



# Oracle

## 1Z0-061 Exam

### Oracle Database 12c: SQL Fundamentals Exam

Thank you for Downloading 1Z0-061 exam PDF Demo

You can Buy Latest 1Z0-061 Full Version Download

<https://www.certkillers.net/Exam/1Z0-061>

<https://www.certkillers.net>

## Version: 9.0

---

### Question: 1

---

Evaluate the following SQL statement:

```
SQL> SELECT promo_id, promo_category
FROM promotions
WHERE promo_category = 'Internet' ORDER BY 2 DESC
UNION
SELECT promo_id, promo_category
FROM promotions
WHERE promo_category = 'TV'
UNION
SELECT promo_id, promo_category
FROM promotions
WHERE promo_category = 'Radio';
```

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

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

---

**Answer: D**

---

---

### Question: 2

---

View the Exhibit and examine the structure of the product, component, and PDT\_COMP tables.

In product table, PDTNO is the primary key.

In component table, COMPNO is the primary key.

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

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

Evaluate the following query:

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

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

### PRODUCT

Name	Null?	Type
PDTNO	NOT NULL	NUMBER(3)
PDTNAME		VARCHAR2(25)
QTY		NUMBER(6,2)

### COMPONENT

Name	Null?	Type
COMPNO	NOT NULL	NUMBER(4)
COMPNAME		VARCHAR2(25)
QTY		NUMBER(6,2)

### PDT\_COMP

Name	Null?	Type
PDTNO	NOT NULL	NUMBER(2)
COMPNO	NOT NULL	NUMBER(3)

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

**Answer: C**

### Question: 3

View the Exhibit for the structure of the student and faculty tables.

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

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

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

Examine the following two SQL statements:

### Statement 1

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

### Statement 2

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

Which statement is true regarding the outcome?

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

---

**Answer: D**

---

**Question: 4**

View the Exhibits and examine products and sales tables.

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

Table SALES		
Name	Null?	Type
PROD_ID	NOT NULL	NUMBER
CUST_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
CHANNEL_ID	NOT NULL	NUMBER
PROMO_ID	NOT NULL	NUMBER
QUANTITY_SOLD	NOT NULL	NUMBER(10,2)

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

```
SQL>SELECT p.prod_name, i.item_cnt
      FROM (SELECT prod_id, COUNT(*) item_cnt
            FROM sales
            GROUP BY prod_id) i RIGHT OUTER JOIN products p
      ON i.prod_id = p.prod_id;
```

What happens when the above statement is executed?

- A. The statement executes successfully and produces the required output.
- B. The statement produces an error because item\_cnt cannot be displayed in the outer query.
- C. The statement produces an error because a subquery in the from clause and outer-joins cannot be used together.

D. The statement produces an error because the group by clause cannot be used in a subquery in the from clause.

---

**Answer: A**

---

---

**Question: 5**

---

You want to create a table employees in which the values of columns EMPLOYEES\_ID and LOGIN\_ID must be unique and not null. Which two SQL statements would create the required table?

CertKillers.net

- A) CREATE TABLE employees(  
employee\_id NUMBER,  
login\_id NUMBER,  
employee\_name VARCHAR2(25),  
hire\_date DATE,  
CONSTRAINT emp\_id\_pk PRIMARY KEY (employee\_id, login\_id));
- B) CREATE TABLE employees(  
employee\_id NUMBER CONSTRAINT emp\_id\_pk PRIMARY KEY,  
login\_id NUMBER UNIQUE,  
employee\_name VARCHAR2(25),  
hire\_date DATE);
- C) CREATE TABLE employees(  
employee\_id NUMBER,  
login\_id NUMBER,  
employee\_name VARCHAR2(100),  
hire\_date DATE,  
CONSTRAINT emp\_id\_uk UNIQUE (employee\_id, login\_id));
- D) CREATE TABLE employees(  
employee\_id NUMBER,  
login\_id NUMBER,  
employee\_name VARCHAR2(100),  
hire\_date DATE,  
CONSTRAINT emp\_id\_uk UNIQUE (employee\_id, login\_id),  
CONSTRAINT emp\_id\_nn NOT NULL (employee\_id, login\_id));
- E) CREATE TABLE employees(  
employee\_id NUMBER,  
login\_id NUMBER,  
employee\_name VARCHAR2(100),  
hire\_date DATE,  
CONSTRAINT emp\_id\_uk UNIQUE (employee\_id, login\_id),  
CONSTRAINT emp\_id\_nn NOT NULL (employee\_id, login\_id));
- F) CREATE TABLE employees(  
employee\_id NUMBER CONSTRAINT emp\_id\_nn NOT NULL,  
login\_id NUMBER CONSTRAINT login\_id\_nn NOT NULL,  
employee\_name VARCHAR2(100),  
hire\_date DATE,  
CONSTRAINT emp\_num\_id\_uk UNIQUE (employee\_id, login\_id));

- A. Option A  
B. Option B  
C. Option C

- D. Option D
- E. Option E
- F. Option F

---

**Answer: A,F**

---



---

**Question: 6**

---

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

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

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

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

The query generates an error. What is the reason for the error?

