): Returns **date** rounded to the unit that is specified by the format model **fmt.** If the format model **fmt** is omitted, **date** is rounded to the nearest day.

**TRUNC(date[, 'fmt'])**: Returns **date** with the time portion of the day truncated to the unit that is specifiedby the format model **fmt**. If the format model **fmt** is omitted, **date** is truncated to the nearest day.

**The CONCAT Function**

The CONCAT function joins **two** character literals, columns, or expressions to yield one larger character expression. Numeric and date literals are implicitly cast as characters when they occur as parameters to the CONCAT function. Numeric or date expressions are evaluated before being converted to strings ready to be concatenated. The CONCAT function takes two parameters. Its syntax is CONCAT(s1, s2), where s1 and s2 represent string literals, character column values, or expressions resulting in character values.

The **INSTR**(source string, search item, [start position],[**nth** occurrence of search item]) function returns a

number that represents the position in the source string, beginning from the given start position, where the **nth** occurrence of the search item begins:

instr('http://www.domain.com','.',1,2) = 18

The **TRIM** function literally trims off leading or trailing (or both) character strings from a given source string:

**QUESTION 75**

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?

1. 152526 \*\*\*\*
2. \*\*152525.99
3. 152525\*\* \*\*
4. an error message

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

The **LPAD**(string, length after padding, padding string) and **RPAD** (string, length after padding, padding string) functions add a padding string of characters to the left or right of a string until it reaches the specified length after padding.

**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?

1. SELECT cust\_first\_name
2. FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>0 AND

SUBSTR(cust\_first\_name, -2, 1)='a';

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>'' AND SUBSTR(cust\_first\_name, -2, 1)='a';

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')IS NOT NULL AND SUBSTR(cust\_first\_name, 1,-2)='a';

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>0 AND

SUBSTR(cust\_first\_name, LENGTH(cust\_first\_name),-2)='a';

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

The **SUBSTR**(string, start position, number of characters) function accepts three parameters and returns a string consisting of the number of characters extracted from the source string, beginning at the specified start position:

substr('http://www.domain.com',12,6) = domain

The position at which the first character of the returned string begins.

When position is 0 (zero), then it is treated as 1.

When position is positive, then the function counts from the beginning of string to find the first character. When position is negative, then the function counts backward from the end of string. substring\_length

The length of the returned string. SUBSTR calculates lengths using characters as defined by the input character set. SUBSTRB uses bytes instead of characters. SUBSTRC uses Unicode complete characters. SUBSTR2 uses UCS2 code points. SUBSTR4 uses UCS4 code points. When you do not specify a value for this argument, then the function

The **INSTR**(source string, search item, [start position],[nth occurrence of search item]) function returns a

number that represents the position in the source string, beginning from the given start position, where the nth

occurrence of the search item begins:

instr('http://www.domain.com','.',1,2) = 18

**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?

1. Abigail PA
2. Abigail Pa
3. Abigail IS
4. an error message

**Correct Answer:** B

**Section: (none)**

**Explanation**

**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?

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>0 AND

SUBSTR(cust\_first\_name, -2, 1)='a';

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>'' AND SUBSTR(cust\_first\_name, -2, 1)='a';

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')IS NOT NULL AND SUBSTR(cust\_first\_name, 1,-2)='a';

1. SELECT cust\_first\_name FROM customers

WHERE INSTR(cust\_first\_name, 'e')<>0 AND

SUBSTR(cust\_first\_name, LENGTH(cust\_first\_name),-2)='a';

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

The **SUBSTR**(string, start position, number of characters) function accepts three parameters and returns a string consisting of the number of characters extracted from the source string, beginning at the specified start position:

substr('http://www.domain.com',12,6) = domain

The position at which the first character of the returned string begins.

When position is 0 (zero), then it is treated as 1.

