1. What is a stored program unit in a database?

a. Table

b. Procedure

c. Trigger

d. View

Answer: b. Procedure

2. Which of the following is not a part of a procedure in a database?

a. Declaration section

b. Exception section

c. Body section

d. Trigger section

Answer: d. Trigger section

3. In parameter modes for procedures, which mode allows you to pass values from the calling program to the procedure and vice versa?

a. IN

b. OUT

c. IN OUT

d. DEFAULT

Answer: c. IN OUT

4. What is one of the advantages of using procedures in a database?

a. Simplified data modeling

b. Enhanced security

c. Dynamic table creation

d. Reduced data redundancy

Answer: b. Enhanced security

5. Which of the following is not a type of trigger in a database?

a. Before Trigger

b. After Trigger

c. During Trigger

d. Instead Of Trigger

Answer: c. During Trigger

6. What is the syntax for creating a trigger in SQL?

a. CREATE PROCEDURE

b. CREATE FUNCTION

c. CREATE TRIGGER

d. CREATE VIEW

Answer: c. CREATE TRIGGER

7. Which type of trigger is fired before the execution of a DML statement in SQL?

a. Before Trigger

b. After Trigger

c. Instead Of Trigger

d. Concurrent Trigger

Answer: a. Before Trigger

8. What is the purpose of a package specification in a database?

a. Contains the implementation details of a package

b. Declares the public interface of a package

c. Stores data for the package

d. Executes procedures in the package

Answer: b. Declares the public interface of a package

9. Which part of a package in a database contains the actual code and implementation details?

a. Package specification

b. Package body

c. Package header

d. Package declaration

Answer: b. Package body

10. What is a "bodiless package" in a database?

a. A package without a package specification

b. A package without a package body

c. A package without any code or implementation

d. A package without parameters

Answer: b. A package without a package body

11. Which of the following is not an advantage of using packages in a database?

a. Encapsulation of code

b. Improved performance

c. Simplified maintenance

d. Increased data redundancy

Answer: d. Increased data redundancy

12. What does a "PRAGMA AUTONOMOUS_TRANSACTION" do in a trigger?

a. Executes the trigger automatically

b. Is used to create a trigger

c. Starts a new transaction within the trigger

d. Stops a running transaction

Answer: c. Starts a new transaction within the trigger

13. Which of the following is a valid parameter mode for a procedure in SQL?

a. IN OUT INCREMENT

b. OUT DECREMENT

c. INCREMENT OUT

d. OUT IN

Answer: a. IN OUT INCREMENT

14. What type of trigger is used to replace a DML statement with another statement?

a. Before Trigger

b. After Trigger

c. Instead Of Trigger

d. Around Trigger

Answer: c. Instead Of Trigger

15. What is the primary purpose of a trigger in a database?

a. To define data types

b. To declare variables

c. To enforce data integrity and automate actions

d. To create views

Answer: c. To enforce data integrity and automate actions

16. Which of the following is not a valid part of a procedure in SQL?

a. Parameter list

b. Declaration section

c. Exception section

d. Trigger section

Answer: d. Trigger section

17. What is the primary purpose of a package in SQL?

a. To store data

b. To define triggers

c. To group related procedures, functions, and variables

d. To create views

Answer: c. To group related procedures, functions, and variables

18. Which parameter mode allows a procedure to receive values from the calling program but not return values back?

a. IN

b. OUT

c. IN OUT

d. DEFAULT

Answer: a. IN

19. Which of the following is not a type of package in SQL?

a. Standalone package

b. Bodiless package

c. Composite package

d. Nested package

Answer: c. Composite package

20. What is the advantage of using triggers in a database?

a. Improved code encapsulation

b. Enhanced security

c. Simplified package creation

d. Dynamic table creation

Answer: b. Enhanced security

1. What is the primary purpose of a transaction in a DBMS?

A. To retrieve data from the database

B. To update the database

C. To manage database schemas

D. To organize data into tables

Answer: B

2. Which of the following is not a property of a transaction in DBMS?

A. Isolation

B. Atomicity

C. Consistency

D. Transparency

Answer: D

3. Which property of transactions ensures that a transaction's changes are permanent and will survive system failures?

A. Atomicity

B. Consistency

C. Isolation

D. Durability

Answer: D

4. What does the ACID acronym stand for in the context of transactions?

A. Atomicity, Completeness, Isolation, Durability

B. Atomicity, Consistency, Isolation, Durability

C. Availability, Consistency, Isolation, Durability

D. Allotment, Concurrency, Integration, Durability

Answer: B

5. Which of the following is not a common concurrency problem in DBMS?

A. Deadlock

B. Dirty Read

C. Lost Update

D. Normalization

Answer: D

6. What is the primary goal of a concurrency control mechanism in a DBMS?

A. To ensure that transactions are executed concurrently without any restrictions

B. To improve the performance of the database system

C. To prevent conflicts and maintain data consistency in a multi-user environment

D. To reduce the storage space required for the database

Answer: C

7. Which of the following is a common method for achieving isolation in DBMS?

A. Two-Phase Locking

B. Time-based Synchronization

C. Data Duplication

D. Data Sharding

Answer: A

8. What is a serializable schedule in the context of concurrency control?

A. A schedule in which transactions are executed one after the other

B. A schedule that preserves the original order of transactions

C. A schedule that produces the same result as if transactions were executed serially

D. A schedule that allows concurrent execution without any restrictions

Answer: C

9. Which of the following is a benefit of using serializability in a DBMS?

A. Improved system performance

B. Reduced data consistency

C. Enhanced data integrity

D. Faster query processing

Answer: C

10. What is the primary purpose of a lock in a DBMS?

A. To restrict access to specific data items

B. To encrypt data for security

C. To optimize query execution

D. To permanently delete data

Answer: A

11. In a DBMS, what is a shared lock used for?

A. To allow multiple transactions to write to the same data item

B. To prevent multiple transactions from reading the same data item simultaneously

C. To allow multiple transactions to read the same data item simultaneously

D. To break deadlocks

Answer: C

12. Which concurrency control technique uses a timeout mechanism to resolve conflicts?

A. Two-Phase Locking

B. Timestamp-Based Protocol

C. Optimistic Concurrency Control

D. Strict Two-Phase Locking

Answer: C

13. What is a deadlock in the context of concurrency control?

A. A situation where two transactions are waiting for each other to release locks

B. A situation where a transaction is permanently blocked

C. A situation where transactions cannot be rolled back

D. A situation where transactions cannot be committed

Answer: A

14. What is the purpose of a deadlock detection mechanism in a DBMS?

A. To prevent deadlocks from occurring

B. To identify and resolve deadlocks

C. To escalate conflicts between transactions

D. To increase the isolation level

Answer: B

15. Which of the following is not a common deadlock prevention technique?

A. Wait-Die

B. Wound-Wait

C. Timeout

D. Rollback

Answer: C

16. What is the purpose of an intent lock in a DBMS?

A. To indicate the intention of a transaction to acquire a shared lock

B. To indicate the intention of a transaction to acquire an exclusive lock

C. To prevent deadlocks

D. To release all locks

Answer: B

17. In a DBMS, what is the purpose of a transaction log?

A. To record all user queries

B. To store database schema information

C. To maintain a record of all committed and uncommitted transactions

D. To store backup copies of data

Answer: C

18. What is the primary goal of a checkpoint in a DBMS?

A. To initiate a transaction rollback

B. To recover from system crashes

C. To release all locks held by a transaction

D. To optimize query execution

Answer: B

19. Which of the following is an example of a conflict-serializable schedule in a DBMS?

A. Schedule S1: T1 → T2 → T3

B. Schedule S2: T2 → T1 → T3

C. Schedule S3: T1 → T3 → T2

D. Schedule S4: T3 → T1 → T2

Answer: A

20. What does the isolation level "Serializable" in a DBMS ensure?

A. It allows dirty reads

B. It provides the highest level of isolation

C. It allows transactions to write to the same data simultaneously

D. It does not allow any concurrency

Answer: B

21. Which of the following is a benefit of using a lower isolation level, such as "Read Uncommitted," in a DBMS?

A. Improved data integrity

B. Higher isolation between transactions

C. Reduced contention for locks

D. Faster query performance

Answer: D

22. In the context of locking, what is a lock mode?

A. The time duration for which a lock is held

B. The type of lock (shared or exclusive) and its compatibility with other locks

C. The order in which locks are acquired

D. The number of transactions waiting for a lock

Answer: B

23. Which of the following is a drawback of using a high isolation level, such as "Serializable," in a DBMS?

A. Increased likelihood of deadlocks

B. Improved data consistency

C. Lower transaction throughput

D. Reduced data integrity

Answer: C

24. What is the purpose of a transaction manager in a DBMS?

A. To optimize query execution

B. To manage database schemas

C. To ensure the ACID properties of transactions

D. To store backup copies of data

Answer: C

25. What is the primary goal

of a deadlock prevention technique like "Wait-Die"?

A. To escalate conflicts between transactions

B. To prevent transactions from waiting indefinitely

C. To improve query performance

D. To increase data redundancy

Answer: B

26. In a DBMS, what is a transaction's isolation level?

A. The number of locks acquired by the transaction

B. The duration for which a transaction is active

C. The level of visibility a transaction has into other transactions' changes

D. The number of concurrent transactions

Answer: C

27. Which of the following is a disadvantage of using optimistic concurrency control?

A. Higher contention for locks

B. Increased likelihood of deadlocks

C. Slower query performance

D. Limited data consistency

Answer: D

28. What is the purpose of a timestamp in a DBMS?

A. To record the time when a transaction started

B. To ensure data encryption

C. To prevent conflicts between transactions

D. To optimize query execution

Answer: A

29. What is a conflict-serializable schedule?

A. A schedule that contains conflicts between transactions

B. A schedule in which transactions are executed serially

C. A schedule that preserves the original order of transactions

D. A schedule that is equivalent to a serial schedule with the same transactions

Answer: D

30. Which of the following is not a common concurrency control mechanism in a DBMS?

A. Two-Phase Commit

B. Optimistic Concurrency Control

C. Strict Two-Phase Locking

D. Timestamp-Based Protocol

Answer: A

31. What is the purpose of a transaction ID in a DBMS?

A. To identify the user who initiated the transaction

B. To indicate the transaction's priority

C. To uniquely identify and track each transaction

D. To store backup copies of data

Answer: C

32. Which of the following is a benefit of using a higher isolation level, such as "Serializable," in a DBMS?

A. Improved query performance

B. Reduced likelihood of deadlocks

C. Higher data consistency

D. Lower transaction throughput

Answer: C

33. What is the primary goal of a timeout-based deadlock prevention technique?

A. To prevent transactions from waiting indefinitely

B. To prioritize transactions based on their importance

C. To escalate conflicts between transactions

D. To increase the isolation level

Answer: A

34. What does a "lost update" refer to in the context of concurrency control?

A. A situation where a transaction is permanently blocked

B. A situation where a transaction is rolled back

C. A situation where one transaction overwrites the changes made by another transaction

D. A situation where transactions cannot be committed

Answer: C

35. Which of the following is not a common technique for deadlock detection in a DBMS?

A. Wait-Die

B. Wound-Wait

C. Timeout

D. Rollback

Answer: D

36. What is the primary purpose of a deadlock prevention technique like "Wound-Wait"?

A. To escalate conflicts between transactions

B. To prevent transactions from waiting indefinitely

C. To improve query performance

D. To increase data redundancy

Answer: A

37. Which of the following statements about the "Repeatable Read" isolation level in a DBMS is true?

A. It allows dirty reads.

B. It allows lost updates.

C. It prevents phantom reads.

D. It has the lowest isolation level.

Answer: C

38. What is the primary goal of a transaction recovery manager in a DBMS?

A. To escalate conflicts between transactions

B. To optimize query execution

C. To ensure the durability of transactions

D. To manage database schemas

Answer: C

39. Which of the following is not a common cause of deadlocks in a DBMS?

A. Circular Wait

B. Resource Preemption

C. Hold and Wait

D. No Concurrency

Answer: D

40. What is the purpose of a data dictionary in a DBMS?

A. To store user data

B. To maintain a log of transactions

C. To manage database schemas and metadata

D. To perform data encryption

Answer: C

Assuming we have a "Bank" table with the following sample data:

account_number	account_holder	balance
1001	John Doe	5000.00
1002	Jane Smith	7500.50
1003	Alice Johnson	3000.25

```sql
CREATE TABLE Bank (
    account_number NUMBER PRIMARY KEY,
    account_holder VARCHAR2(100),
    balance NUMBER(10, 2)
);
```

Here's a PL/SQL code snippet based on this data:

PL/SQL Code Snippet:

```plsql
DECLARE
    v_balance NUMBER;
BEGIN
    SELECT balance
    INTO v_balance
    FROM Bank
    WHERE account_holder = 'John Doe';
    DBMS_OUTPUT.PUT_LINE('John Doe\'s Balance: $' || v_balance);
END;
```

Based on above details gives the following question's answer

1. What is the primary purpose of the PL/SQL code snippet?

a) Updates John Doe's account balance

b) Deletes John Doe's account record

c) Retrieves and displays John Doe's account balance

d) Inserts a new account record for John Doe

Answer: c) Retrieves and displays John Doe's account balance

2. What is the data type of the "balance" column in the "Bank" table?

a) String

b) Date

c) Number

d) Boolean

Answer: c) Number

3. Which SQL operation is performed in the PL/SQL code?

a) INSERT

b) DELETE

c) SELECT

d) UPDATE

Answer: c) SELECT

4. What is the purpose of the `INTO` clause in the code?

a) To indicate the end of the PL/SQL block

b) To declare a new variable

c) To specify the source of data for the SELECT statement

d) To define a cursor

Answer: c) To specify the source of data for the SELECT statement

5. What does the `DBMS_OUTPUT.PUT_LINE` statement do in the code?

a) Updates the database records

b) Deletes database records

c) Retrieves data from the database

d) Displays a message in the console

Answer: d) Displays a message in the console

here are 10 multiple-choice questions (MCQs) based on a PL/SQL code snippet

PL/SQL Code Snippet:

```plsql
DECLARE
  v_employee_count NUMBER;
BEGIN
  SELECT COUNT() INTO v_employee_count
  FROM Employees;
  DBMS_OUTPUT.PUT_LINE('Total Employees: ' || v_employee_count);
END;
```

MCQs:

1. What is the primary purpose of the PL/SQL code snippet?

a) Updates employee records

b) Deletes employee records

c) Retrieves and displays the total number of employees

d) Inserts a new employee record

Answer: c) Retrieves and displays the total number of employees

2. In the code snippet, what is the value stored in the `v_employee_count` variable?

a) Employee names

b) Employee IDs

c) Total number of employees

d) Employee salaries

Answer: c) Total number of employees

3. Which SQL operation is performed in the PL/SQL code?

a) INSERT

b) DELETE

c) SELECT

d) UPDATE

Answer: c) SELECT

4. What is the purpose of the `INTO` clause in the code?

a) To indicate the end of the PL/SQL block

b) To declare a new variable

c) To specify the source of data for the SELECT statement

d) To define a cursor

Answer: c) To specify the source of data for the SELECT statement

5. What does the `DBMS_OUTPUT.PUT_LINE` statement do in the code?

a) Updates the database records

b) Deletes database records

c) Retrieves data from the database

d) Displays a message in the console

Answer: d) Displays a message in the console

6. Which PL/SQL construct allows you to handle exceptions in a structured manner?

a) TRY-CATCH

b) EXCEPTION

c) ERROR-HANDLER

d) ON-ERROR

Answer: b) EXCEPTION

7. In PL/SQL, what is the primary purpose of a cursor?

a) To define variables

b) To loop through a result set

c) To declare procedures

d) To manage transactions

Answer: b) To loop through a result set

8. What is the expected output of the code snippet if there are 100 employees in the "Employees" table?

a) Total Employees: 100

b) Total Employees: 0

c) Total Employees: 1

d) Total Employees: 99

Answer: a) Total Employees: 100

9. What type of variable is `v_employee_count` in the code snippet?

a) String

b) Date

c) Number

d) Boolean

Answer: c) Number

10. In PL/SQL, how can you pass a parameter to a stored procedure?

a) Using a RETURN statement

b) Using a SELECT statement

c) Using an IN parameter

d) Using a WHERE clause

Answer: c) Using an IN parameter

Here's a PL/SQL package with a "College" table and some basic code snippets :

~~~
```sql

-- Create the College table

CREATE TABLE College (

   student_id NUMBER PRIMARY KEY,

   student_name VARCHAR2(50),

   major VARCHAR2(50)

);


+------------+--------------+-------------------+

| student_id | student_name |      major       |

+------------+--------------+-------------------+

|     1      |  John Smith  | Computer Science |

|     2      |   Jane Doe   |     Biology      |

|     3      | Alice Johnson|     History      |

|     4      |   Bob Brown  |   Mathematics    |

|     5      |  Eva Williams|    Chemistry     |

+------------+--------------+-------------------+
~~~

~~~
```plsql

-- Create a PL/SQL package
CREATE OR REPLACE PACKAGE College_Package AS

   -- Function to retrieve student count by major
   FUNCTION getStudentCountByMajor(major IN VARCHAR2) RETURN NUMBER;

   FUNCTION mcq1 RETURN VARCHAR2;

   FUNCTION mcq2 RETURN NUMBER;

END College_Package;
/

CREATE OR REPLACE PACKAGE BODY College_Package AS
~~~

```
  -- Function to retrieve student count by major
  FUNCTION getStudentCountByMajor(major IN VARCHAR2) RETURN NUMBER IS
    cnt NUMBER;
  BEGIN
    SELECT COUNT() INTO cnt FROM College WHERE major = major;
    RETURN cnt;
  END;

  FUNCTION mcq1 RETURN VARCHAR2 IS
  BEGIN
    RETURN 'student_id';
  END;

  FUNCTION mcq2 RETURN NUMBER IS

    biology_count NUMBER;

  BEGIN

    biology_count := getStudentCountByMajor('Biology');

    RETURN biology_count;

  END;


END College_Package;

/
```

MCQ 1: Which column is used to uniquely identify students?

A) student_id

B) student_name

C) major

D) None of the above

Answer: A) student_id

MCQ 2: How many students are majoring in Computer Science?

A) 1

B) 2

C) 3

D) 0

Answer: A) 1

MCQ 3: What is the data type of the "student_name" column in the College table?

A) NUMBER

B) VARCHAR2

C) DATE

D) BOOLEAN

Answer: B) VARCHAR2

MCQ 4: Which PL/SQL construct is used to loop through records in a result set?

A) FOR loop

B) IF statement

C) WHILE loop

D) CASE statement

Answer: A) FOR loop

MCQ 5: How many students are majoring in Chemistry?

A) 1

B) 2

C) 3

D) 0

Answer: D) 0

MCQ 6: Which PL/SQL keyword is used to declare a variable?

A) DEFINE

B) DECLARE

C) VARIABLE

D) SET

Answer: B) DECLARE

MCQ 7: What is the output of the following PL/SQL code?

```plsql
DECLARE
   total_students NUMBER;
BEGIN
   total_students := College_Package.getStudentCountByMajor('Computer Science');
   DBMS_OUTPUT.PUT_LINE('Total students in Computer Science: ' || total_students);
END;
```

A) Total students in Computer Science: 1

B) Total students in Computer Science: 2

C) Total students in Computer Science: 3

D) Total students in Computer Science: 0

Answer: A) Total students in Computer Science: 1

MCQ 8: Which PL/SQL statement is used to raise an exception?