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

---

**Answer: C**

---



---

**Question: 7**

---



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 two statements are true regarding the creation and storage of data in the above table structure?

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

---

**Answer: B,C**

---

VARCHAR2(size) Variable-length character data (A maximum size must be specified: minimum size is 1; maximum size is 4,000.)

CHAR [(size)] Fixed-length character data of length size bytes (Default and minimum size is 1; maximum size is 2,000.)

NUMBER [(p, s)] Number having precision p and scale s (Precision is the total number of decimal digits and scale is the number of digits to the right of the decimal point; precision can range from 1 to 38, and scale can range from -84 to 127.)

DATE Date and time values to the nearest second between January 1, 4712 B.C., and December 31, 9999 A.D.

---

### Question: 8

---

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

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

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

- A. DATE

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

---

**Answer: D**

---

INTERVAL DAY TO SECOND stores a period of time in terms of days, hours, minutes, and seconds. This data type is useful for representing the precise difference between two datetime values.

You can perform a number of arithmetic operations on date (DATE), timestamp (TIMESTAMP, TIMESTAMP WITH TIME ZONE, and TIMESTAMP WITH LOCAL TIME ZONE) and interval (INTERVAL DAY TO SECOND and INTERVAL YEAR TO MONTH) data.

– It stores duration of the credit as days

– The format stored is numeric format, and you know that numeric values can be easily added and subtracted with date data type without using conversion functions (i.e. SELECT SYSDATE – 1 FROM DUAL;)

– The interest has to be calculated for the number of days an individual has taken a credit for, so it will be easy to calculate interest by using the interest rate and duration of the the credit which is numeric format.

References:

---

**Question: 9**

---

Which three tasks can be performed using SQL functions built into Oracle Database?