When position is positive, then the function counts from the beginning of string to find the first character. When position is negative, then the function counts backward from the end of string. substring\_length

The length of the returned string. SUBSTR calculates lengths using characters as defined by the input character set. SUBSTRB uses bytes instead of characters. SUBSTRC uses Unicode complete characters. SUBSTR2 uses UCS2 code points. SUBSTR4 uses UCS4 code points. When you do not specify a value for this argument, then the function

The **INSTR**(source string, search item, [start position],[nth occurrence of search item]) function returns a

number that represents the position in the source string, beginning from the given start position, where the nth

occurrence of the search item begins:

instr('http://www.domain.com','.',1,2) = 18

**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?

1. Abigail PA
2. Abigail Pa
3. Abigail IS
4. an error message

**Correct Answer:** B

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 78**

Evaluate the following query:

**SQL> SELECT TRUNC(ROUND(156.00,-1),-1) FROM DUAL;** What would be the outcome?

1. 16
2. 100
3. 160
4. 200
5. 150

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Function Purpose**

ROUND(**column**|**expression**, **n**) Rounds the column, expression, or value to **n** decimal places or, if **n** is omitted, no decimal places (If **n** is negative, numbers to the left of decimal point are rounded.)

TRUNC(**column**|**expression**, **n**) Truncates the column, expression, or value to **n** decimal places or, if **n** is omitted, **n** defaults to zero.

**QUESTION 80**

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

1. The default internal storage of dates is in the numeric format.
2. The default internal storage of dates is in the character format.
3. The RR date format automatically calculates the century from the SYSDATE function and does not allow the user to enter the century.
4. The RR date format automatically calculates the century from the SYSDATE function but allows the user to enter the century if required.

**Correct Answer:** A, D

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Working with Dates**

The Oracle Database stores dates in an internal numeric format, representing the century, year, month, day, hours, minutes, and seconds.

The default display and input format for any date is DD-MON-RR.

**RR Date Format**

The RR date format is similar to the YY element, but you can use it to specify different centuries. Use the RR date format element instead of YY so that the century of the return value varies according to the specified two-digit year and the last two digits of the current year. The table in the slide summarizes the behavior of the RR element.

|  |  |  |  |
| --- | --- | --- | --- |
| **Current Year** | **Given Date** | **Interpreted (RR)** | **Interpreted (YY)** |
| 1994 | 27-OCT-95 | 1995 | 1995 |
|  |  |  |  |
| 1994 | 27-OCT-17 | 2017 | 1917 |
|  |  |  |  |
| 2001 | 27-OCT-17 | 2017 | 2017 |
|  |  |  |  |
| 2048 | 27-OCT-52 | 1952 | 2052 |
|  |  |  |  |
| 2051 | 27-OCT-47 | 2147 | 2047 |

Note the values shown in the last two rows of the above table. As we approach the middle of the century, then

the RR behavior is probably not what you want.

This data is stored internally as follows:

CENTURY YEAR MONTH DAY HOUR MINUTE SECOND 19 87 06 17 17 10 43

**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?

1. an error because the ROUND function specified is invalid
2. an error because the WHERE condition specified is invalid
3. number of days since the promo started based on the current Chicago date and time
4. number of days since the promo started based on the current Singapore date and time

**Correct Answer:** C

**Section: (none)**

**Explanation**

**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?

1. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1) FROM customers

WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name,' ')+1))='Mc';

1. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1) FROM customers

WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)) LIKE 'Mc%';

1. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1) FROM customers

WHERE SUBSTR(cust\_name, INSTR(cust\_name,' ')+1) LIKE INITCAP('MC%');

1. SELECT SUBSTR(cust\_name, INSTR(cust\_name,' ')+1) FROM customers

WHERE INITCAP(SUBSTR(cust\_name, INSTR(cust\_name,' ')+1)) = INITCAP('MC%');

**Correct Answer:** B

**Section: (none)**

**Explanation**

**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.)

1. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*') "CUST NAME" FROM customers

WHERE INSTR(cust\_name, ' ',1,2)<>0;

1. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name),'\*') "CUST NAME" FROM customers

WHERE INSTR(cust\_name, ' ',-1,2)<>0;

1. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name)- INSTR (cust\_name,''),'\*') "CUST NAME"

FROM customers

WHERE INSTR(cust\_name, ' ',-1,-2)<>0;

1. SELECT LPAD(SUBSTR(cust\_name,INSTR(cust\_name,' ')),LENGTH(cust\_name)- INSTR(cust\_name,' '),'\*') "CUST NAME"

FROM customers

WHERE INSTR(cust\_name, ' ',1,2)<>0 ;