A) RAISE

B) THROW

C) EXCEPTION

D) ERROR

Answer: A) RAISE

MCQ 9: What is the purpose of the PRIMARY KEY constraint in the College table?

A) It enforces unique values in the "student_name" column.

B) It enforces unique values in the "major" column.

C) It ensures that the "student_id" column is not null.

D) It uniquely identifies each row in the table.

Answer: D) It uniquely identifies each row in the table.

MCQ 10: Which PL/SQL construct is used to handle exceptions in a controlled manner?

A) TRY...CATCH block

B) EXCEPTION block

C) ERROR block

D) HANDLE block

Answer: B) EXCEPTION block

Here's a PL/SQL code snippet for a hypothetical "hospital" table, along with 10 multiple-choice questions (MCQs)

Let's create a PL/SQL trigger for the "hospital" table. This trigger updates the "patient_count" column in a separate "hospital_stats" table whenever a new patient is inserted into the "hospital" table.

```sql
-- Create the hospital_stats table to store statistics.
CREATE TABLE hospital_stats (
   total_patients NUMBER
);

-- Create a sequence to generate unique IDs for each patient.
CREATE SEQUENCE patient_id_seq START WITH 1;

-- Create the hospital table.
CREATE TABLE hospital (
   patient_id NUMBER PRIMARY KEY,
   patient_name VARCHAR2(50),
   admission_date DATE,
   discharge_date DATE
);

-- Create the trigger to update patient count in hospital_stats.
```

```
CREATE OR REPLACE TRIGGER update_patient_count
AFTER INSERT ON hospital
FOR EACH ROW
BEGIN
    UPDATE hospital_stats
    SET total_patients = total_patients + 1;
END;
/
```

Multiple-Choice Questions (MCQs):

1. What is the purpose of the "update_patient_count" trigger in the "hospital" table?

a) To automatically update all patient records.

b) To update the total count of patients in the "hospital_stats" table when a new patient is inserted.

c) To prevent new records from being inserted.

d) To calculate the average length of stay for all patients.

Correct Answer: b

2. In which event(s) will the "update_patient_count" trigger execute?

a) Before inserting a new patient record.

b) After deleting a patient record.

c) Before updating an existing patient record.

d) After inserting a new patient record.

Correct Answer: d

3. What does `AFTER INSERT ON hospital` mean in the trigger definition?

a) The trigger fires before a new patient record is inserted.

b) The trigger fires after a patient record is deleted.

c) The trigger fires after a new patient record is inserted.

d) The trigger fires before an existing patient record is updated.

Correct Answer: c

4. What is the purpose of the "hospital_stats" table in the code snippet?

a) To store patient names.

b) To store admission and discharge dates.

c) To store statistics related to the hospital, such as the total number of patients.

d) To store the patient IDs.

Correct Answer: c

5. How is the "patient_id" assigned in the "hospital" table?

a) Manually entered by the user.

b) Generated automatically using a sequence.

c) Copied from the "patient_id" in the "hospital_stats" table.

d) Set to a constant value.

Correct Answer: b

6. What happens if you attempt to insert a new patient record without specifying values for "patient_name," "admission_date," and "discharge_date"?

a) The trigger inserts default values.

b) The trigger raises an error.

c) The trigger inserts NULL values.

d) The trigger generates random values.

Correct Answer: b

7. Which keyword is used to specify the trigger action timing in PL/SQL?

a) WHEN

b) BEFORE

c) AFTER

d) TRIGGER

Correct Answer: c

8. What is the primary purpose of the `UPDATE hospital_stats SET total_patients = total_patients + 1;` statement in the trigger?

a) To delete a patient record.

b) To insert a new patient record.

c) To update the "patient_count" column in the "hospital_stats" table.

d) To calculate the average length of stay for all patients.

Correct Answer: c

9. Can you have multiple triggers with the same timing (e.g., AFTER INSERT) on the same table?

a) No, only one trigger is allowed per table.

b) Yes, but they must have different names.

c) Yes, and they execute in a random order.

d) No, it will result in an error.

Correct Answer: b

10. What does the `CREATE SEQUENCE patient_id_seq START WITH 1;` statement do in the code snippet?

a) It creates a new table.

b) It defines a new trigger.

c) It creates a sequence for generating unique patient IDs.

d) It initializes the patient ID to 1.

Correct Answer: c

A PL/SQL procedure that takes two numbers as input parameters, adds them together, and then displays the result using dbms_output:

```sql
CREATE OR REPLACE PROCEDURE add_numbers (
   p_num1 IN NUMBER,
   p_num2 IN NUMBER
) AS
   v_result NUMBER;
BEGIN
   -- Perform the addition
   v_result := p_num1 + p_num2;

   -- Display the result
   DBMS_OUTPUT.PUT_LINE('The sum of ' || p_num1 || ' and ' || p_num2 || ' is ' || v_result);
END add_numbers;
/
```

Here's an example of how to call this procedure:

```sql
DECLARE
   num1 NUMBER := 10;
   num2 NUMBER := 20;
BEGIN
   add_numbers(num1, num2);
END;
/
```

This will call the `add_numbers` procedure with `num1` and `num2` as arguments and display the sum.

Based on given pl sql answer the following mcq:

1. What is the purpose of the PL/SQL procedure mentioned in the code snippet?

A. To subtract two numbers.

B. To add two numbers and display the result.

C. To multiply two numbers.

D. To divide two numbers.

Answer: B

2. How many input parameters does the `add_numbers` procedure have?

A. None

B. One

C. Two

D. Three

Answer: C

3. What data type are the input parameters `p_num1` and `p_num2` in the `add_numbers` procedure?

A. VARCHAR2

B. DATE

C. NUMBER

D. BOOLEAN

Answer: C

4. What is the purpose of the `DBMS_OUTPUT.PUT_LINE` statement in the procedure?

A. It calculates the sum of two numbers.

B. It displays the result of the addition.

C. It defines a new variable.

D. It retrieves data from the database.

Answer: B

5. How is the result of the addition operation displayed in the output?

A. Using the PRINT statement

B. Using the RETURN statement

C. Using the DBMS_OUTPUT.PUT_LINE statement

D. Using the DISPLAY statement

Answer: C

6. What should you do to call the `add_numbers` procedure with specific numbers as arguments?

A. Use the CALL statement.

B. Use the SELECT statement.

C. Use the DECLARE block.

D. Use the EXECUTE statement.

Answer: C

7. In the example provided for calling the procedure, what are the values of `num1` and `num2`?

A. num1 = 20, num2 = 10

B. num1 = 10, num2 = 30

C. num1 = 10, num2 = 20

D. num1 = 30, num2 = 10

Answer: C

8. What is the result of calling the `add_numbers` procedure with `num1` and `num2` as arguments in the example?

A. 10

B. 20

C. The sum of 10 and 20 is 30

D. There will be no output.

Answer: C

9. Which SQL statement is used to create a PL/SQL procedure?

A. CREATE PROCEDURE

B. DECLARE PROCEDURE

C. EXECUTE PROCEDURE

D. CALL PROCEDURE

Answer: A

10. What is the purpose of the `DECLARE` block in the example?

A. To define a new variable.

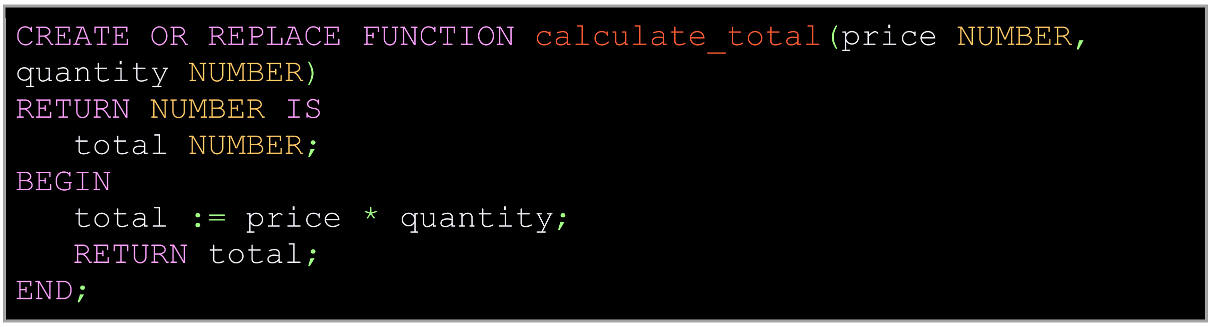
B. To execute SQL statements.

C. To declare and initialize variables before calling the procedure.

D. To declare a function.

Answer: C

Ques1 - Consider the following PL/SQL function: (Difficulty level – Easy)



```
CREATE OR REPLACE FUNCTION calculate_total(price NUMBER,
quantity NUMBER)
RETURN NUMBER IS
   total NUMBER;
BEGIN
   total := price * quantity;
   RETURN total;
END;
```

What does the PL/SQL function `calculate_total` do?

A. It calculates the average of `price` and `quantity`.

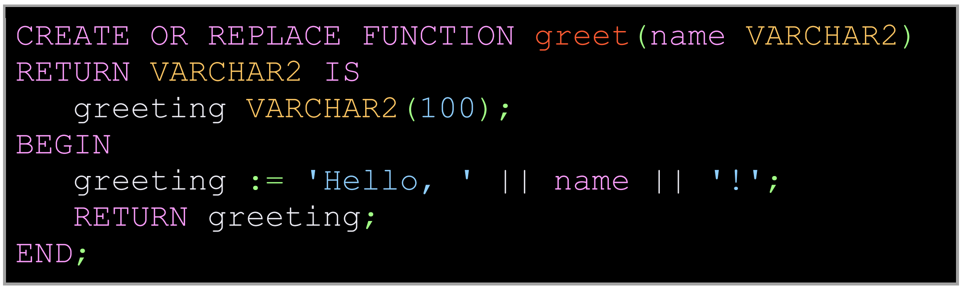
B. It calculates the sum of `price` and `quantity`.

C. It calculates the product of `price` and `quantity`.

D. It calculates the difference between `price` and `quantity`.

Correct Option: C

Ques2 - Consider the following PL/SQL function: (Difficulty level – Easy)



```
CREATE OR REPLACE FUNCTION greet(name VARCHAR2)
RETURN VARCHAR2 IS
   greeting VARCHAR2(100);
BEGIN
   greeting := 'Hello, ' || name || '!';
   RETURN greeting;
END;
```

What does the PL/SQL function `greet` do?

A. It calculates the length of the input string `name`.

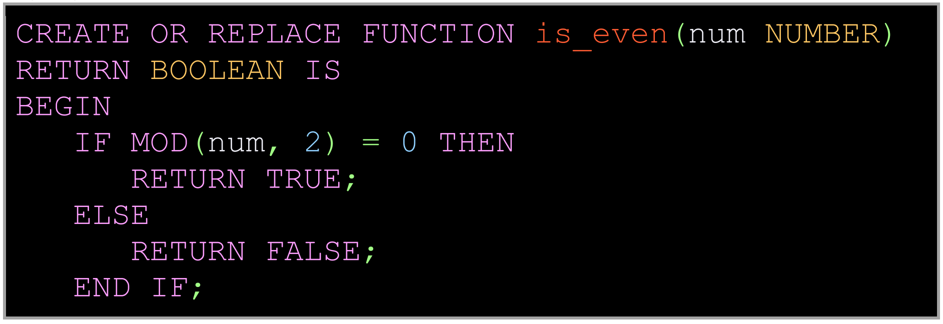
B. It calculates the square of a numeric input.

C. It generates a greeting message with the input `name`.

D. It calculates the factorial of a numeric input.

Correct Option: C

Ques3 - Consider the following PL/SQL function: (Difficulty level – Easy)

```plsql



```
CREATE OR REPLACE FUNCTION is_even(num NUMBER)
RETURN BOOLEAN IS
BEGIN
   IF MOD(num, 2) = 0 THEN
      RETURN TRUE;
   ELSE
      RETURN FALSE;
   END IF;
```
```

```
END;
```

What does the PL/SQL function `is_even` do?

A. It checks if the input `num` is an even number and returns `TRUE` if it is, `FALSE` otherwise.

B. It checks if the input `num` is a positive number and returns `TRUE` if it is, `FALSE` otherwise.

C. It checks if the input `num` is a prime number and returns `TRUE` if it is, `FALSE` otherwise.

D. It calculates the factorial of the input `num`.

Correct Option: A

Ques4 - Consider the following PL/SQL function: (Difficulty level – Easy)

```plsql
CREATE OR REPLACE FUNCTION get_employee_salary(emp_id NUMBER)
RETURN NUMBER IS
   salary NUMBER;
BEGIN
   -- Retrieve the salary of the employee with the given emp_id
   SELECT salary INTO salary FROM employees WHERE employee_id = emp_id;
   RETURN salary;
END;
```

What does the PL/SQL function `get_employee_salary` do?

A. It calculates the average salary of all employees.

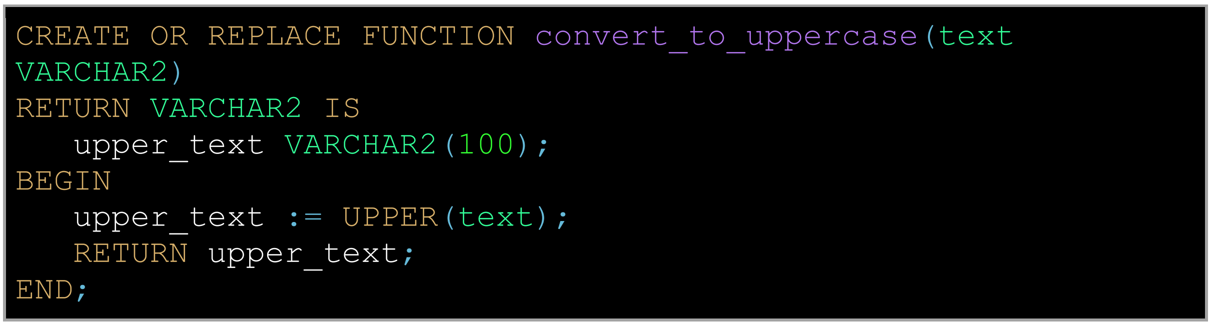
B. It retrieves the salary of the employee with the specified `emp_id`.

C. It calculates the total salary of all employees.

D. It retrieves the highest salary among all employees.

Correct Option: B

Ques5 - Consider the following PL/SQL function: (Difficulty level – Easy)



```
CREATE OR REPLACE FUNCTION convert_to_uppercase(text
VARCHAR2)
RETURN VARCHAR2 IS
   upper_text VARCHAR2(100);
BEGIN
   upper_text := UPPER(text);
   RETURN upper_text;
END;
```

What does the PL/SQL function `convert_to_uppercase` do?

A. It calculates the length of the input `text`.

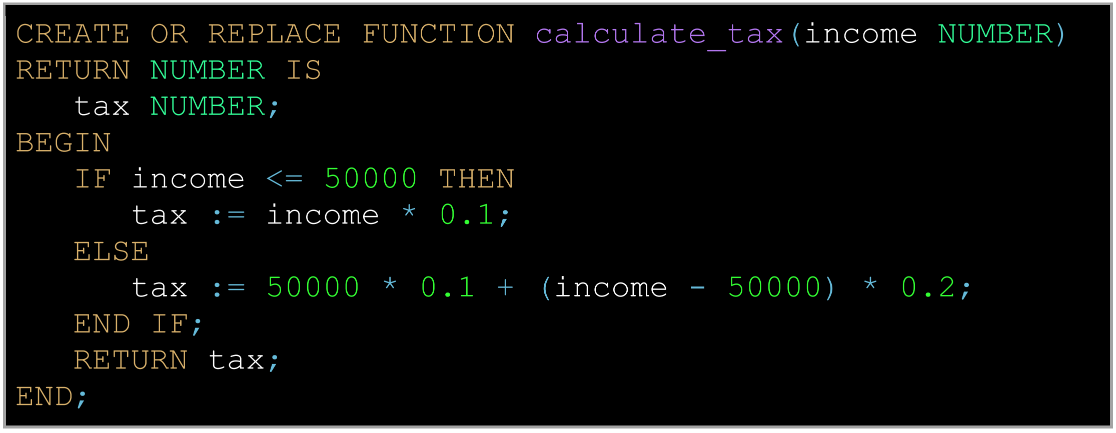
B. It calculates the square of a numeric input.

C. It converts the input `text` to uppercase.

D. It calculates the factorial of a numeric input.

Correct Option: C

Ques6 - Consider the following PL/SQL function: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE FUNCTION calculate_tax(income NUMBER)
RETURN NUMBER IS
   tax NUMBER;
BEGIN
   IF income <= 50000 THEN
      tax := income * 0.1;
   ELSE
      tax := 50000 * 0.1 + (income - 50000) * 0.2;
   END IF;
   RETURN tax;
END;
```

What does the PL/SQL function `calculate_tax` do?

A. It calculates the total income after applying a tax rate.

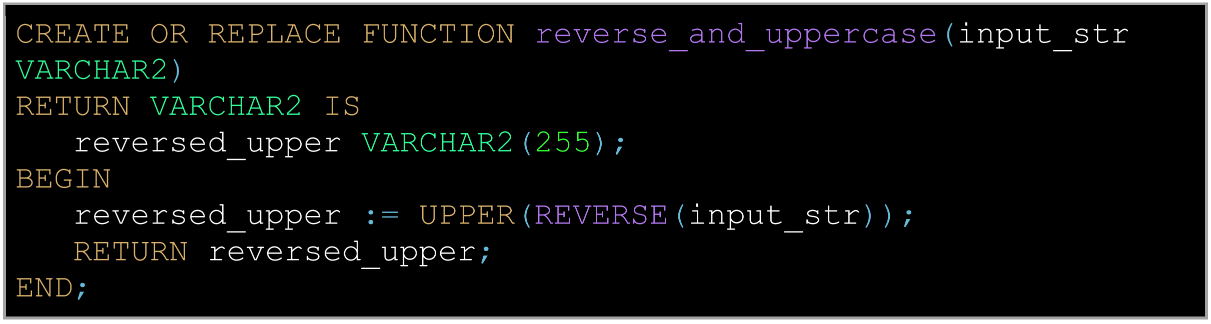
B. It calculates the square root of the input number `income`.

C. It calculates the factorial of the input number `income`.

D. It calculates the tax amount based on the input income.

Correct Option: D

Ques7 - Consider the following PL/SQL function: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE FUNCTION reverse_and_uppercase(input_str
VARCHAR2)
RETURN VARCHAR2 IS
   reversed_upper VARCHAR2(255);
BEGIN
   reversed_upper := UPPER(REVERSE(input_str));
   RETURN reversed_upper;
END;
```

What does the PL/SQL function `reverse_and_uppercase` do?