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

---

**Answer: A,B,C**

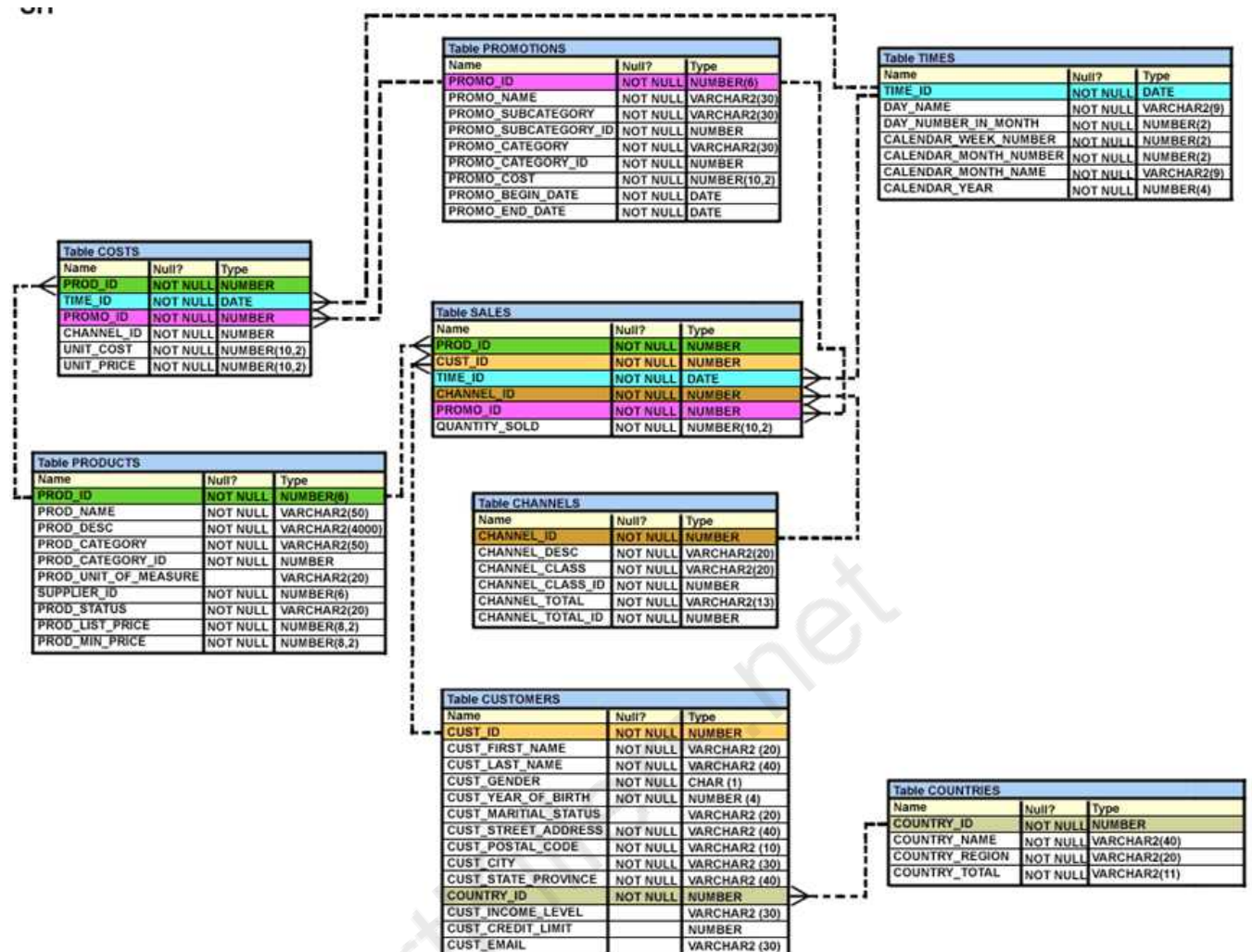
---

---

**Question: 10**

---

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 prices as values. Which three delete statements are valid?

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

- A. Option A  
B. Option B  
C. Option C  
D. Option D

---

**Answer: B,C,D**

---

---

**Question: 11**

---

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

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

- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

---

**Answer: A,D,E**

---

---

**Question: 12**

---

You want to display 5 percent of the employees with the highest salaries in the EMPLOYEES table. Which query will generate the required result?

- A) `SELECT employee_id, last_name, salary  
FROM employees  
ORDER BY salary  
FETCH FIRST 5 PERCENT ROWS ONLY;`
- B) `SELECT employee_id, last_name, salary  
FROM employees  
ORDER BY salary DESC  
FETCH FIRST 5 PERCENT ROWS ONLY;`
- C) `SELECT employee_id, last_name, salary  
FROM employees  
ORDER BY salary DESC  
FETCH FIRST 5 PERCENT ROWS ONLY WITH TIES;`
- D) `SELECT employee_id, last_name, salary  
FROM employees  
ORDER BY salary DESC  
FETCH 5 PERCENT ROWS ONLY;`

- A. Option A  
B. Option B  
C. Option C  
D. Option D

---

**Answer: B**

---

FETCH can only be used with FIRST OR NEXT. The syntax is:  
FETCH { FIRST | NEXT } [integer-literal | ?] {ROW | ROWS} ONLY  
OFFSET { integer-literal | ? } {ROW | ROWS}  
References:

---

**Question: 13**

---

In the customers table, the CUST\_CITY column contains the value 'Paris' for the CUST\_FIRST\_NAME 'Abigail'.  
Evaluate the following query:

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

What would be the outcome?

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

---

**Answer: B**

---



---

**Question: 14**

---

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

CUST_STATUS		
Name	Null?	Type
CUSTNO	NOT NULL	NUMBER(2)
AMT_SPENT		NUMBER(10,2)
CREDIT_LIMIT		NUMBER(10,2)

CUSTNO	AMT_SPENT	CREDIT_LIMIT
1	1000	1000
2	2000	2500
3		3000
4	3000	2800

You issue the following SQL statement:

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

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

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

---

**Answer: C**

---

#### The NULLIF Function

The NULLIF function tests two terms for equality. If they are equal the function returns a null, else it returns the first of the two terms tested.

The NULLIF function takes two mandatory parameters of any data type. The syntax is NULLIF(ifunequal, comparison\_term), where the parameters ifunequal and comparison\_term are compared. If they are identical, then NULL is returned. If they differ, the ifunequal parameter is returned.

---

#### Question: 15

---

Which normal form is a table in if it has no multi-valued attributes and no partial dependencies?

- A. First normal form
- B. Second normal form
- C. Third normal form
- D. Fourth normal form

---

**Answer: B**

---

According to the Second Normal Form (2NF) there must be no partial dependencies on a concatenated key.

---

#### Question: 16

---

Examine the types and examples of relationships that follow:

1. One-to-one a) Teacher to students
2. One-to-many b) Employees to Manager
3. Many-to-one c) Person to SSN
4. Many-to-many d) Customers to products

Which option indicates the correctly matched relationships?

- A. 1-a, 2-b, 3-c, and 4-d
- B. 1-c, 2-d, 3-a, and 4-b
- C. 1-c, 2-a, 3-b, and 4-d
- D. 1-d, 2-b, 3-a, and 4-c

---

**Answer: C**

---



CertKillers.net

## Thank You for trying 1Z0-061 PDF Demo

To Buy Latest 1Z0-061 Full Version Download visit link below

<https://www.certkillers.net/Exam/1Z0-061>

## Start Your 1Z0-061 Preparation

**[Limited Time Offer]** Use Coupon “CKNET” for Further discount on your purchase. Test your 1Z0-061 preparation with actual exam questions.

<https://www.certkillers.net>