**Correct Answer:** A,B

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 85**

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

**name Null Type**

**---------------------------**

**INV\_NO NOT NULL NUMBER(3)**

**INV\_DATE DATE**

**INV\_AMT NUMBER(10,2)**

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

1. inv\_amt ='0255982' : requires explicit conversion
2. inv\_date > '01-02-2008' : uses implicit conversion
3. CONCAT(inv\_amt,inv\_date) : requires explicit conversion
4. inv\_date = '15-february-2008' : uses implicit conversion
5. inv\_no BETWEEN '101' AND '110' : uses implicit conversion

**Correct Answer:** D,E

**Section: (none)**

**Explanation**

**Explanation/Reference:**

In some cases, the Oracle server receives data of one data type where it expects data of a different data type. When this happens, the Oracle server can automatically convert the data to the expected data type. This data type conversion can be done **implicitly** by the Oracle server or **explicitly** by the user.

**Explicit** data type conversions are performed by using theconversion functions. Conversion functions convert

a value from one data type to another. Generally, the form of the function names follows the convention **data** **type** TO **data type**. The first data type is the input data type and the second data type is the output. **Note:** Although implicit data type conversion is available, it is recommended that you do the explicit data typeconversion to ensure the reliability of your SQL statements.

**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 .)

1. SELECT transdate + '10' FROM cust\_trans;
2. SELECT \* FROM cust\_trans WHERE transdate = '01-01-07';
3. SELECT transamt FROM cust\_trans WHERE custno > '11';
4. SELECT \* FROM cust\_trans WHERE transdate='01-JANUARY-07';
5. SELECT custno + 'A' FROM cust\_trans WHERE transamt > 2000;

**Correct Answer:** A,C,D

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 87**

You want to display the date for the first Monday 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?

1. It executes successfully and returns the correct result.
2. It executes successfully but does not return the correct result.
3. It generates an error because TO\_CHAR should be replaced with TO\_DATE.
4. It generates an error because rrrr should be replaced by rr in the format string.
5. It generates an error because fm and double quotation marks should not be used in the format string.

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

* NEXT\_DAY(date, 'char'): Finds the date of the next specified day of the week ('char') following date. The value of char may be a number representing a day or a character string.
* LAST\_DAY(date): Finds the date of the last day of the month that contains date

The second innermost function is evaluated next. TO\_CHAR('28-OCT-2009', 'fmMonth') converts the given date based on the Month format mask and returns the character string October. The fm modifier trims trailing blank spaces from the name of the month.

**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 .)

1. SELECT SYSDATE - '01-JAN-2007' FROM DUAL;
2. SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;
3. SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL;
4. SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;
5. SELECT TO\_DATE(SYSDATE, 'DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL;

**Correct Answer:** B,C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**What is the outcome?**

1. It executes successfully and returns the correct result.
2. It executes successfully but does not return the correct result.
3. It generates an error because TO\_CHAR should be replaced with TO\_DATE.
4. It generates an error because rrrr should be replaced by rr in the format string.
5. It generates an error because fm and double quotation marks should not be used in the format string.

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

* NEXT\_DAY(date, 'char'): Finds the date of the next specified day of the week ('char') following date. The value of char may be a number representing a day or a character string.
* LAST\_DAY(date): Finds the date of the last day of the month that contains date

The second innermost function is evaluated next. TO\_CHAR('28-OCT-2009', 'fmMonth') converts the given date based on the Month format mask and returns the character string October. The fm modifier trims trailing blank spaces from the name of the month.

**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 .)

1. SELECT SYSDATE - '01-JAN-2007' FROM DUAL;
2. SELECT SYSDATE - TO\_DATE('01/JANUARY/2007') FROM DUAL;
3. SELECT SYSDATE - TO\_DATE('01-JANUARY-2007') FROM DUAL;
4. SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY') - '01-JAN-2007' FROM DUAL;
5. SELECT TO\_DATE(SYSDATE, 'DD/MONTH/YYYY') - '01/JANUARY/2007' FROM DUAL;

**Correct Answer:** B,C

**Section: (none)**

**Explanation**

**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?

1. SELECT TO\_CHAR('11-oct-2007', 'fmDdspth "of" Month, Year') FROM DUAL;
2. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdspth of month, year') FROM DUAL;
3. SELECT TO\_CHAR(TO\_DATE('11-oct-2007'), 'fmDdthsp "of" Month, Year') FROM DUAL;
4. SELECT TO\_DATE(TO\_CHAR('11-oct-2007','fmDdspth ''of'' Month, Year')) FROM DUAL;