A. It calculates the length of the input string `input_str`.

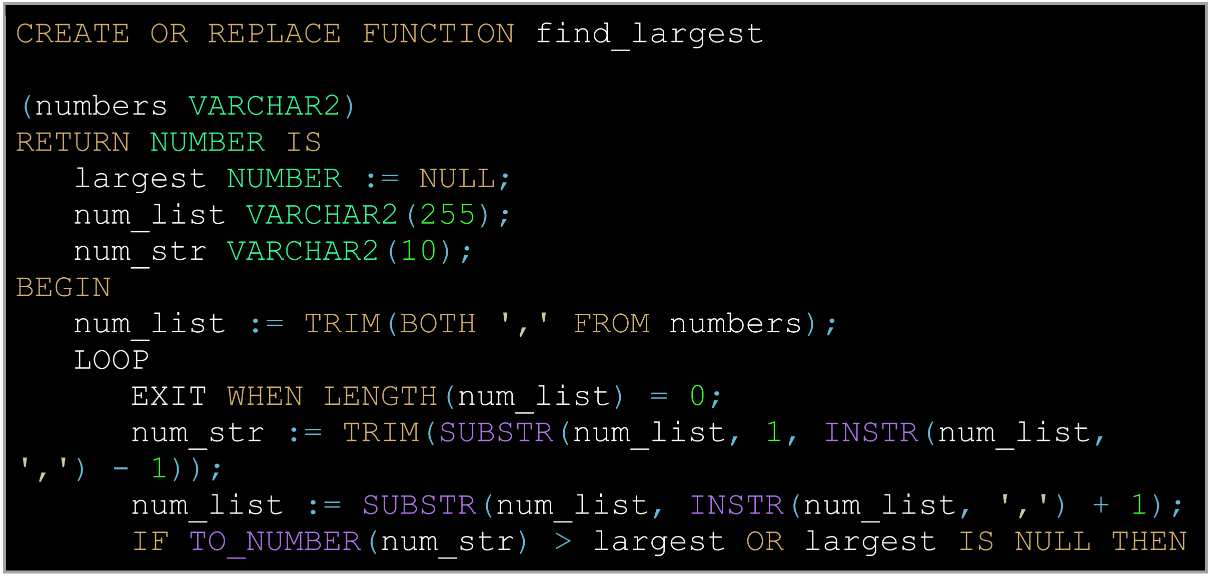
B. It calculates the square of the input number `input_str`.

C. It reverses the characters in the input string `input_str` and converts them to uppercase.

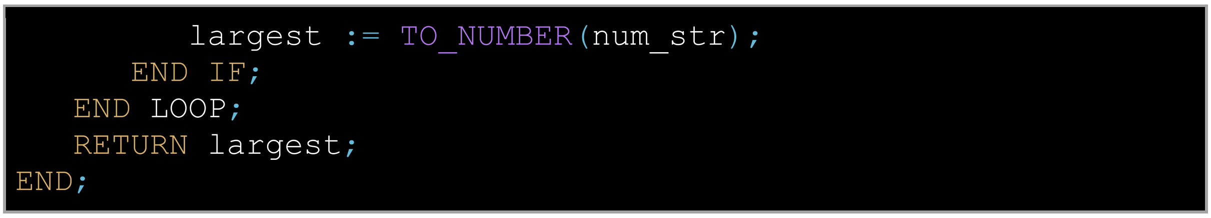
D. It calculates the factorial of the input number `input_str`.

Correct Option: C

Ques8 - Consider the following PL/SQL function: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE FUNCTION find_largest

 (numbers VARCHAR2)
RETURN NUMBER IS
   largest NUMBER := NULL;
   num_list VARCHAR2(255);
   num_str VARCHAR2(10);
BEGIN
   num_list := TRIM(BOTH ',' FROM numbers);
   LOOP
      EXIT WHEN LENGTH(num_list) = 0;
      num_str := TRIM(SUBSTR(num_list, 1, INSTR(num_list,
',') - 1));
      num_list := SUBSTR(num_list, INSTR(num_list, ',') + 1);
      IF TO_NUMBER(num_str) > largest OR largest IS NULL THEN
```



```
            largest := TO_NUMBER(num_str);
        END IF;
    END LOOP;
    RETURN largest;
END;
```

What does the PL/SQL function `find_largest` do?

A. It calculates the square root of the input string `numbers`.

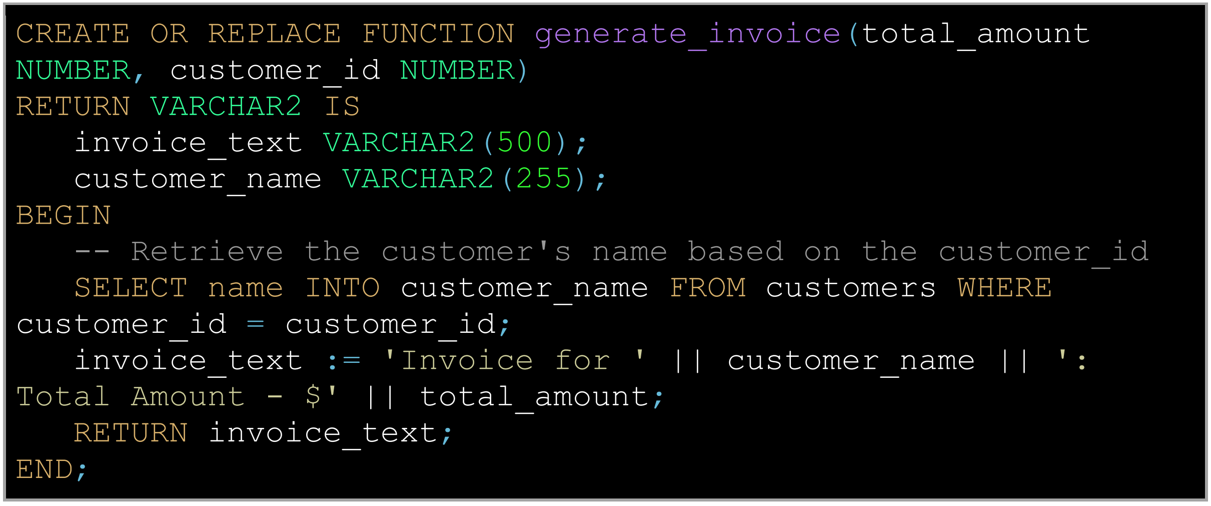
B. It calculates the sum of all numbers in the input string `numbers`.

C. It retrieves the largest number from a comma-separated list of numbers in the input string `numbers`.

D. It calculates the factorial of all numbers in the input string `numbers`.

Correct Option: C

Ques9 - Consider the following PL/SQL function: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE FUNCTION generate_invoice(total_amount
NUMBER, customer_id NUMBER)
RETURN VARCHAR2 IS
    invoice_text VARCHAR2(500);
    customer_name VARCHAR2(255);
BEGIN
    -- Retrieve the customer's name based on the customer_id
    SELECT name INTO customer_name FROM customers WHERE
customer_id = customer_id;
    invoice_text := 'Invoice for ' || customer_name || ':
Total Amount - $' || total_amount;
    RETURN invoice_text;
END;
```

What does the PL/SQL function `generate_invoice` do?

A. It generates an invoice text for a customer with the specified `customer_id` and total amount.

B. It calculates the average total amount for all customers.

C. It retrieves the customer's name based on the customer_id.

D. It calculates the total amount for a customer with the specified `customer_id`.

Correct Option: A

Ques10 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

```plsql
CREATE OR REPLACE PACKAGE product_recommendations AS
   FUNCTION recommend_products(customer_id NUMBER) RETURN
VARCHAR2;
   FUNCTION get_product_rating(product_id NUMBER) RETURN
NUMBER;
   FUNCTION get_product_reviews(product_id NUMBER) RETURN
NUMBER;
END product_recommendations;
/
```

What does the PL/SQL package `product_recommendations` contain?

A. It contains three PL/SQL functions, `recommend_products`, `get_product_rating`, and `get_product_reviews`, for providing product recommendations and retrieving product ratings and reviews.

B. It contains two PL/SQL triggers, `recommend_products`, `get_product_rating`, and `get_product_reviews`, for providing product recommendations and retrieving product ratings and reviews.

C. It contains four PL/SQL procedures, `recommend_products`, `get_product_rating`, and `get_product_reviews`, for providing product recommendations and retrieving product ratings and reviews.

D. It contains one PL/SQL function, `product_recommendations`, and one PL/SQL procedure, `product_recommendations`, for providing product recommendations and retrieving product ratings and reviews.

Correct Option: A

Ques11 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

```plsql
CREATE OR REPLACE PACKAGE inventory_management AS
   FUNCTION check_stock_availability(product_id NUMBER,
warehouse_id NUMBER) RETURN BOOLEAN;
   FUNCTION transfer_product(product_id NUMBER,
source_warehouse_id NUMBER, destination_warehouse_id NUMBER,
quantity NUMBER) RETURN NUMBER;
   FUNCTION get_product_location(product_id NUMBER) RETURN
VARCHAR2;
END inventory_management;
```

/

What does the PL/SQL package `inventory_management` contain?

A. It contains four PL/SQL functions, `check_stock_availability`, `transfer_product`, and `get_product_location`, for managing inventory and retrieving product locations.

B. It contains three PL/SQL triggers, `check_stock_availability`, `transfer_product`, and `get_product_location`, for managing inventory and retrieving product locations.

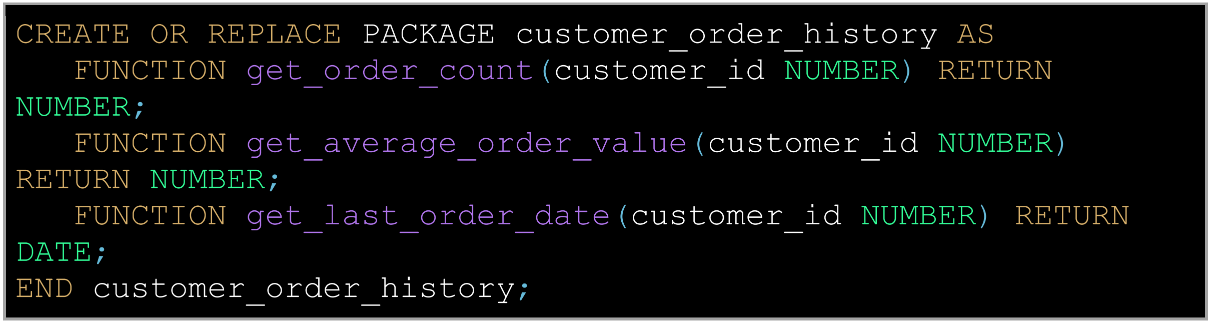
C. It contains two PL/SQL

procedures, `check_stock_availability`, `transfer_product`, and `get_product_location`, for managing inventory and retrieving product locations.

D. It contains one PL/SQL function, `inventory_management`, and one PL/SQL procedure, `inventory_management`, for managing inventory and retrieving product locations.

Correct Option: A

Ques12 - Consider the following PL/SQL package specification: (Difficulty level – Hard)



```plsql
CREATE OR REPLACE PACKAGE customer_order_history AS
   FUNCTION get_order_count(customer_id NUMBER) RETURN NUMBER;
   FUNCTION get_average_order_value(customer_id NUMBER) RETURN NUMBER;
   FUNCTION get_last_order_date(customer_id NUMBER) RETURN DATE;
END customer_order_history;
/
```

What does the PL/SQL package `customer_order_history` contain?

A. It contains three PL/SQL functions, `get_order_count`, `get_average_order_value`, and `get_last_order_date`, for retrieving customer order history statistics.

B. It contains three PL/SQL triggers, `get_order_count`, `get_average_order_value`, and `get_last_order_date`, for retrieving customer order history statistics.

C. It contains three PL/SQL procedures, `get_order_count`, `get_average_order_value`, and `get_last_order_date`, for retrieving customer order history statistics.

D. It contains one PL/SQL function, `customer_order_history`, and one PL/SQL procedure, `customer_order_history`, for retrieving customer order history statistics.

Correct Option: A

Ques13 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

```plsql
CREATE OR REPLACE PACKAGE employee_performance AS
   FUNCTION calculate_performance_rating(employee_id NUMBER,
year NUMBER) RETURN NUMBER;
   FUNCTION get_top_performing_employee(year NUMBER) RETURN
VARCHAR2;
END employee_performance;
/
```

What does the PL/SQL package `employee_performance` contain?

A. It contains two PL/SQL functions, `calculate_performance_rating` and `get_top_performing_employee`, for calculating employee performance ratings and identifying the top-performing employee.

B. It contains one PL/SQL triggers, `calculate_performance_rating` and `get_top_performing_employee`, for calculating employee performance ratings and identifying the top-performing employee.

C. It contains three PL/SQL procedures, `calculate_performance_rating` and `get_top_performing_employee`, for calculating employee performance ratings and identifying the top-performing employee.

D. It contains four PL /SQL function, `employee_performance`, and one PL/SQL procedure, `employee_performance`, for calculating employee performance ratings and identifying the top-performing employee.

Correct Option: A

Ques14 - Consider the following PL/SQL package specification: (Difficulty level – Easy)

```plsql
CREATE OR REPLACE PACKAGE employee_info AS
   FUNCTION get_employee_name(emp_id NUMBER) RETURN VARCHAR2;
   FUNCTION get_employee_salary(emp_id NUMBER) RETURN NUMBER;
END employee_info;
/
```

What does the PL/SQL package `employee_info` contain?

A. It contains four PL/SQL functions, `get_employee_name` and `get_employee_salary`.

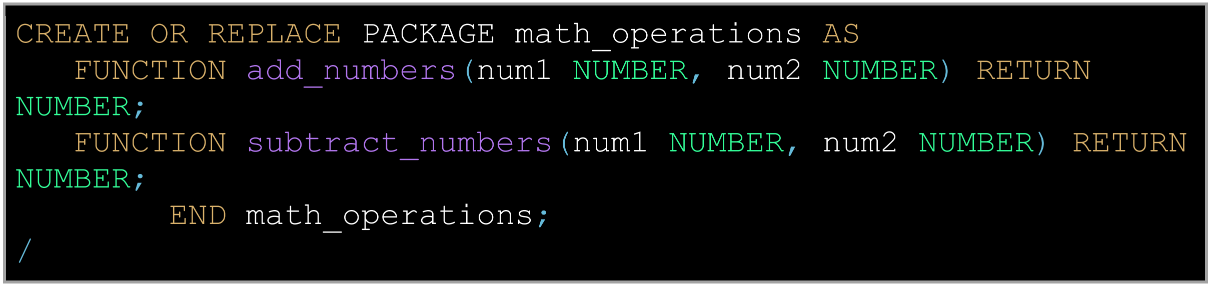
B. It contains PL/SQL triggers, `get_employee_name` and `get_employee_salary`.

C. It contains two PL/SQL procedures, `get_employee_name` and `get_employee_salary`.

D. It contains one PL/SQL function, `employee_info`, and one PL/SQL procedure, `employee_info`.

Correct Option: A

Ques15 - Consider the following PL/SQL package specification: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE PACKAGE math_operations AS
   FUNCTION add_numbers(num1 NUMBER, num2 NUMBER) RETURN
NUMBER;
   FUNCTION subtract_numbers(num1 NUMBER, num2 NUMBER) RETURN
NUMBER;
        END math_operations;
/
```

What does the PL/SQL package `math_operations` contain?

A. It contains two PL/SQL functions, `add_numbers` and `subtract_numbers`, for performing mathematical operations.

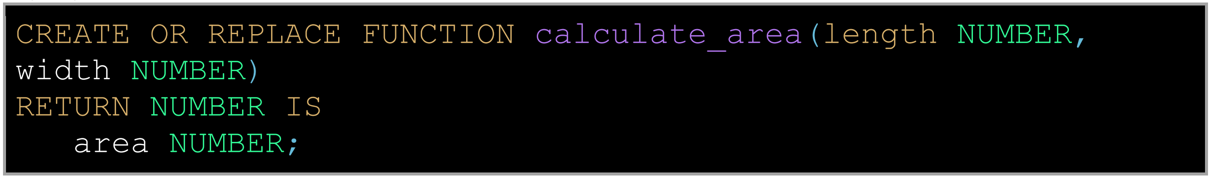
B. It contains two PL/SQL triggers, `add_numbers` and `subtract_numbers`, for performing mathematical operations.

C. It contains two PL/SQL procedures, `add_numbers` and `subtract_numbers`, for performing mathematical operations.

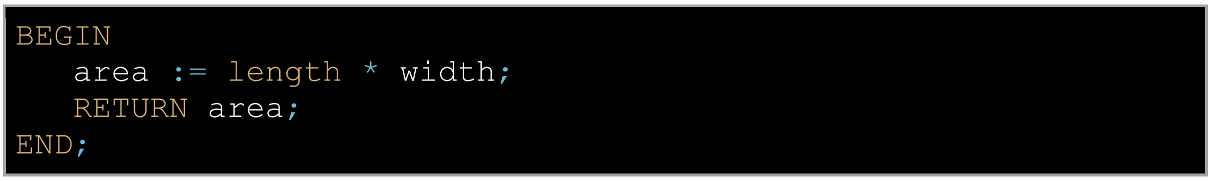
D. It contains one PL/SQL function, `math_operations`, and one PL/SQL procedure, `math_operations`, for performing mathematical operations.

Correct Option: A

Ques1 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION calculate_area(length NUMBER,
width NUMBER)
RETURN NUMBER IS
   area NUMBER;
```



```
BEGIN
    area := length * width;
    RETURN area;
END;
```

What does the PL/SQL function `calculate_area` do?

A. It calculates the perimeter of a rectangle.

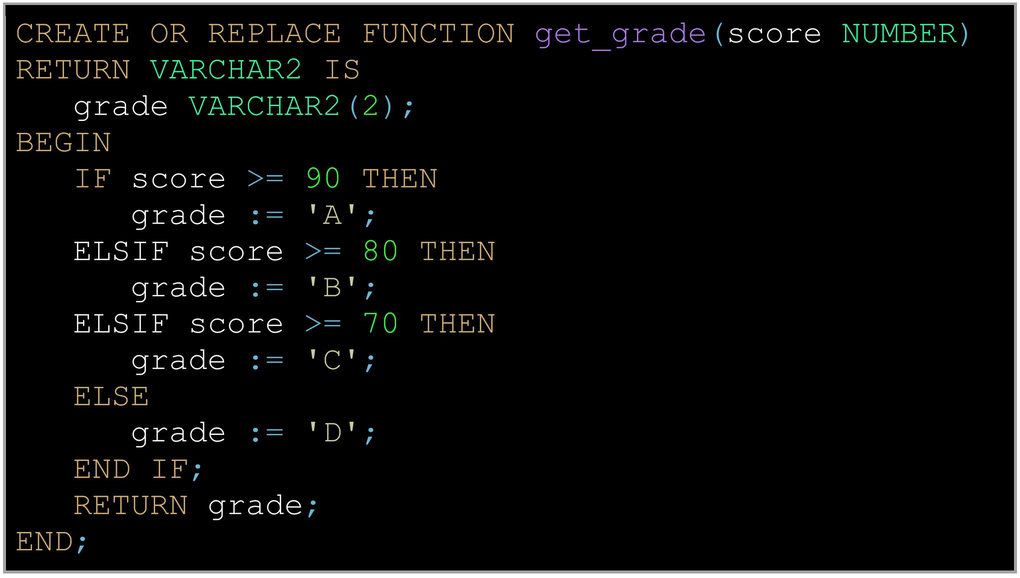
B. It calculates the area of a rectangle.