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Using the **TO\_CHAR** Function with Dates

TO\_CHAR converts a datetime data type to a value of VARCHAR2 data type in the format specified by the format\_model. A format model is a character literal that describes the format of datetime stored in a character string. For example, the datetime format model for the string '11-Nov-1999' is 'DD-Mon-YYYY'. You can use the TO\_CHAR function to convert a date from its default format to the one that you specify. Guidelines

* The format model must be enclosed with single quotation marks and is case-sensitive.
* The format model can include any valid date format element. But be sure to separate the date value from the format model with a comma.
* The names of days and months in the output are automatically padded with blanks.
* To remove padded blanks or to suppress leading zeros, use the fill mode **fm** element.

**Elements of the Date Format Model**

**---------------------------------------------------------------------**

**DY** Three-letter abbreviation of the day of the week

**DAY** Full name of the day of the week

1. Numeric day of the month
2. Two-digit value for the month

**MON** Three-letter abbreviation of the month

**MONTH** Full name of the month

1. Full year in numbers

**YEAR** Year spelled out (in English)

**QUESTION 91**

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

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

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'));

**Correct Answer:** AB

**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. SELECTpromo\_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;

**Correct Answer:** CD

**QUESTION 93**

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

Using the CUSTOMERS table, y ou 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?

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;

**Correct Answer:** D

**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;

**Correct Answer:** AD

**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.

**Correct Answer:** A

**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;

**Correct Answer:** AC

**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.

**Correct Answer:** C

**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.

**Correct Answer:** C

**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?

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.

**Correct Answer:** A

**QUESTION 100**

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

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

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;

**Correct Answer:** AB

**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;

**Correct Answer:** A

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

Correct Answer: ABC

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

Correct Answer: ACD

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

Correct Answer: BD

QUESTION 75

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

Correct Answer: A

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';

Correct Answer: A

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

Correct Answer: B

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

Correct Answer: C

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?

A. 'Oder' and 'Aus'

B. an error because the TRIM function specified is not valid

C. an error because the LOWER function specified is not valid

D. an error because the REPLACE function specified is not valid

Correct Answer: B

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.

Correct Answer: AD

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. 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

Correct Answer: C

QUESTION 82

xamine 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%');

Correct Answer: B

**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?

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(\*).

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Restricting Group Results with the HAVING Clause**

You use the HAVING clause to specify the groups that are to be displayed, thus further restricting the groups on the basis of aggregate information.

In the syntax, **group\_condition** restricts the groups of rows returned to those groups for which the specified condition is true.

The Oracle server performs the following steps when you use the HAVING clause:

1.  Rows are grouped.

2.  The group function is applied to the group.

3.  The groups that match the criteria in the HAVING clause are displayed.

The HAVING clause can precede the GROUP BY clause, but it is recommended that you place the GROUP BY clause first because it is more logical. Groups are formed and group functions are calculated before the HAVING clause is applied to the groups in the SELECT list.

**Note:** The WHERE clause restricts rows, whereas the HAVING clause restricts groups.

**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?

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.

**Correct Answer:** A

**Section: (none)**

**Explanation**

**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.

**Correct Answer:** B

**Section: (none)**

**Explanation**

**Explanation/Reference:**

The general form of the SELECT statement is further enhanced by the addition of the HAVING clause and becomes:

SELECT column|expression|group\_function(column|expression [alias]),…}

FROM table

[WHERE condition(s)]

[GROUP BY {col(s)|expr}]

[HAVING group\_condition(s)]

[ORDER BY {col(s)|expr|numeric\_pos} [ASC|DESC] [NULLS FIRST|LAST]];

An important difference between the HAVING clause and the other SELECT statement clauses is that it may only be specified if a GROUP BY clause is present. This dependency is sensible since group-level rows must exist before they can be restricted. The HAVING clause can occur before the GROUP BY clause in the SELECT statement. However, it is more common to place the HAVING clause after the GROUP BY clause. All grouping is performed and group functions are executed prior to evaluating the HAVING clause.