C. It calculates the volume of a rectangle.

D. It calculates the diagonal length of a rectangle.

Correct Option: B

Ques7 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION get_grade(score NUMBER)
RETURN VARCHAR2 IS
    grade VARCHAR2(2);
BEGIN
    IF score >= 90 THEN
        grade := 'A';
    ELSIF score >= 80 THEN
        grade := 'B';
    ELSIF score >= 70 THEN
        grade := 'C';
    ELSE
        grade := 'D';
    END IF;
    RETURN grade;
END;
```

What does the PL/SQL function `get_grade` do?

A. It calculates the square root of the input `score`.

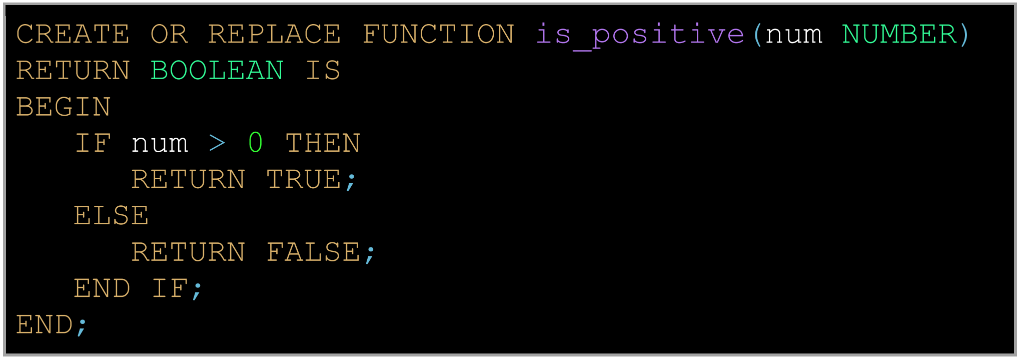
B. It calculates the average of multiple scores.

C. It assigns a grade ('A', 'B', 'C', or 'D') based on the input `score`.

D. It calculates the factorial of the input `score`.

Correct Option: C

Ques2 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION is_positive(num NUMBER)
RETURN BOOLEAN IS
BEGIN
   IF num > 0 THEN
      RETURN TRUE;
   ELSE
      RETURN FALSE;
   END IF;
END;
```

What does the PL/SQL function `is_positive` do?

A. It checks if the input `num` is a positive number and returns `TRUE` if it is, `FALSE` otherwise.

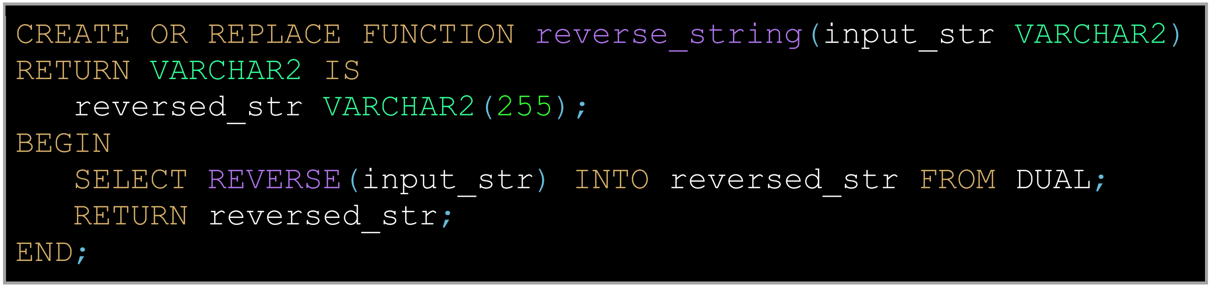
B. It checks if the input `num` is an even number and returns `TRUE` if it is, `FALSE` otherwise.

C. It calculates the square of the input `num`.

D. It calculates the factorial of the input `num`.

Correct Option: A

Ques3 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION reverse_string(input_str VARCHAR2)
RETURN VARCHAR2 IS
   reversed_str VARCHAR2(255);
BEGIN
   SELECT REVERSE(input_str) INTO reversed_str FROM DUAL;
   RETURN reversed_str;
END;
```

What does the PL/SQL function `reverse_string` do?

A. It calculates the length of the input string `input_str`.

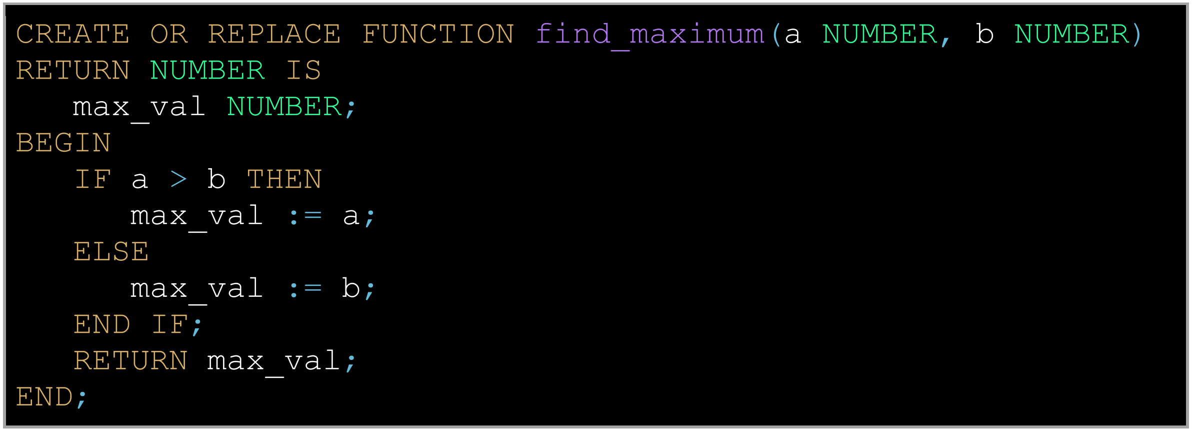
B. It calculates the square root of the input number `input_str`.

C. It reverses the characters in the input string `input_str`.

D. It calculates the factorial of the input number `input_str`.

Correct Option: C

Ques4 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION find_maximum(a NUMBER, b NUMBER)
RETURN NUMBER IS
   max_val NUMBER;
BEGIN
   IF a > b THEN
      max_val := a;
   ELSE
      max_val := b;
   END IF;
   RETURN max_val;
END;
```

What does the PL/SQL function `find_maximum` do?

A. It calculates the average of two numbers.

B. It calculates the sum of two

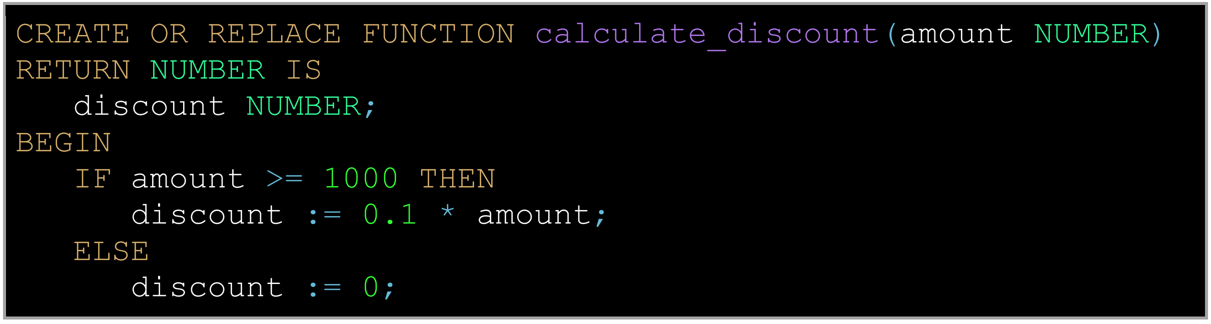
numbers.

C. It calculates the maximum value between two numbers.

D. It calculates the factorial of two numbers.

Correct Option: C

Ques5 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION calculate_discount(amount NUMBER)
RETURN NUMBER IS
   discount NUMBER;
BEGIN
   IF amount >= 1000 THEN
      discount := 0.1 * amount;
   ELSE
      discount := 0;
```

```
    END IF;
    RETURN discount;
END;
```

What does the PL/SQL function `calculate_discount` do?

A. It calculates the total cost after applying a discount of 10%.

B. It calculates the total cost without any discount.

C. It calculates the total cost after applying a discount of 1%.

D. It calculates the total cost after applying a discount of 5%.

Correct Option: A

Ques6 - Consider the following PL/SQL function: (Difficulty level – Easy)

```plsql
CREATE OR REPLACE FUNCTION is_vowel(character CHAR)
RETURN BOOLEAN IS
BEGIN
    IF character IN ('A', 'E', 'I', 'O', 'U', 'a', 'e', 'i',
'o', 'u') THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END;
```

What does the PL/SQL function `is_vowel` do?

A. It checks if the input character is a consonant and returns `TRUE` if it is, `FALSE` otherwise.

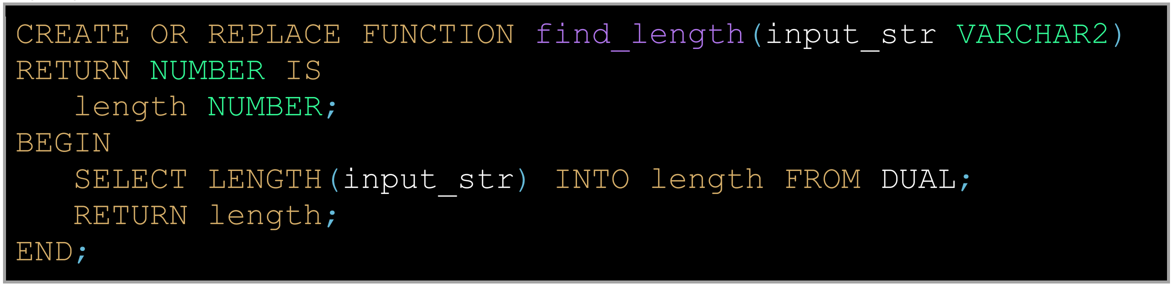
B. It checks if the input character is a digit and returns `TRUE` if it is, `FALSE` otherwise.

C. It checks if the input character is a vowel and returns `TRUE` if it is, `FALSE` otherwise.

D. It calculates the square root of the input character.

Correct Option: C

Ques7 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION find_length(input_str VARCHAR2)
RETURN NUMBER IS
    length NUMBER;
BEGIN
    SELECT LENGTH(input_str) INTO length FROM DUAL;
    RETURN length;
END;
```

What does the PL/SQL function `find_length` do?

A. It calculates the factorial of the length of the input string `input_str`.

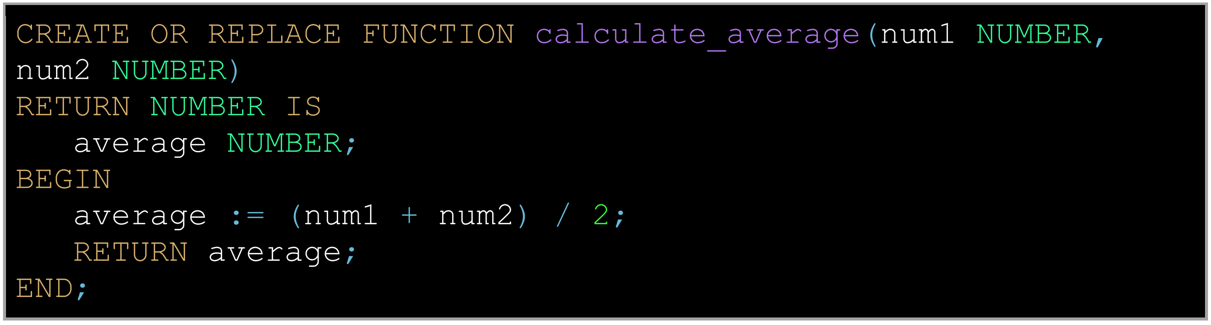
B. It calculates the square root of the length of the input string `input_str`.

C. It retrieves the length of the input string `input_str`.

D. It checks if the length of the input string `input_str` is even and returns `TRUE` if it is, `FALSE` otherwise.

Correct Option: C

Ques8 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION calculate_average(num1 NUMBER,
num2 NUMBER)
RETURN NUMBER IS
    average NUMBER;
BEGIN
    average := (num1 + num2) / 2;
    RETURN average;
END;
```

What does the PL/SQL function `calculate_average` do?

A. It calculates the sum of two numbers.

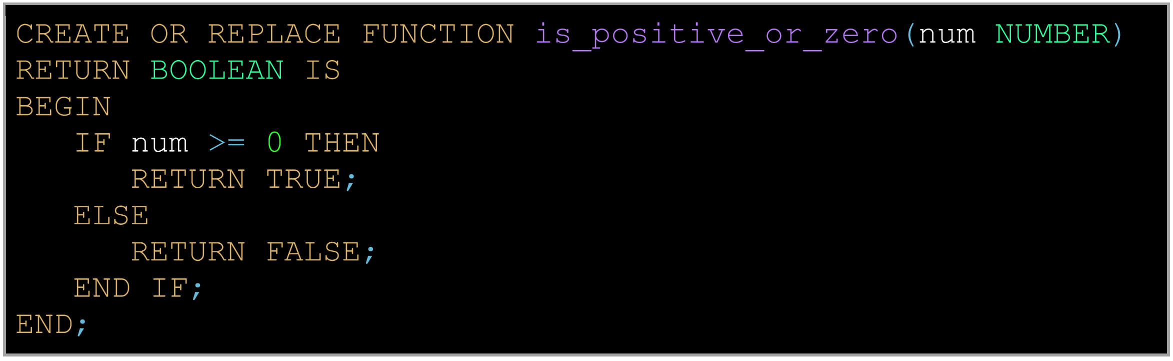
B. It calculates the product of two numbers.

C. It calculates the average of two numbers.

D. It calculates the square root of two numbers.

Correct Option: C

Ques9 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION is_positive_or_zero(num NUMBER)
RETURN BOOLEAN IS
BEGIN
   IF num >= 0 THEN
      RETURN TRUE;
   ELSE
      RETURN FALSE;
   END IF;
END;
```

What does the PL/SQL function `is_positive_or_zero` do?

A. It checks if the input `num` is a positive number and returns `TRUE` if it is, `FALSE` otherwise.

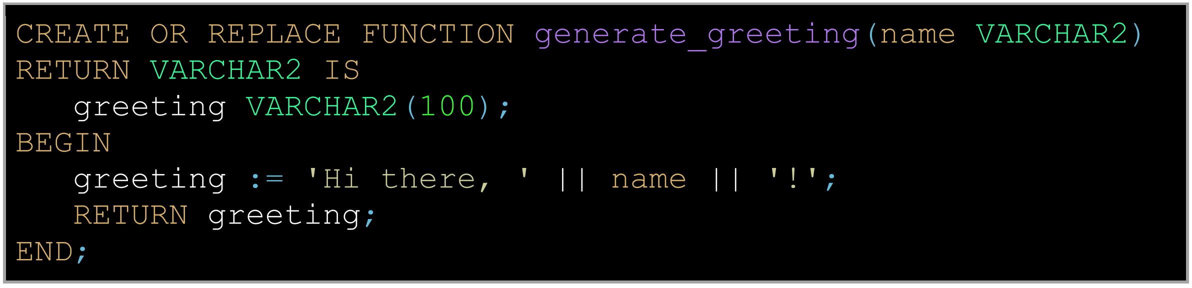
B. It checks if the input `num` is an even number and returns `TRUE` if it is, `FALSE` otherwise.

C. It checks if the input `num` is a non-negative number and returns `TRUE` if it is, `FALSE` otherwise.

D. It calculates the factorial of the input `num`.

Correct Option: C

Ques10 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION generate_greeting(name VARCHAR2)
RETURN VARCHAR2 IS
   greeting VARCHAR2(100);
BEGIN
   greeting := 'Hi there, ' || name || '!';
   RETURN greeting;
END;
```

What does the PL/SQL function `generate_greeting` do?

A. It calculates the length of the input string `name`.

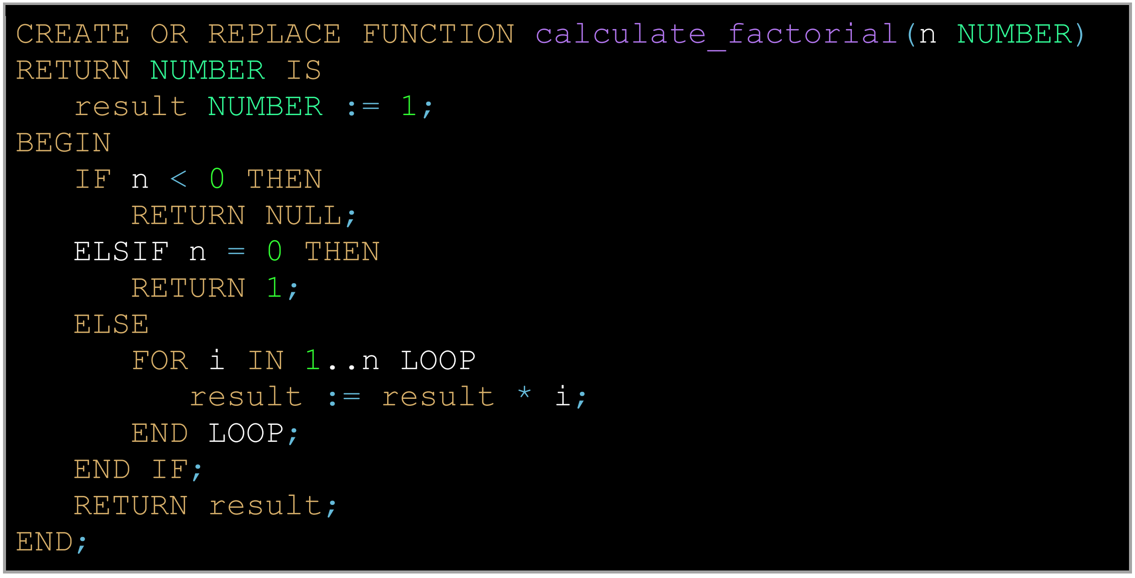
B. It calculates the square of a numeric input.

C. It generates a friendly greeting message with the input `name`.

D. It calculates the factorial of a numeric input.

Correct Option: C

Ques11 - Consider the following PL/SQL function: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE FUNCTION calculate_factorial(n NUMBER)
RETURN NUMBER IS
   result NUMBER := 1;
BEGIN
   IF n < 0 THEN
      RETURN NULL;
   ELSIF n = 0 THEN
      RETURN 1;
   ELSE
      FOR i IN 1..n LOOP
         result := result * i;
      END LOOP;
   END IF;
   RETURN result;
END;
```

What does the PL/SQL function `calculate_factorial` do?

A. It calculates the factorial of a non-negative integer `n`.

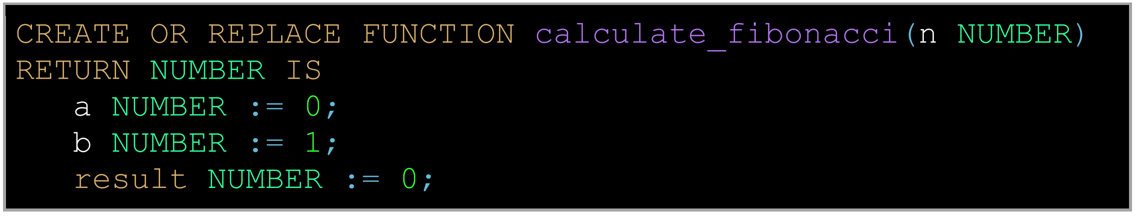
B. It calculates the square root of the input number `n`.

C. It calculates the average of multiple numbers.

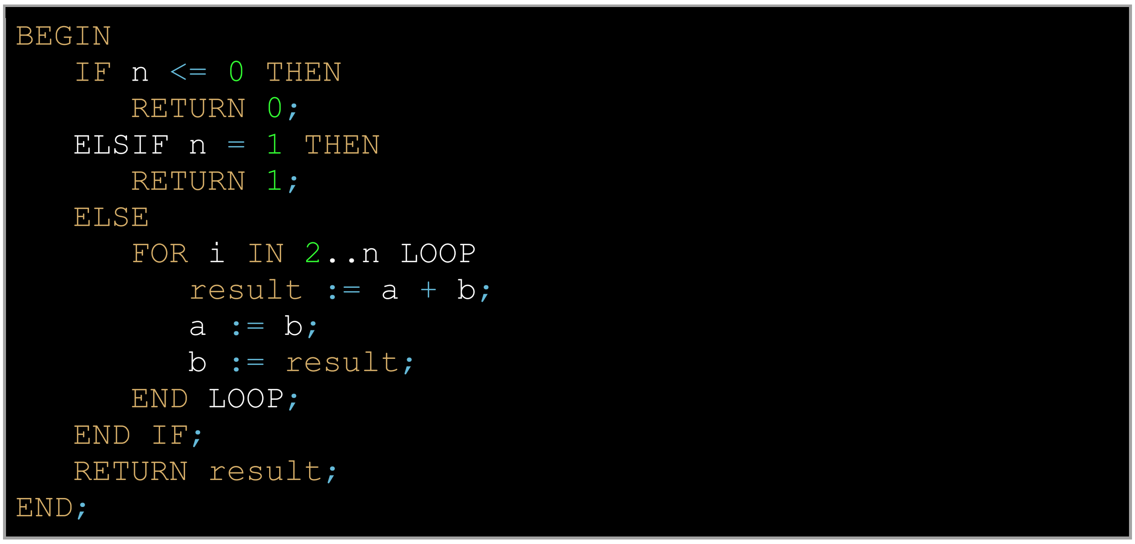
D. It calculates the sum of all integers from 1 to `n`.

Correct Option: A

Ques12 - Consider the following PL/SQL function: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE FUNCTION calculate_fibonacci(n NUMBER)
RETURN NUMBER IS
   a NUMBER := 0;
   b NUMBER := 1;
   result NUMBER := 0;
```



~~~plsql
BEGIN
   IF n <= 0 THEN
      RETURN 0;
   ELSIF n = 1 THEN
      RETURN 1;
   ELSE
      FOR i IN 2..n LOOP
         result := a + b;
         a := b;
         b := result;
      END LOOP;
   END IF;
   RETURN result;
END;
~~~

```

**What does the PL/SQL function `calculate_fibonacci` do?**

A. It calculates the sum of the first `n` Fibonacci numbers.

B. It calculates the square root of the input number `n`.

C. It calculates the factorial of the input number `n`.

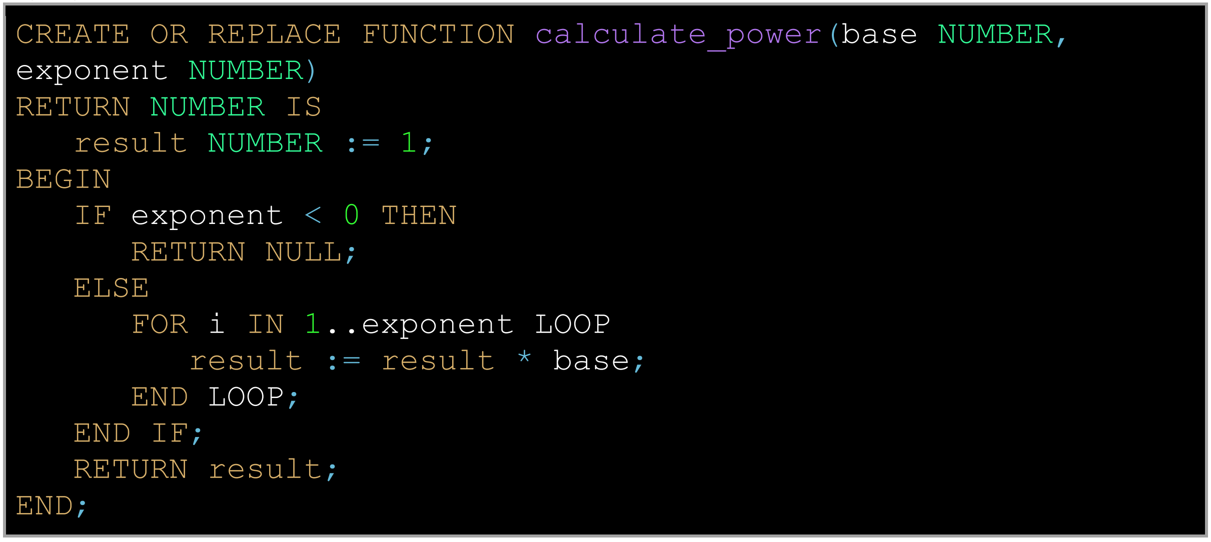
D. It calculates the `n`-th Fibonacci number.

**Correct Option:** D

---

**Ques13 - Consider the following PL/SQL function: (Difficulty level – Medium)**

```plsql



~~~plsql
CREATE OR REPLACE FUNCTION calculate_power(base NUMBER,
exponent NUMBER)
RETURN NUMBER IS
   result NUMBER := 1;
BEGIN
   IF exponent < 0 THEN
      RETURN NULL;
   ELSE
      FOR i IN 1..exponent LOOP
         result := result * base;
      END LOOP;
   END IF;
   RETURN result;
END;
~~~

```

What does the PL/SQL function `calculate_power` do?
```

A. It calculates the product of `base` and `exponent`.

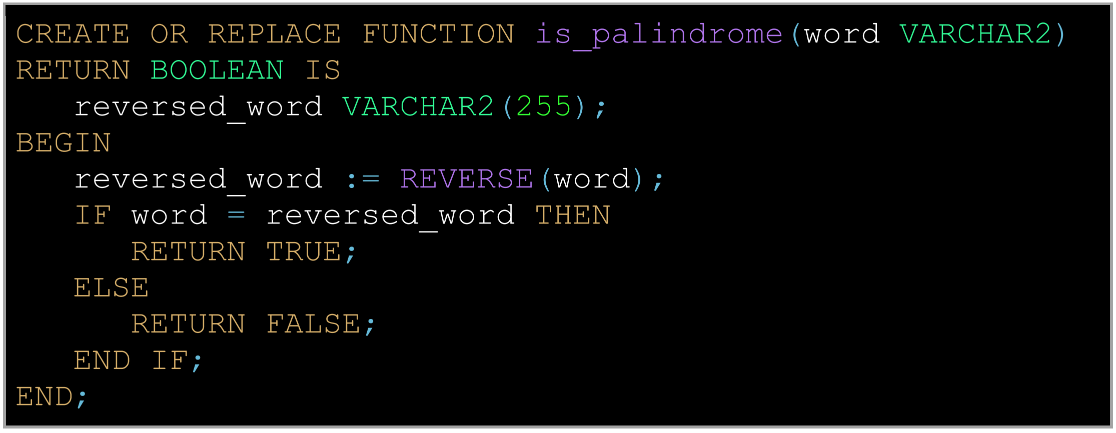
B. It calculates the square root of `base` raised to the power of `exponent`.

C. It calculates the factorial of `exponent`.

D. It calculates `base` raised to the power of `exponent`.

Correct Option: D

Ques14 - Consider the following PL/SQL function: (Difficulty level – Medium)

```plsql



~~~
CREATE OR REPLACE FUNCTION is_palindrome(word VARCHAR2)
RETURN BOOLEAN IS
   reversed_word VARCHAR2(255);
BEGIN
   reversed_word := REVERSE(word);
   IF word = reversed_word THEN
      RETURN TRUE;
   ELSE
      RETURN FALSE;
   END IF;
END;
~~~

```

What does the PL/SQL function `is_palindrome` do?

A. It checks if the input string `word` is a palindrome (reads the same forwards and backwards) and returns `TRUE` if it is, `FALSE` otherwise.

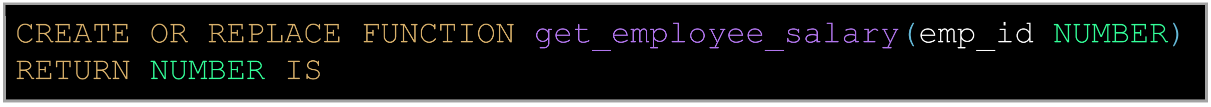
B. It calculates the length of the input string `word`.

C. It calculates the square root of the input number `word`.

D. It checks if the input string `word` contains any digits and returns `TRUE` if it does, `FALSE` otherwise.

Correct Option: A

Ques25 - Consider the following PL/SQL function: (Difficulty level – Medium)

```plsql



~~~
CREATE OR REPLACE FUNCTION get_employee_salary(emp_id NUMBER)
RETURN NUMBER IS
~~~
```

~~~plsql
   salary NUMBER;
BEGIN
   -- Retrieve the salary of the employee with the given emp_id
   SELECT salary INTO salary FROM employees WHERE employee_id = emp_id;
   IF SQL%FOUND THEN
      RETURN salary;
   ELSE
      RETURN NULL;
   END IF;
END;
~~~

```

**What does the PL/SQL function `get_employee_salary` do?**

A. It calculates the average salary of all employees.

B. It retrieves the salary of the employee with the specified `emp_id`.

C. It calculates the total salary of all employees.

D. It retrieves the highest salary among all employees.

**Correct Option:** B

---

**Ques15 - Consider the following PL/SQL function: (Difficulty level – Medium)**

```plsql

~~~plsql
CREATE OR REPLACE FUNCTION count_words(sentence VARCHAR2)
RETURN NUMBER IS
   word_count NUMBER := 0;
BEGIN
   FOR i IN 1..LENGTH(sentence) LOOP
      IF SUBSTR(sentence, i, 1) = ' ' THEN
         word_count := word_count + 1;
      END IF;
   END LOOP;
   -- Add one to count the last word
   word_count := word_count + 1;
   RETURN word_count;
END;
~~~

```

What does the PL/SQL function `count_words` do?

A. It calculates the number of characters in the input sentence.
```

B. It calculates the number of words in the input sentence.

C. It calculates the number of vowels in the input sentence.

D. It calculates the number of digits in the input sentence.

Correct Option: B

---

2 mark questions –

Ques1 - Consider the following PL/SQL package specification: (Difficulty level – Medium)

```plsql
CREATE OR REPLACE PACKAGE product_discounts AS
   FUNCTION calculate_discount(product_id NUMBER, quantity NUMBER) RETURN NUMBER;
   FUNCTION apply_discount_to_order(order_id NUMBER) RETURN BOOLEAN;
END product_discounts;
/
```

What does the PL/SQL package `product_discounts` contain?

A. It contains two PL/SQL functions, `calculate_discount` and `apply_discount_to_order`, for calculating and applying product discounts.

B. It contains two PL/SQL triggers, `calculate_discount` and `apply_discount_to_order`, for calculating and applying product discounts.

C. It contains two PL/SQL procedures, `calculate_discount` and `apply_discount_to_order`, for calculating and applying product discounts.

D. It contains one PL/SQL function, `product_discounts`, and one PL/SQL procedure, `product_discounts`, for calculating and applying product discounts.

Correct Option: A

Ques2 - Consider the following PL/SQL package specification: (Difficulty level – Medium)

```plsql
CREATE OR REPLACE PACKAGE order_processing AS
   FUNCTION process_order(order_id NUMBER) RETURN BOOLEAN;
```

```plsql
    FUNCTION validate_payment(order_id NUMBER) RETURN BOOLEAN;
END order_processing;
/
```

What does the PL/SQL package `order_processing` contain?

A. It contains two PL/SQL functions, `process_order` and `validate_payment`, for processing orders and validating payments.

B. It contains two PL/SQL triggers, `process_order` and `validate_payment`, for processing orders and validating payments.

C. It contains two PL/SQL procedures, `process_order` and `validate_payment`, for processing orders and validating payments.

D. It contains one PL/SQL function, `order_processing`, and one PL/SQL procedure, `order_processing`, for processing orders and validating payments.

Correct Option: A

Ques3 - Consider the following PL/SQL package specification: (Difficulty level – Medium)

```plsql
CREATE OR REPLACE PACKAGE employee_management AS
    FUNCTION hire_employee(name VARCHAR2, salary NUMBER)
RETURN NUMBER;
    FUNCTION terminate_employee(employee_id NUMBER) RETURN
BOOLEAN;
END employee_management;
/
```

What does the PL/SQL package `employee_management` contain?

A. It contains two PL/SQL functions, `hire_employee` and `terminate_employee`, for hiring and terminating employees.

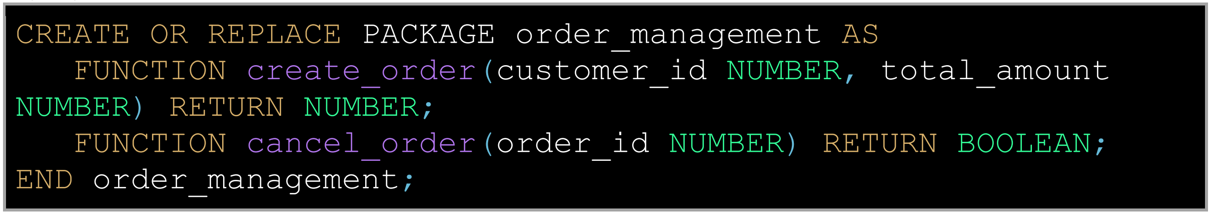
B. It contains two PL/SQL triggers, `hire_employee` and `terminate_employee`, for hiring and terminating employees.

C. It contains two PL/SQL procedures, `hire_employee` and `terminate_employee`, for hiring and terminating employees.

D. It contains one PL/SQL function, `employee_management`, and one PL/SQL procedure, `employee_management`, for hiring and terminating employees.

Correct Option: A

Ques4 - Consider the following PL/SQL package specification: (Difficulty level – Medium)



```plsql
CREATE OR REPLACE PACKAGE order_management AS
   FUNCTION create_order(customer_id NUMBER, total_amount NUMBER) RETURN NUMBER;
   FUNCTION cancel_order(order_id NUMBER) RETURN BOOLEAN;
END order_management;
/
```

What does the PL/SQL package `order_management` contain?

A. It contains two PL/SQL functions, `create_order` and `cancel_order`, for creating and canceling orders.

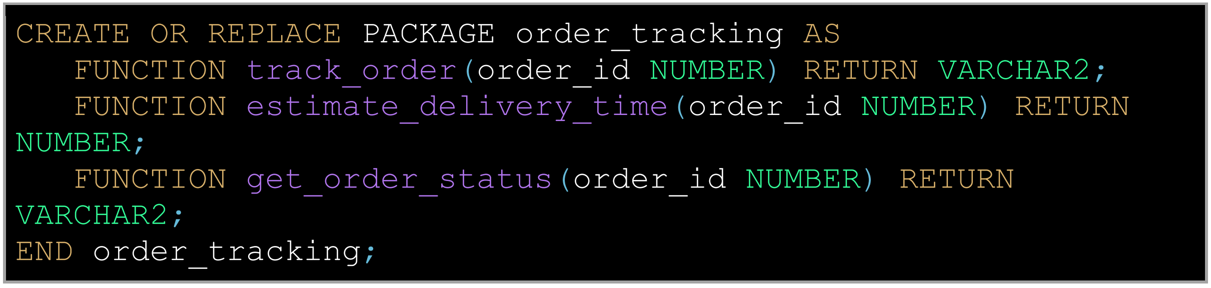
B. It contains two PL/SQL triggers, `create_order` and `cancel_order`, for creating and canceling orders.

C. It contains two PL/SQL procedures, `create_order` and `cancel_order`, for creating and canceling orders.

D. It contains one PL/SQL function, `order_management`, and one PL/SQL procedure, `order_management`, for creating and canceling orders.

Correct Option: A

Ques5 - Consider the following PL/SQL package specification: (Difficulty level – Hard)



```plsql
CREATE OR REPLACE PACKAGE order_tracking AS
   FUNCTION track_order(order_id NUMBER) RETURN VARCHAR2;
   FUNCTION estimate_delivery_time(order_id NUMBER) RETURN NUMBER;
   FUNCTION get_order_status(order_id NUMBER) RETURN VARCHAR2;
END order_tracking;
/
```

What does the PL/SQL package `order_tracking` contain?

A. It contains three PL/SQL functions, `track_order`, `estimate_delivery_time`, and `get_order_status`, for tracking orders and estimating delivery times.

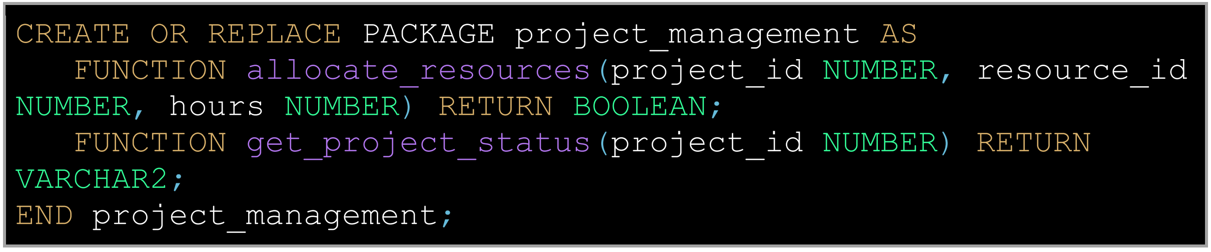
B. It contains three PL/SQL triggers, `track_order`, `estimate_delivery_time`, and `get_order_status`, for tracking orders and estimating delivery times.

C. It contains three PL/SQL procedures, `track_order`, `estimate_delivery_time`, and `get_order_status`, for tracking orders and estimating delivery times.

D. It contains one PL/SQL function, `order_tracking`, and one PL/SQL procedure, `order_tracking`, for tracking orders and estimating delivery times.

Correct Option: A

Ques6 - Consider the following PL/SQL package specification: (Difficulty level – Hard)



```plsql
CREATE OR REPLACE PACKAGE project_management AS
   FUNCTION allocate_resources(project_id NUMBER, resource_id NUMBER, hours NUMBER) RETURN BOOLEAN;
   FUNCTION get_project_status(project_id NUMBER) RETURN VARCHAR2;
END project_management;
/
```

What does the PL/SQL package `project_management` contain?

A. It contains two PL/SQL functions, `allocate_resources` and `get_project_status`, for resource allocation and project status retrieval.

B. It contains two PL/SQL triggers, `allocate_resources` and `get_project_status`, for resource allocation and project status retrieval.

C. It contains two PL/SQL procedures, `allocate_resources` and `get_project_status`, for resource allocation and project status retrieval.

D. It contains one PL/SQL function, `project_management`, and one PL/SQL procedure, `project_management`, for resource allocation and project status retrieval.

Correct Option: A

Ques7 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

```plsql
```

~~~plsql
CREATE OR REPLACE PACKAGE student_grading AS
   FUNCTION calculate_final_grade(student_id NUMBER,
course_id NUMBER) RETURN CHAR;
   FUNCTION get_student_ranking(course_id NUMBER) RETURN
NUMBER;
END student_grading;
~~~

/

```

**What does the PL/SQL package `student_grading` contain?**

A. It contains two PL/SQL functions, `calculate_final_grade` and `get_student_ranking`, for calculating student grades and retrieving student rankings in a course.

B. It contains two PL/SQL triggers, `calculate_final_grade` and `get_student_ranking`, for calculating student grades and retrieving student rankings in a course.

C. It contains two PL/SQL procedures, `calculate_final_grade` and `get_student_ranking`, for calculating student grades and retrieving student rankings in a course.