**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.

**Correct Answer:** AC

**Section: (none)**

**Explanation**

**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?

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

**Correct Answer:** D

**Section: (none)**

**Explanation**

**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 in the table.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**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.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Creating Joins with the USING Clause**

Natural joins use all columns with matching names and data types to join the tables. The USING clause can be used to specify only those columns that should be used for an equijoin.

**The Natural JOIN USING Clause**

The format of the syntax for the natural JOIN USING clause is as follows:

SELECT table1.column, table2.column

FROM table1

JOIN table2 USING (join\_column1, join\_column2…);

While the pure natural join contains the NATURAL keyword in its syntax, the JOIN…USING syntax does not. An error is raised if the keywords NATURAL and USING occur in the same join clause. The JOIN…USING clause allows one or more equijoin columns to be explicitly specified in brackets after the USING keyword. This avoids the shortcomings associated with the pure natural join. Many situations demand that tables be joined only on certain columns, and this format caters to this requirement.

**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.

**Correct Answer:** BD

**Section: (none)**

**Explanation**

**Explanation/Reference:**

NATURAL JOIN operation

A NATURAL JOIN is a JOIN operation that creates an implicit join clause for you based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.

If the SELECT statement in which the NATURAL JOIN operation appears has an asterisk (\*) in the select list, the asterisk will be expanded to the following list of columns (in this order):

All the common columns

Every column in the first (left) table that is not a common column Every column in the second (right) table that is not a common column

An asterisk qualified by a table name (for example, COUNTRIES.\*) will be expanded to every column of that table that is not a common column.

If a common column is referenced without being qualified by a table name, the column reference points to the column in the first (left) table if the join is an INNER JOIN or a LEFT OUTER JOIN. If it is a RIGHT OUTER JOIN, unqualified references to a common column point to the column in the second (right) table.

Syntax

TableExpression NATURAL [ { LEFT | RIGHT } [ OUTER ] | INNER ] JOIN { TableViewOrFunctionExpression |

( TableExpression ) }

Examples

If the tables COUNTRIES and CITIES have two common columns named COUNTRY and

COUNTRY\_ISO\_CODE, the following two SELECT statements are equivalent:

SELECT \* FROM COUNTRIES NATURAL JOIN CITIES

SELECT \* FROM COUNTRIES JOIN CITIES

**USING** (COUNTRY, COUNTRY\_ISO\_CODE)

**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 s tatement 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.

**Correct Answer:** D

**Section: (none)**

**Explanation**

**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';

**Correct Answer:** AC

**Section: (none)**

**Explanation**

**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?

**Exhibit:**

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.

**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

**Correct Answer:** AB

**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

**Correct Answer:** AE

**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);

**Correct Answer:** C

**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.

**Correct Answer:** AD

**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

**Correct Answer:** ABCF

**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.

**Correct Answer:** ABF

**QUESTION 138**

Which statement is true regarding subqueries?

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

**Correct Answer:** D

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Using the ANY Operator in Multiple-Row Subqueries**

The ANY operator (and its synonym, the SOME operator) compares a value to **each** value returned by a subquery.

<ANY means less than the maximum.

>ANY means more than the minimum.

=ANY is equivalent to IN

**Using the ALL Operator in Multiple-Row Subqueries**

The ALL operator compares a value to **every** value returned by a subquery. >ALL means more than the maximum and

<ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators.

**QUESTION 139**

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

1. They can contain a subquery within a subquery.
2. They can return multiple columns as well as rows.
3. They cannot contain a subquery within a subquery.
4. They can return only one column but multiple rows.
5. They can contain group functions and GROUP BY and HAVING clauses.
6. They can contain group functions and the GROUP BY clause, but not the HAVING clause.

**Correct Answer:** ABE

**Section: (none)**

**Explanation**

**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?

1. SELECT prod\_name FROM products

WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products);

1. SELECT prod\_name FROM products

WHERE (unit\_price \* qty\_in\_hand) = (SELECT MAX(unit\_price \* qty\_in\_hand) FROM products GROUP BY prod\_name);

1. 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);

1. 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;

**Correct Answer:** A

**Section: (none)**

**Explanation**

**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?

1. It produces an error.
2. It shows the names of all products in the table.
3. It shows the names of products whose list price is the second highest in the table.
4. It shows the names of all products whose list price is less than the maximum list price.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**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?