D. It contains one PL/SQL function, `student_grading`, and one PL/SQL procedure, `student_grading`, for calculating student grades and retrieving student rankings in a course.

**Correct Option:** A

---

**Ques8 - Consider the following PL/SQL package specification: (Difficulty level – Hard)**

```plsql

~~~plsql
CREATE OR REPLACE PACKAGE medical_records AS
   FUNCTION get_patient_history(patient_id NUMBER) RETURN
CLOB;
   FUNCTION analyze_patient_data(patient_id NUMBER) RETURN
CLOB;
END medical_records;
/
~~~

```

What does the PL/SQL package `medical_records` contain?

A. It contains two PL/SQL functions, `get_patient_history` and `analyze_patient_data`, for retrieving patient medical history and analyzing patient data.

B. It contains two PL/SQL triggers, `get_patient_history` and `analyze_patient_data`, for retrieving patient medical history and analyzing patient data.

C. It contains two PL/SQL procedures, `get_patient_history` and `analyze_patient_data`, for retrieving patient medical history and analyzing patient data.
```

D. It contains one PL/SQL function, `medical_records`, and one PL/SQL procedure, `medical_records`, for retrieving patient medical history and analyzing patient data.

Correct Option: A

Ques9 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

```plsql
CREATE OR REPLACE PACKAGE order_tracking AS
    FUNCTION track_order(order_id NUMBER) RETURN VARCHAR2;
    FUNCTION estimate_delivery_time(order_id NUMBER) RETURN
NUMBER;
    FUNCTION get_order_status(order_id NUMBER) RETURN
VARCHAR2;
END order_tracking;
/
```

What does the PL/SQL package `order_tracking` contain?

A. It contains three PL/SQL functions, `track_order`, `estimate_delivery_time`, and `get_order_status`, for tracking orders and estimating delivery times.

B. It contains three PL/SQL triggers, `track_order`, `estimate_delivery_time`, and `get_order_status`, for tracking orders and estimating delivery times.

C. It contains three PL/SQL procedures, `track_order`, `estimate_delivery_time`, and `get_order_status`, for tracking orders and estimating delivery times.

D. It contains one PL/SQL function, `order_tracking`, and one PL/SQL procedure, `order_tracking`, for tracking orders and estimating delivery times.

Correct Option: A

Ques10 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

```plsql
CREATE OR REPLACE PACKAGE project_management AS
    FUNCTION allocate_resources(project_id NUMBER, resource_id
NUMBER, hours NUMBER) RETURN BOOLEAN;
    FUNCTION get_project_status(project_id NUMBER) RETURN
VARCHAR2;
```

~~~~
END project_management;
/
```
~~~~

What does the PL/SQL package `project_management` contain?

A. It contains two PL/SQL functions, `allocate_resources` and `get_project_status`, for resource allocation and project status retrieval.

B. It contains two PL/SQL triggers, `allocate_resources` and `get_project_status`, for resource allocation and project status retrieval.

C. It contains two PL/SQL procedures, `allocate_resources` and `get_project_status`, for resource allocation and project status retrieval.

D. It contains one PL/SQL function, `project_management`, and one PL/SQL procedure, `project_management`, for resource allocation and project status retrieval.

Correct Option: A

Ques11 - Consider the following PL/SQL package specification: (Difficulty level – Hard)

~~~~
```plsql
CREATE OR REPLACE PACKAGE student_grading AS
   FUNCTION calculate_final_grade(student_id NUMBER,
course_id NUMBER) RETURN CHAR;
   FUNCTION get_student_ranking(course_id NUMBER) RETURN
NUMBER;
END student_grading;
/
```
~~~~

What does the PL/SQL package `student_grading` contain?

A. It contains two PL/SQL functions, `calculate_final_grade` and `get_student_ranking`, for calculating student grades and retrieving student rankings in a course.

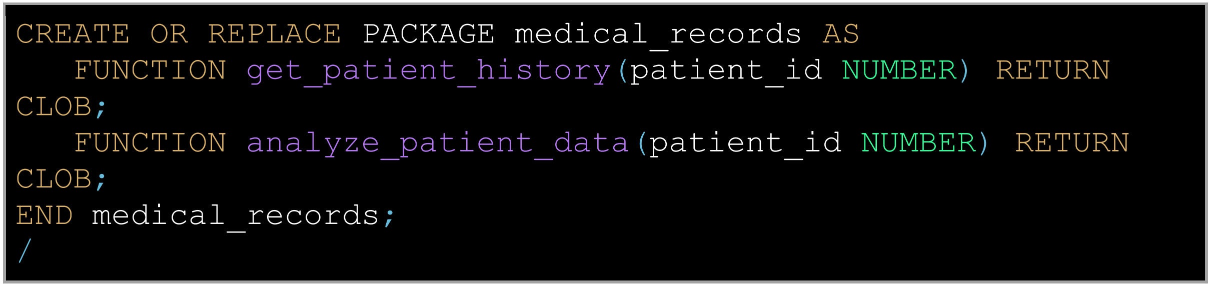
B. It contains two PL/SQL triggers, `calculate_final_grade` and `get_student_ranking`, for calculating student grades and retrieving student rankings in a course.

C. It contains two PL/SQL procedures, `calculate_final_grade` and `get_student_ranking`, for calculating student grades and retrieving student rankings in a course.

D. It contains one PL/SQL function, `student_grading`, and one PL/SQL procedure, `student_grading`, for calculating student grades and retrieving student rankings in a course.

Correct Option: A

Ques12 - Consider the following PL/SQL package specification: (Difficulty level – Hard)



```plsql
CREATE OR REPLACE PACKAGE medical_records AS
   FUNCTION get_patient_history(patient_id NUMBER) RETURN
CLOB;
   FUNCTION analyze_patient_data(patient_id NUMBER) RETURN
CLOB;
END medical_records;
/
```

What does the PL/SQL package `medical_records` contain?

A. It contains two PL/SQL functions, `get_patient_history` and `analyze_patient_data`, for retrieving patient medical history and analyzing patient data.

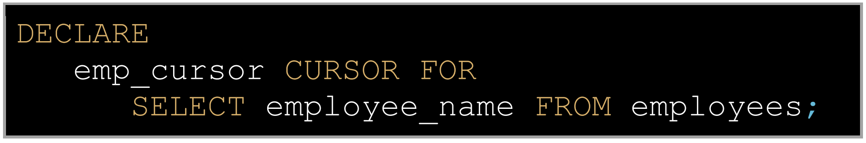
B. It contains two PL/SQL triggers, `get_patient_history` and `analyze_patient_data`, for retrieving patient medical history and analyzing patient data.

C. It contains two PL/SQL procedures, `get_patient_history` and `analyze_patient_data`, for retrieving patient medical history and analyzing patient data.

D. It contains one PL/SQL function, `medical_records`, and one PL/SQL procedure, `medical_records`, for retrieving patient medical history and analyzing patient data.

Correct Option: A

Ques13 - Consider the following SQL cursor declaration: (Difficulty level – Easy)



```sql
DECLARE
   emp_cursor CURSOR FOR
      SELECT employee_name FROM employees;
```

What does the SQL cursor `emp_cursor` do?

A. It retrieves all columns from the `employees` table.

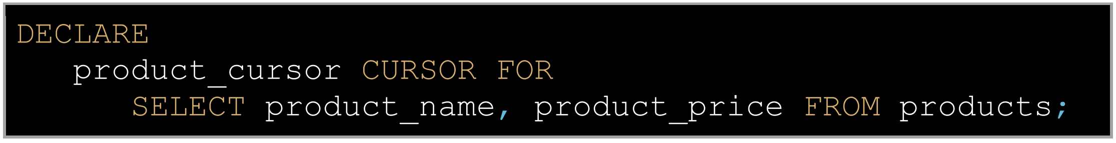
B. It retrieves the `employee_name` column from the `employees` table.

C. It updates the `employee_name` column in the `employees` table.

D. It deletes records from the `employees` table.

Correct Option: B

Ques14 - Consider the following SQL cursor declaration: (Difficulty level – Easy)



```sql
DECLARE
   product_cursor CURSOR FOR
      SELECT product_name, product_price FROM products;
```

What does the SQL cursor `product_cursor` do?

A. It retrieves all columns from the `products` table.

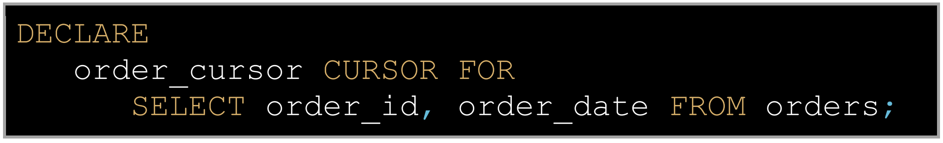
B. It retrieves the `product_name` and `product_price` columns from the `products` table.

C. It updates the `product_name` and `product_price` columns in the `products` table.

D. It deletes records from the `products` table.

Correct Option: B

Ques15 - Consider the following SQL cursor declaration: (Difficulty level – Easy)



```sql
DECLARE
   order_cursor CURSOR FOR
      SELECT order_id, order_date FROM orders;
```

What does the SQL cursor `order_cursor` do?

A. It retrieves all columns from the `orders` table.

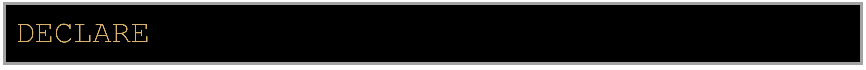
B. It retrieves the `order_id` and `order_date` columns from the `orders` table.

C. It updates the `order_id` and `order_date` columns in the `orders` table.

D. It deletes records from the `orders` table.

Correct Option: B

Ques1 - Consider the following SQL cursor declaration: (Difficulty level – Easy)



```sql
DECLARE
```

```
    customer_cursor CURSOR FOR
        SELECT customer_name FROM customers;
```

What does the SQL cursor `customer_cursor` do?

A. It retrieves all columns from the `customers` table.

B. It retrieves the `customer_name` column from the `customers` table.

C. It updates the `customer_name` column in the `customers` table.

D. It deletes records from the `customers` table.

Correct Option: B

Ques2 - Consider the following SQL cursor declaration: (Difficulty level – Easy)

```sql
DECLARE
    employee_cursor CURSOR FOR
        SELECT employee_id, employee_name FROM employees;
```

What does the SQL cursor `employee_cursor` do?

A. It retrieves all columns from the `employees` table.

B. It retrieves the `employee_id` and `employee_name` columns from the `employees` table.

C. It updates the `employee_id` and `employee_name` columns in the `employees` table.

D. It deletes records from the `employees` table.

Correct Option: B

Ques3 - Consider the following SQL cursor declaration: (Difficulty level – Easy)

```sql
DECLARE
    product_cursor CURSOR FOR
        SELECT product_id FROM products WHERE product_price >
100;
```

What does the SQL cursor `product_cursor` do?

A. It retrieves all columns from the `products` table.

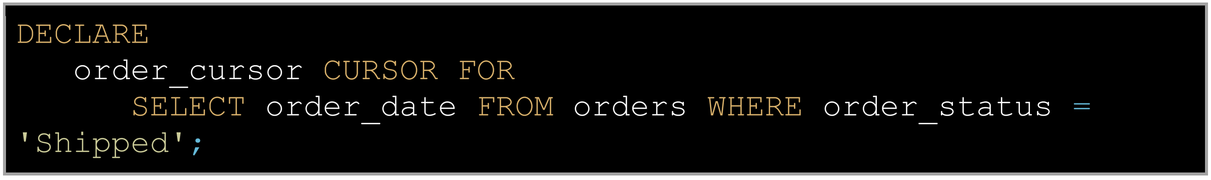
B. It retrieves the `product_id` column from the `products` table for products with a price greater than 100.

C. It updates the `product_id` column in the `products` table.

D. It deletes records from the `products` table.

Correct Option: B

Ques4 - Consider the following SQL cursor declaration: (Difficulty level – Easy)



```sql
DECLARE
   order_cursor CURSOR FOR
      SELECT order_date FROM orders WHERE order_status =
'Shipped';
```

What does the SQL cursor `order_cursor` do?

A. It retrieves all columns from the `orders` table.

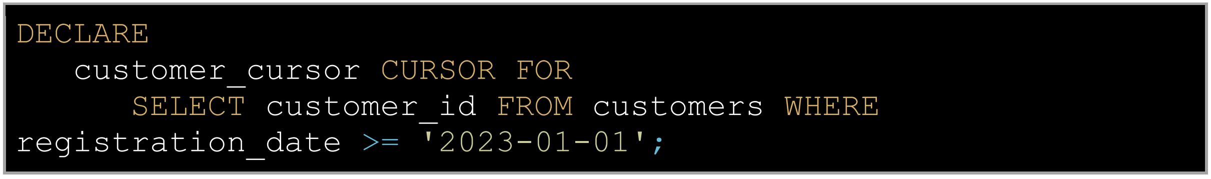
B. It retrieves the `order_date` column from the `orders` table for orders with a status of 'Shipped'.

C. It updates the `order_date` column in the `orders` table.

D. It deletes records from the `orders` table.

Correct Option: B

Ques5 - Consider the following SQL cursor declaration: (Difficulty level – Easy)



```sql
DECLARE
   customer_cursor CURSOR FOR
      SELECT customer_id FROM customers WHERE
registration_date >= '2023-01-01';
```

What does the SQL cursor `customer_cursor` do?

A. It retrieves all columns from the `customers` table.

B. It retrieves the `customer_id` column from the `customers` table for customers registered on or after January 1, 2023.

C. It updates the `customer_id` column in the `customers` table.

D. It deletes records from the `customers` table.

Correct Option: B

Ques6 - Consider the following SQL cursor declaration: (Difficulty level – Easy)

```sql
DECLARE
   employee_cursor CURSOR FOR
      SELECT department_id, COUNT(*) FROM employees GROUP BY
department_id;
```

What does the SQL cursor `employee_cursor` do?

A. It retrieves all columns from the `employees` table.

B. It retrieves the `department_id` and the count of employees in each department from the `employees` table.

C. It updates the `department_id` and employee counts in the `employees` table.

D. It deletes records from the `employees` table.

Correct Option: B

Ques7 - Consider the following SQL cursor declaration: (Difficulty level – Easy)

```sql
DECLARE
   product_cursor CURSOR FOR
      SELECT product_name, product_category FROM products
WHERE product_category = 'Electronics';
```

What does the SQL cursor `product_cursor` do?

A. It retrieves all columns from the `products` table.

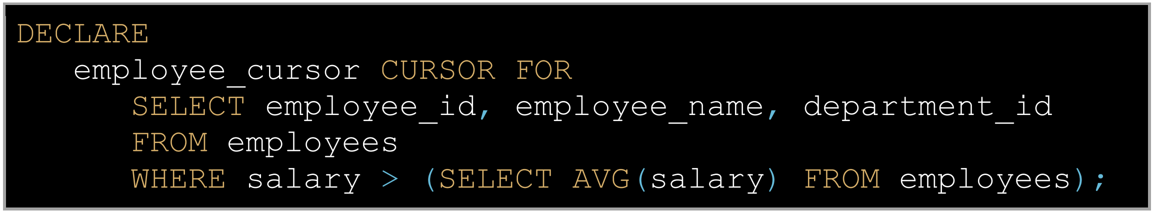
B. It retrieves the `product_name` and `product_category` columns from the `products` table for products in the 'Electronics' category.

C. It updates the `product_name` and `product_category` columns in the `products` table.

D. It deletes records from the `products` table.

Correct Option: B

Ques8 - Consider the following SQL cursor declaration: (Difficulty level – Hard)



```sql
DECLARE
   employee_cursor CURSOR FOR
      SELECT employee_id, employee_name, department_id
      FROM employees
      WHERE salary > (SELECT AVG(salary) FROM employees);
```

What does the SQL cursor `employee_cursor` do?

A. It retrieves all columns from the `employees` table.

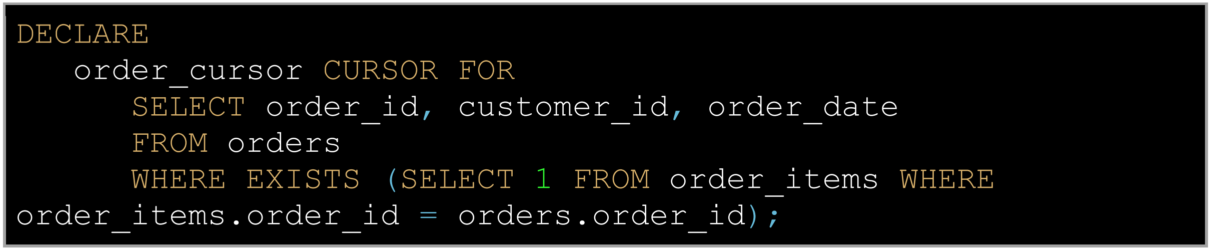
B. It retrieves the `employee_id`, `employee_name`, and `department_id` columns from the `employees` table for employees with salaries above the average salary in the company.

C. It updates the `employee_id`, `employee_name`, and `department_id` columns in the `employees` table.

D. It deletes records from the `employees` table.

Correct Option: B

Ques9 - Consider the following SQL cursor declaration: (Difficulty level – Hard)



```sql
DECLARE
   order_cursor CURSOR FOR
      SELECT order_id, customer_id, order_date
      FROM orders
      WHERE EXISTS (SELECT 1 FROM order_items WHERE
order_items.order_id = orders.order_id);
```

What does the SQL cursor `order_cursor` do?

A. It retrieves all columns from the `orders` table.

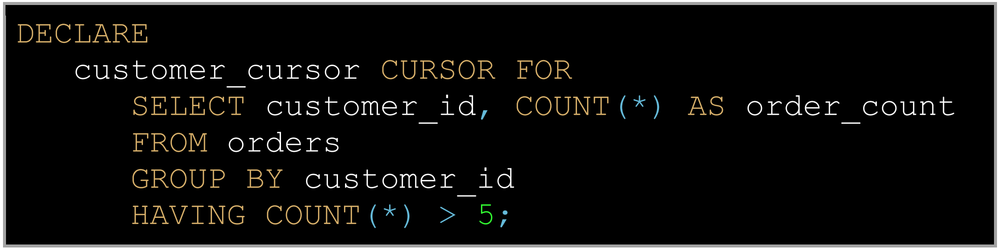
B. It retrieves the `order_id`, `customer_id`, and `order_date` columns from the `orders` table for orders that have associated order items.

C. It updates the `order_id`, `customer_id`, and `order_date` columns in the `orders` table.

D. It deletes records from the `orders` table.

Correct Option: B

Ques10 - Consider the following SQL cursor declaration: (Difficulty level – Hard)



```sql
DECLARE
   customer_cursor CURSOR FOR
      SELECT customer_id, COUNT(*) AS order_count
      FROM orders
      GROUP BY customer_id
      HAVING COUNT(*) > 5;
```

What does the SQL cursor `customer_cursor` do?

A. It retrieves all columns from the `orders` table.

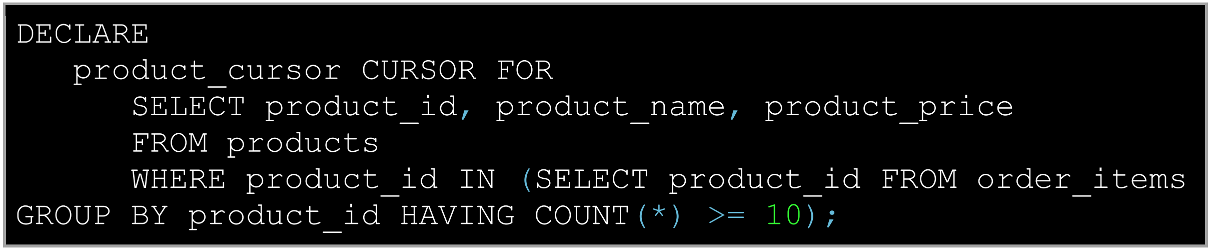
B. It retrieves the `customer_id` and the count of orders placed by each customer from the `orders` table for customers who have placed more than 5 orders.

C. It updates the `customer_id` and order counts in the `orders` table.

D. It deletes records from the `orders` table.

Correct Option: B

Ques11 - Consider the following SQL cursor declaration: (Difficulty level – Hard)



```sql
DECLARE
   product_cursor CURSOR FOR
      SELECT product_id, product_name, product_price
      FROM products
      WHERE product_id IN (SELECT product_id FROM order_items
GROUP BY product_id HAVING COUNT(*) >= 10);
```

What does the SQL cursor `product_cursor` do?

A. It retrieves all columns from the `products` table.

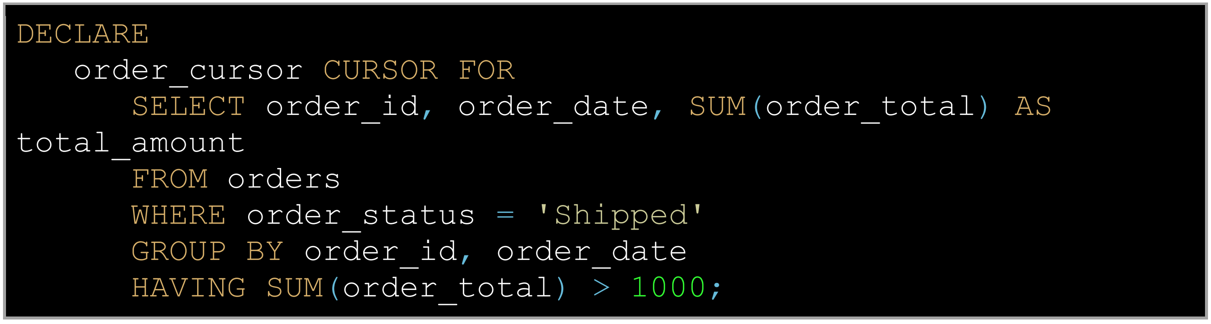
B. It retrieves the `product_id`, `product_name`, and `product_price` columns from the `products` table for products that have been ordered at least 10 times.

C. It updates the `product_id`, `product_name`, and `product_price` columns in the `products` table.

D. It deletes records from the `products` table.

Correct Option: B

Ques12 - Consider the following SQL cursor declaration: (Difficulty level – Hard)



```sql
DECLARE
   order_cursor CURSOR FOR
      SELECT order_id, order_date, SUM(order_total) AS
total_amount
      FROM orders
      WHERE order_status = 'Shipped'
      GROUP BY order_id, order_date
      HAVING SUM(order_total) > 1000;
```

What does the SQL cursor `order_cursor` do?

A. It retrieves all columns from the `orders` table.

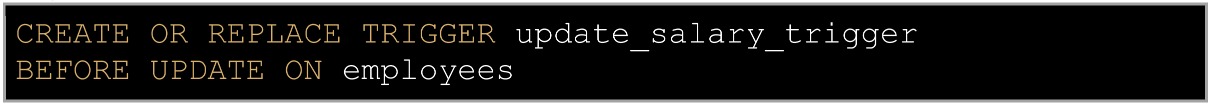
B. It retrieves the `order_id`, `order_date`, and total order amount columns from the `orders` table for shipped orders with a total amount greater than 1000.

C. It updates the `order_id`, `order_date`, and total order amount columns in the `orders` table.

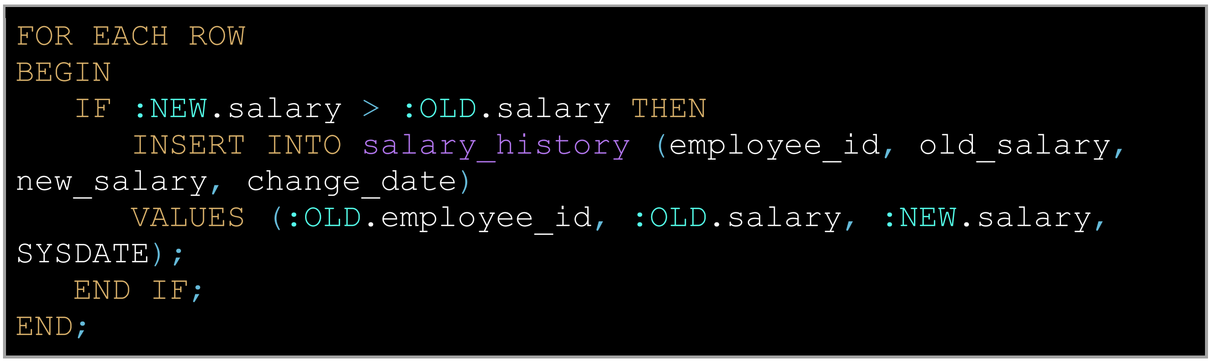
D. It deletes records from the `orders` table.

Correct Option: B

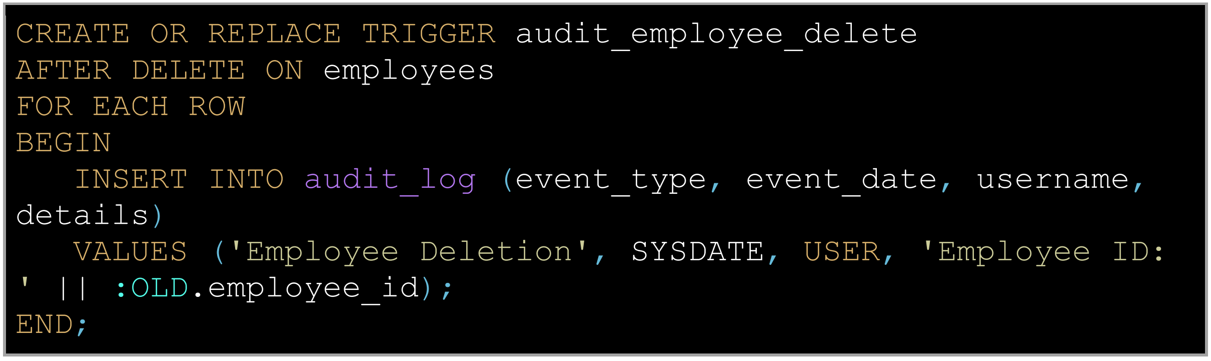
Ques13 - Consider the following SQL trigger: (Difficulty level – Hard)



```sql
CREATE OR REPLACE TRIGGER update_salary_trigger
BEFORE UPDATE ON employees
```



```sql
FOR EACH ROW
BEGIN
   IF :NEW.salary > :OLD.salary THEN
      INSERT INTO salary_history (employee_id, old_salary,
new_salary, change_date)
      VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary,
SYSDATE);
   END IF;
END;
```

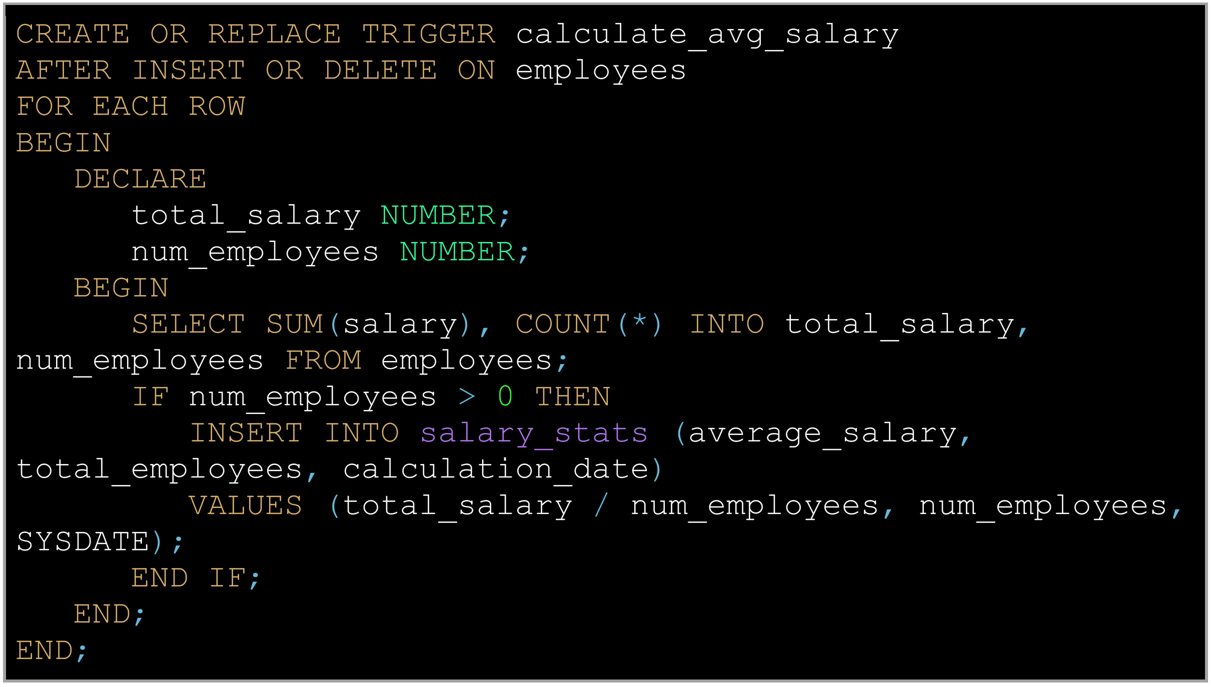
/
~~~

What does the SQL trigger `update_salary_trigger` do?

A. It updates the salary of all employees in the `employees` table.

B. It inserts a record into the `salary_history` table whenever an employee's salary is increased.

C. It deletes records from the `employees` table whenever an employee's salary is updated.

D. It calculates the average salary of all employees.

Correct Option: B

Ques14 - Consider the following SQL trigger: (Difficulty level – Hard)

```sql



```sql
CREATE OR REPLACE TRIGGER audit_employee_delete
AFTER DELETE ON employees
FOR EACH ROW
BEGIN
   INSERT INTO audit_log (event_type, event_date, username,
details)
   VALUES ('Employee Deletion', SYSDATE, USER, 'Employee ID:
' || :OLD.employee_id);
END;
~~~

/
```

What does the SQL trigger `audit_employee_delete` do?

A. It updates employee records in the `employees` table.

B. It inserts a record into the `audit_log` table whenever an employee is deleted.

C. It inserts a record into the `employees` table whenever an employee is deleted.

D. It calculates the total number of employees in the `employees` table.
```

**Correct Option:** B

---

**Ques15 - Consider the following SQL trigger: (Difficulty level – Hard)**



```sql
CREATE OR REPLACE TRIGGER calculate_avg_salary
AFTER INSERT OR DELETE ON employees
FOR EACH ROW
BEGIN
   DECLARE
      total_salary NUMBER;
      num_employees NUMBER;
   BEGIN
      SELECT SUM(salary), COUNT(*) INTO total_salary,
num_employees FROM employees;
      IF num_employees > 0 THEN
         INSERT INTO salary_stats (average_salary,
total_employees, calculation_date)
         VALUES (total_salary / num_employees, num_employees,
SYSDATE);
      END IF;
   END;
END;
/
```

What does the SQL trigger `calculate_avg_salary` do?

A. It updates the salary of all employees in the `employees` table.

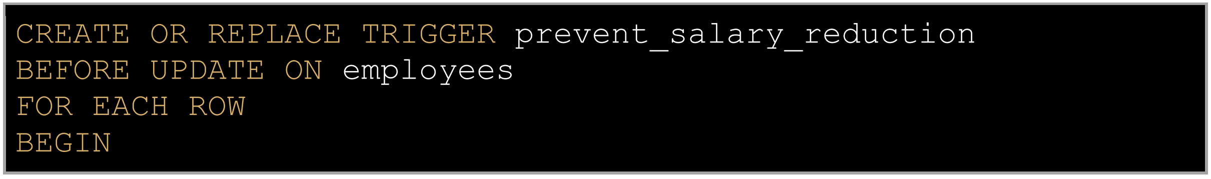
B. It calculates the average salary and total number of employees whenever a new employee is inserted or an employee is deleted.

C. It inserts a record into the `salary_stats` table whenever an employee is deleted.

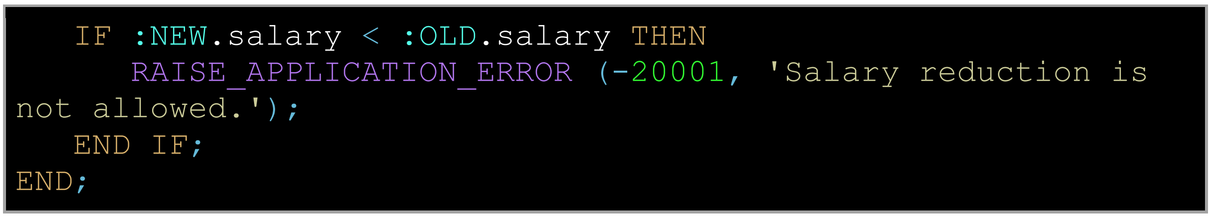
D. It calculates the total number of employees in the `employees` table.

Correct Option: B

Ques1 - Consider the following SQL trigger: (Difficulty level – Hard)



```sql
CREATE OR REPLACE TRIGGER prevent_salary_reduction
BEFORE UPDATE ON employees
FOR EACH ROW
BEGIN
```



~~~~
    IF :NEW.salary < :OLD.salary THEN
        RAISE_APPLICATION_ERROR (-20001, 'Salary reduction is not allowed.');
    END IF;
END;
~~~~

/
```

What does the SQL trigger `prevent_salary_reduction` do?

A. It updates the salary of all employees in the `employees` table.

B. It prevents any attempt to reduce an employee's salary and raises a custom application error if such an update is detected.

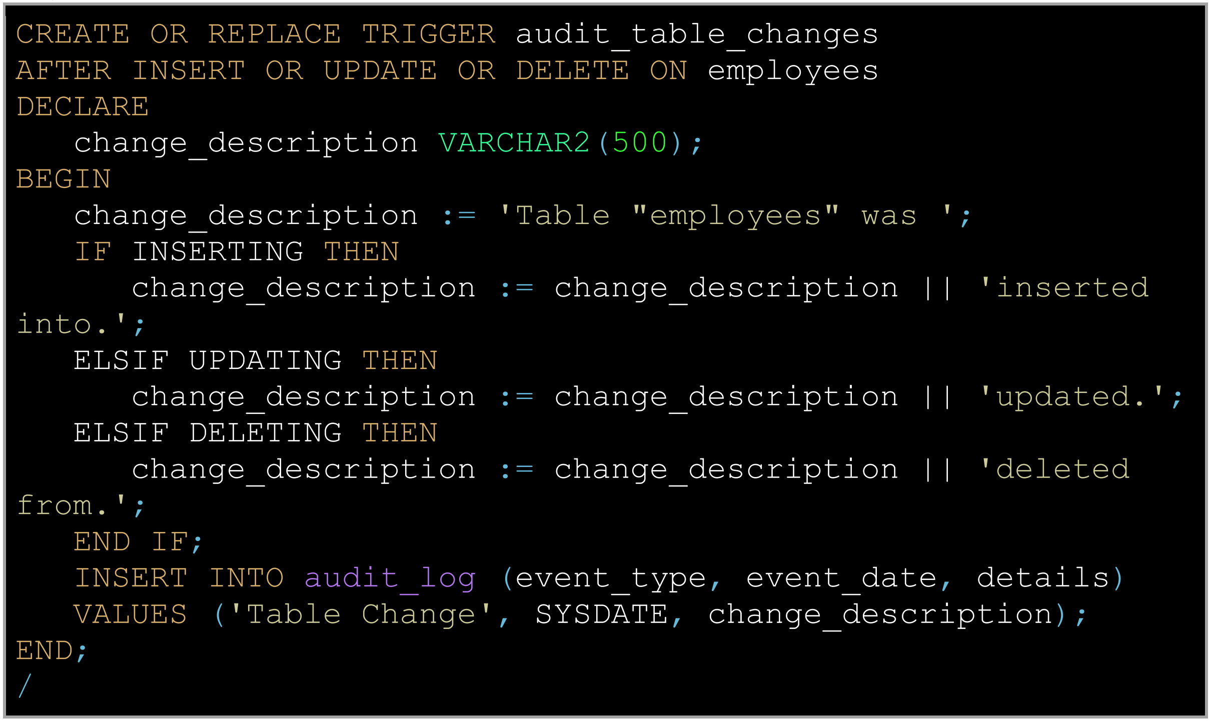
C. It inserts a record into the `salary_history` table whenever an employee's salary is increased.

D. It calculates the average salary of all employees.

Correct Option: B

Ques2 - Consider the following SQL trigger: (Difficulty level – Medium)

```sql



~~~~
CREATE OR REPLACE TRIGGER audit_table_changes
AFTER INSERT OR UPDATE OR DELETE ON employees
DECLARE
    change_description VARCHAR2(500);
BEGIN
    change_description := 'Table "employees" was ';
    IF INSERTING THEN
        change_description := change_description || 'inserted into.';
    ELSIF UPDATING THEN
        change_description := change_description || 'updated.';
    ELSIF DELETING THEN
        change_description := change_description || 'deleted from.';
    END IF;
    INSERT INTO audit_log (event_type, event_date, details)
    VALUES ('Table Change', SYSDATE, change_description);
END;
/
~~~~

```

What does the SQL trigger `audit_table_changes` do?

A. It updates the `employees` table whenever a change is made to it.
```

B. It inserts a record into the `audit_log` table whenever a change (insert, update, or delete) is made to the `employees` table, including a description of the change.

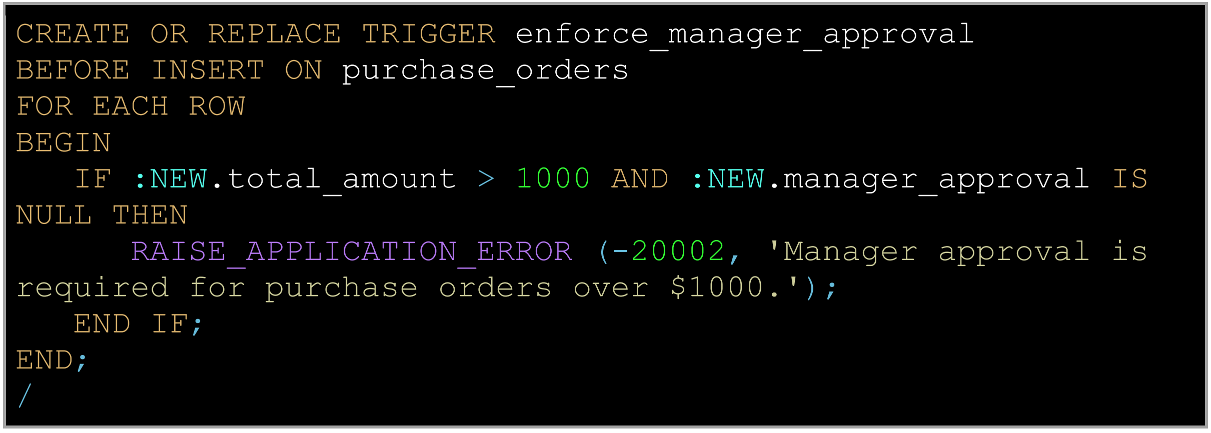
C. It calculates the total number of employees in the `employees` table.