1. It produces an error.
2. It executes but returns no rows.
3. It generates output for NULL as well as the other values produced by the subquery.
4. It ignores the NULL value and generates output for the other values produced by the subquery.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**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.**

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 152**

Which statement is true regarding the UNION operator?

1. By default, the output is not sorted.
2. NULL values are not ignored during duplicate checking.
3. Names of all columns must be identical across all SELECT statements.
4. The number of columns selected in all SELECT statements need not be the same.

**Correct Answer:** B

**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?

1. It executes successfully and displays rows in the descending order of PROMO\_CATEGORY.
2. It produces an error because positional notation cannot be used in the ORDER BY clause with SET operators.
3. It executes successfully but ignores the ORDER BY clause because it is not located at the end of the compound statement.
4. 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.

**Correct Answer:** D

**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.)

1. ORDER BY 2,1
2. ORDER BY CUST\_NO
3. ORDER BY 2,cust\_id
4. ORDER BY "CUST\_NO"
5. ORDER BY "Last Name"

**Correct Answer:** ACE

**QUESTION 158**

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

1. It locks only the columns specified in the SELECT list.
2. It locks the rows that satisfy the condition in the SELECT statement.
3. It can be used only in SELECT statements that are based on a single table.
4. It can be used in SELECT statements that are based on a single or multiple tables.
5. After it is enforced by a SELECT statement, no other query can access the same rows until a COMMIT or ROLLBACK is issued.

**Correct Answer:** BD

**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?

1. It executes successfully.
2. It gives an error because a complex view cannot be used to add data into the SALE\_SUMMARY table.
3. It gives an error because the column names in the subquery and the SALE\_SUMMARY table do not match.
4. 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.

**Correct Answer:** D

**Section: (none)**

**Explanation**

**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?

1. It would not execute because two tables cannot be used in a single UPDATE statement.
2. It would not execute because the SELECT statement cannot be used in place of the table name.
3. It would execute and restrict modifications to only the columns specified in the SELECT statement.
4. It would not execute because a subquery cannot be used in the WHERE clause of an UPDATE statement.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**One UPDATE statement can change rows in only one table, but it can change any number of rows in that table.**

**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?

1. 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;

1. 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;

1. 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;

1. 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;

**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**Updating Two Columns with a Subquery**

You can update multiple columns in the SET clause of an UPDATE statement by writing multiple subqueries. The syntax is as follows:

UPDATE **table**

SET **column** =

(SELECT **column**

FROM **table**

WHERE **condition**)

[ ,

**column** =

(SELECT **column**

FROM **table**

WHERE **condition**)]

[WHERE **condition** ] ;

**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?

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

**Correct Answer:** B

**Section: (none)**

**Explanation**

**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?

1. It would not remove the rows if the table has a primary key.
2. It removes all the rows as well as the structure of the table.
3. It removes all the rows in the table and deleted rows can be rolled back.
4. It removes all the rows in the table and deleted rows cannot be rolled back.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**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.)

1. 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');

1. 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');

1. DELETE FROM sales

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

WHERE promo\_name = 'blowout sale' OR promo\_name = 'everyday low price');

1. D DELETE FROM sales

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

WHERE promo\_name IN ('blowout sale','everyday low price'));

**Correct Answer:** BCD

**Section: (none)**

**Explanation**

**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?

1. DELETE FROM products

WHERE prod\_id = (SELECT prod\_id FROM sales

WHERE time\_id - 3\*365 = SYSDATE );

1. DELETE FROM products

WHERE prod\_id = (SELECT prod\_id

FROM sales

WHERE SYSDATE >= time\_id - 3\*365 );

1. DELETE FROM products

WHERE prod\_id IN (SELECT prod\_id FROM sales

WHERE SYSDATE - 3\*365 >= time\_id);

1. DELETE FROM products

WHERE prod\_id IN (SELECT prod\_id FROM sales

WHERE time\_id >= SYSDATE - 3\*365 );

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 170**

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

1. DELETE can be used to remove only rows from only one table at a time.
2. DELETE can be used to remove only rows from multiple tables at a time.
3. DELETE can be used only on a table that is a parent of a referential integrity constraint.
4. DELETE can be used to remove data from specific columns as well as complete rows.
5. DELETE and TRUNCATE can be used on a table that is a parent of a referential integrity constraint having ON DELETE rule .

**Correct Answer:** AE

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Transactions, consisting of INSERT, UPDATE, and DELETE (or even MERGE) commands can be made permanent (with a COMMIT) or reversed (with a ROLLBACK). A TRUNCATE

command, like any other DDL command, is immediately permanent: it can never be reversed.

**The Transaction Control Statements**

A transaction begins implicitly with the first DML statement. There is no command to explicitly start a transaction. The transaction continues through all subsequent DML statements issued by the session. These statements can be against any number of tables: a transaction is not restricted to one table. It terminates (barring any of the events listed in the previous section) when the session issues a COMMIT or ROLLBACK command. The SAVEPOINT command can be used to set markers that will stage the action of a ROLLBACK, but the same transaction remains in progress irrespective of the use of SAVEPOINT

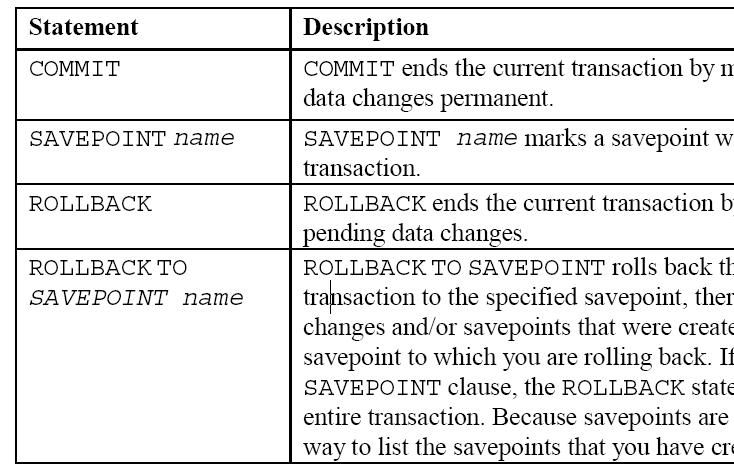
**Explicit Transaction Control Statements**

You can control the logic of transactions by using the COMMIT, SAVEPOINT, and

ROLLBACK

statements.

**Note:** You cannot COMMIT to a SAVEPOINT. SAVEPOINT is not ANSI-standard SQL.



**QUESTION 171**

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

1. COMMIT
2. SELECT
3. CREATE
4. ROLLBACK
5. SAVEPOINT

**Correct Answer:** ACD

**Section: (none)**

**Explanation**

**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.)

1. The rollback generates an error.
2. No SQL statements are rolled back.
3. Only the DELETE statements are rolled back.
4. Only the second DELETE statement is rolled back.
5. Both the DELETE statements and the UPDATE statement are rolled back.

**Correct Answer:** AB

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 173**

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

1. when a DELETE statement is executed
2. when a ROLLBACK command is executed
3. when a PL/SQL anonymous block is executed
4. when a data definition language ( DDL) statement is executed
5. when a TRUNCATE statement is executed after the pending transaction

**Correct Answer:** BDE

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 174**

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

1. A transaction can consist only of a set of DML and DDL statements.
2. A part or an entire transaction can be undone by using ROLLBACK command .
3. A transaction consists of a set of DML or DCL statements.
4. A part or an entire transaction can be made permanent with a COMMIT.
5. A transaction can consist of only a set of queries or DML or DDL statements.

**Correct Answer:** BC

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 175**

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

1. Savepoints are effective only for COMMIT.
2. Savepoints may be used to ROLLBACK.
3. Savepoints can be used for only DML statements.
4. Savepoints are effective for both COMMIT and ROLLBACK.
5. Savepoints can be used for both DML and DDL statements.

**Correct Answer:** BC

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 176**

View the Exhibit; e xamine 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 l date.

Which query would achieve the required result?

**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 .

**Correct Answer:** AE

**QUESTION 171**

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

A. COMMIT

B. SELECT

C. CREATE

D. ROLLBACK

E. SAVEPOINT

**Correct Answer:** ACD

**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

**Correct Answer:** BDE

**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.

**Correct Answer:** BC

**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.

**Correct Answer:** BC

**QUESTION 180**

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.

**Correct Answer:** AD