D. It deletes records from the `employees` table whenever a change is made to it.

**Correct Option:** B

---

**Ques3 - Consider the following SQL trigger: (Difficulty level – Easy)**



```sql
CREATE OR REPLACE TRIGGER enforce_manager_approval
BEFORE INSERT ON purchase_orders
FOR EACH ROW
BEGIN
   IF :NEW.total_amount > 1000 AND :NEW.manager_approval IS
NULL THEN
      RAISE_APPLICATION_ERROR (-20002, 'Manager approval is
required for purchase orders over $1000.');
   END IF;
END;
/
```

What does the SQL trigger `enforce_manager_approval` do?

A. It inserts records into the `purchase_orders` table.

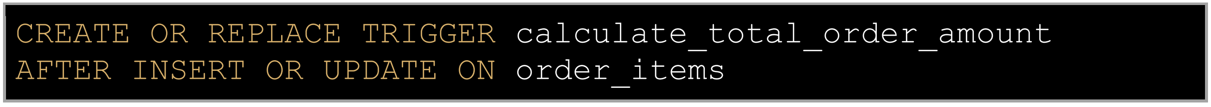
B. It updates records in the `purchase_orders` table.

C. It prevents the insertion of purchase orders with a total amount over $1000 if they don't have manager approval, raising a custom application error if such an insert is attempted.

D. It calculates the total amount of all purchase orders.

Correct Option: C

Ques4 - Consider the following SQL trigger: (Difficulty level – Hard)



```sql
CREATE OR REPLACE TRIGGER calculate_total_order_amount
AFTER INSERT OR UPDATE ON order_items
```

```
FOR EACH ROW
DECLARE
    total_amount NUMBER;
BEGIN
    total_amount := 0;
    SELECT SUM(quantity * unit_price) INTO total_amount FROM
order_items WHERE order_id = :NEW.order_id;
    UPDATE orders SET total_amount = total_amount WHERE
order_id = :NEW.order_id;
END;
/
```

What does the SQL trigger `calculate_total_order_amount` do?

A. It inserts records into the `order_items` table.

B. It updates records in the `order_items` table.

C. It calculates the total order amount for an order whenever a new order item is inserted or an existing order item is updated, and updates the `total_amount` in the `orders` table.

D. It calculates the average order amount.

Correct Option: C

Ques5 - Consider the following SQL trigger: (Difficulty level –Easy)

```sql
CREATE OR REPLACE TRIGGER prevent_duplicate_records
BEFORE INSERT ON employees
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM employees WHERE employee_id =
:NEW.employee_id) THEN
        RAISE_APPLICATION_ERROR (-20003, 'Employee ID must be
unique.');
    END IF;
END;
/
```

What does the SQL trigger `prevent_duplicate_records` do?

A. It inserts records into the `employees` table.

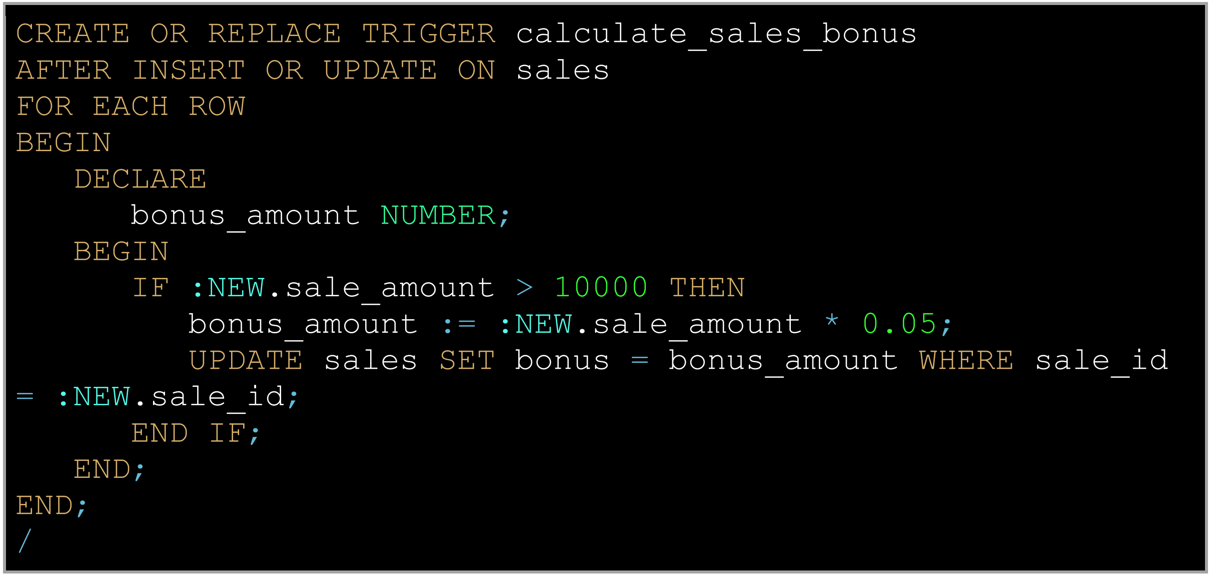
B. It updates records in the `employees` table.

C. It prevents the insertion of duplicate employee records with the same `employee_id`, raising a custom application error if such an insert is attempted.

D. It calculates the total number of employees in the `employees` table.

Correct Option: C

Ques6 - Consider the following SQL trigger: (Difficulty level – Hard)



```sql
CREATE OR REPLACE TRIGGER calculate_sales_bonus
AFTER INSERT OR UPDATE ON sales
FOR EACH ROW
BEGIN
   DECLARE
      bonus_amount NUMBER;
   BEGIN
      IF :NEW.sale_amount > 10000 THEN
         bonus_amount := :NEW.sale_amount * 0.05;
         UPDATE sales SET bonus = bonus_amount WHERE sale_id
= :NEW.sale_id;
      END IF;
   END;
END;
/
```

What does the SQL trigger `calculate_sales_bonus` do?

A. It inserts records into the `sales` table.

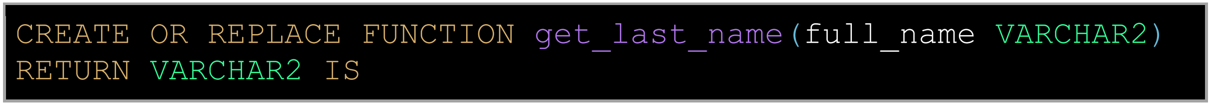
B. It updates records in the `sales` table.

C. It calculates a sales bonus for sales with an amount over $10,000 and updates the `bonus` field in the `sales` table whenever a new sale is inserted or an existing sale is updated.

D. It calculates the average sale amount.

Correct Option: C

Ques7 - Consider the following PL/SQL function: (Difficulty level – Easy)



```plsql
CREATE OR REPLACE FUNCTION get_last_name(full_name VARCHAR2)
RETURN VARCHAR2 IS
```

```
    last_name VARCHAR2(50);
BEGIN
    last_name := SUBSTR(full_name, INSTR(full_name, ' ')+1);
    RETURN last_name;
END;
```

What does the PL/SQL function `get_last_name` do?

A. It calculates the average length of all words in the input `full_name`.

B. It calculates the length of the last word in the input `full_name`.

C. It retrieves the last name from the input `full_name`.

D. It checks if the input `full_name` contains any digits and returns `TRUE` if it does, `FALSE` otherwise.

Correct Option: C

Ques8 - Consider the following PL/SQL function: (Difficulty level – Easy)

```plsql
CREATE OR REPLACE FUNCTION square_number(num NUMBER)
RETURN NUMBER IS
    square NUMBER;
BEGIN
    square := num * num;
    RETURN square;
END;
```

What does the PL/SQL function `square_number` do?

A. It calculates the square root of the input number `num`.

B. It calculates the sum of two numbers.

C. It calculates the square of the input number `num`.

D. It calculates the factorial of the input number `num`.

Correct Option: C

Ques9 - Consider the following PL/SQL function: (Difficulty level – Easy)

```plsql
CREATE OR REPLACE FUNCTION is_prime(number NUMBER)
RETURN BOOLEAN IS
BEGIN
   IF number <= 1 THEN
      RETURN FALSE;
   END IF;
   FOR i IN 2..number-1 LOOP
      IF MOD(number, i) = 0 THEN
         RETURN FALSE;
      END IF;
   END LOOP;
   RETURN TRUE;
END;
```

What does the PL/SQL function `is_prime` do?

A. It checks if the input `number` is a prime number and returns `TRUE` if it is, `FALSE` otherwise.

B. It calculates the square root of the input `number`.

C. It calculates the factorial of the input `number`.

D. It checks if the input `number` is even and returns `TRUE` if it is, `FALSE` otherwise.

Correct Option: A

**Ques10 - Consider the following PL/

SQL function: (Difficulty level – Easy)**

```plsql
CREATE OR REPLACE FUNCTION get_day_of_week(date_value DATE)
RETURN VARCHAR2 IS
   day_of_week VARCHAR2(15);
BEGIN
   SELECT TO_CHAR(date_value, 'Day') INTO day_of_week FROM
DUAL;
   RETURN day_of_week;
END;
```

What does the PL/SQL function `get_day_of_week` do?

A. It calculates the day of the week for the input `date_value` and returns it as a string.

B. It calculates the square root of the input `date_value`.

C. It calculates the average of multiple dates.

D. It retrieves the month of the input `date_value`.

Correct Option: A

Ques11 - Consider the following SQL cursor declaration: (Difficulty level – Hard)

```sql
DECLARE
   product_cursor CURSOR FOR
      SELECT product_id, product_name
      FROM products
      WHERE product_id NOT IN (SELECT DISTINCT product_id FROM order_items);
```

What does the SQL cursor `product_cursor` do?

A. It retrieves all columns from the `products` table.

B. It retrieves the `product_id` and `product_name` columns from the `products` table for products that have not been ordered.

C. It updates the `product_id` and `product_name` columns in the `products` table.

D. It deletes records from the `products` table.

Correct Option: B

Ques12 - Consider the following SQL cursor declaration: (Difficulty level – Hard)

```sql
DECLARE
   customer_cursor CURSOR FOR
      SELECT customer_id, MAX(order_date) AS last_order_date
      FROM orders
      GROUP BY customer_id
      HAVING MAX(order_date) < TO_DATE('2023-01-01', 'YYYY-MM-DD');
```

What does the SQL cursor `customer_cursor` do?

A. It retrieves all columns from the `orders` table.

B. It retrieves the `customer_id` and the last order date columns from the `orders` table for customers whose last order date is before January 1, 2023.

C. It updates the `customer_id` and last order date columns in the `orders` table.

D. It deletes records from the `orders` table.

Correct Option: B

Ques13 - Consider the following SQL cursor declaration: (Difficulty level – Hard)

```sql
DECLARE
   employee_cursor CURSOR FOR
      SELECT employee_id, employee_name, department_id
      FROM employees
      WHERE department_id = (SELECT department_id FROM
departments WHERE department_name = 'Engineering');
```

What does the SQL cursor `employee_cursor` do?

A. It retrieves all columns from the `employees` table.

B. It retrieves the `employee_id`, `employee_name`, and `department_id` columns from the `employees` table for employees in the 'Engineering' department.

C. It updates the `employee_id`, `employee_name`, and `department_id` columns in the `employees` table.

D. It deletes records from the `employees` table.

Correct Option: B

Ques14 - Consider the following SQL cursor declaration: (Difficulty level – Hard)

```sql
DECLARE
 product_cursor CURSOR FOR
      SELECT product_id, product_name
      FROM products
      WHERE product_price = (SELECT MAX(product_price) FROM
products);
```

What does the SQL cursor `product_cursor` do?

A. It retrieves all columns from the `products` table.

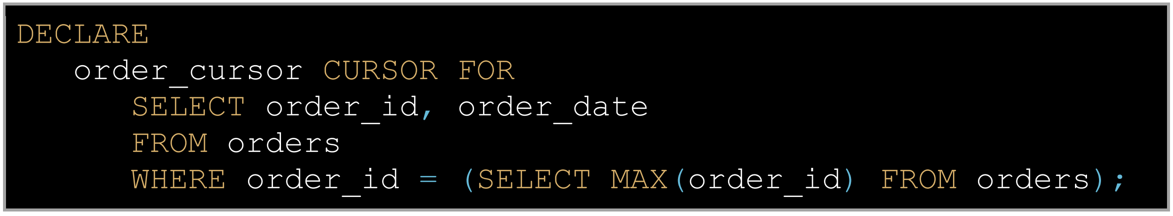
B. It retrieves the `product_id` and `product_name` columns from the `products` table for products with the highest product price.

C. It updates the `product_id` and `product_name` columns in the `products` table.

D. It deletes records from the `products` table.

Correct Option: B

Ques15 - Consider the following SQL cursor declaration: (Difficulty level – Hard)



```sql
DECLARE
   order_cursor CURSOR FOR
      SELECT order_id, order_date
      FROM orders
      WHERE order_id = (SELECT MAX(order_id) FROM orders);
```

What does the SQL cursor `order_cursor` do?

A. It retrieves all columns from the `orders` table.

B. It retrieves the `order_id` and `order_date` columns from the `orders` table for the order with the highest order ID.

C. It updates the `order_id` and `order_date` columns in the `orders` table.

D. It deletes records from the `orders` table.

Correct Option: B

SR	Questions	Option 1	Option 2	Option 3	Option 4	Correct Answer
1	Which of the following is NOT a benefit of using transactions?	Data integrity	High availability	Data consistency	Data durablity	B
2	A transaction that violates the consistency property is considered to be:	Serializable	Inconsistent	Isolate	Error	B
3	Can you change the parameter values of a cursor after it has been declared and opened?	Yes, parameter values can be modified at any time.	No, parameter values are fixed once the cursor is declared and opened.	Parameter values can only be changed during cursor declaration.	Cursors cannot have parameter values.	B
4	Can you declare a cursor without specifying the SELECT statement immediately?	No, a SELECT statement must always be specified.	Yes, a SELECT statement can be added later in the code.	Cursors cannot be declared in PL/SQL.	Cursors are automatically generated in PL/SQL.	A
5	Can you declare multiple cursors with the same name but different parameters in the same PL/SQL block?	Yes, as long as the cursor names are unique.	No, cursor names must be unique regardless of the parameters.	Multiple cursors are not allowed in the same PL/SQL block.	Cursors with parameters cannot have the same name.	B
6	Can you declare multiple cursors within the same PL/SQL block? If so, how do you differentiate them?	No, only one cursor is allowed per block.	Yes, multiple cursors can be declared, and they are differentiated by their data types.	Yes, multiple cursors can be declared, and they are differentiated by their names.	Multiple cursors cannot be used in PL/SQL.	C
7	Can you fetch data from a cursor into individual variables or into a record type? Explain.	Data can only be fetched into individual variables.	Data can only be fetched into a record type.	Data can be fetched into both individual variables and a record type.	Data cannot be fetched from a cursor.	C
8	Can you nest a Cursor FOR Loop inside another Cursor FOR Loop? If so, why might you do so?	No, nesting Cursor FOR Loops is not allowed.	Yes, you can nest Cursor FOR Loops to perform complex data processing and handle related data hierarchies.	Cursor FOR Loops can only be used individually, not nested.	Nesting Cursor FOR Loops results in performance issues.	B
9	Can you use a Cursor FOR Loop to update or delete records in a database table? Explain.	No, Cursor FOR Loops are read-only.	Yes, Cursor FOR Loops can update or delete records using the UPDATE and DELETE statements.	Cursor FOR Loops can only insert records, not update or delete them.	Cursor FOR Loops can only be used for reporting purposes.	B
10	Describe the differences between an implicit cursor and an explicit cursor in PL/SQL.	Implicit cursors are used for data modeling, while explicit cursors are used for data manipulation.	Implicit cursors are automatically created for DML statements, while explicit cursors are user-defined.	Implicit cursors are used for database connections, while explicit cursors are used for loop control.	Implicit cursors are used for hardware design, while explicit cursors are used for web development.	B
11	Describe the purpose of PL/SQL collections, and provide examples of their types.	PL/SQL collections are used for defining variables.	PL/SQL collections are used for database connections.	PL/SQL collections are used for storing multiple values of the same data type.	PL/SQL collections are used for creating triggers.	C
12	Explain how cursor parameters can be used to create dynamic cursors.	Cursor parameters have no role in creating dynamic cursors.	By allowing parameterization of the WHERE clause in the cursor's SELECT statement, you can create dynamic cursors that retrieve specific data based on different criteria.	Cursor parameters can only be used with static cursors.	Cursor parameters can be used to create triggers.	B

13	Explain the concept of triggers in a database context. How are they used in PL/SQL?	Triggers are used for creating web applications.	Triggers are used for hardware design.	Triggers are used for automatically executing PL/SQL code in response to database events.	Triggers are used for data modeling.	C
14	Explain the difference between declaring a cursor and opening a cursor.	Declaring a cursor retrieves data; opening a cursor defines its structure.	Declaring a cursor defines its structure; opening a cursor retrieves data.	Declaring a cursor and opening a cursor are the same.	Declaring a cursor is not a PL/SQL concept.	B
15	Explain the importance of transactions in PL/SQL and how they are managed.	Transactions are used for web development.	Transactions are used for data modeling.	Transactions ensure data consistency and are managed using COMMIT and ROLLBACK statements.	Transactions are not supported in PL/SQL.	C
16	Explain the purpose of a PL/SQL package and its components.	PL/SQL packages are used for web development.	PL/SQL packages are used for encapsulating procedures and functions.	PL/SQL packages are used for data modeling.	PL/SQL packages are used for hardware design.	B
17	How can you pass parameters to a PL/SQL procedure or function?	Parameters are passed using the CALL statement.	Parameters are not supported in PL/SQL.	Parameters are passed as input and output variables.	Parameters are passed using the DECLARE statement.	C
18	How can you resolve a deadlock in a database system?	By terminating one of the transactions involved in the deadlock.	By rolling back all transactions involved in the deadlock.	By increasing the isolation level.	Deadlocks cannot be resolved.	A
19	How do you create and manipulate PL/SQL associative arrays (index-by tables)?	Associative arrays are created using the ARRAY keyword.	Associative arrays are created using the INDEX keyword.	Associative arrays are not supported in PL/SQL.	Associative arrays are created using the TYPE keyword.	D
20	How do you declare a cursor, and what are the required components?	Cursors are automatically declared in PL/SQL.	Cursors are declared using the DECLARE CURSOR statement and require a SELECT statement.	Cursors are declared using the DECLARE keyword.	Cursors are declared using the OPEN statement.	B

21	How do you declare a variable in PL/SQL, and what are the data types supported for variables?	Variables are declared using the DECLARE keyword, and PL/SQL supports only one data type.	Variables are declared using the VAR keyword, and PL/SQL supports multiple data types.	Variables are declared using the VARIABLE keyword, and PL/SQL supports multiple data types.	Variables are not supported in PL/SQL.	B
22	How do you define and use PL/SQL records and record types?	Records are used for creating tables in PL/SQL.	Records are defined using the DECLARE RECORD statement.	Records are used to hold data in a structured format.	Records are not supported in PL/SQL.	C
23	How do you ensure that you've fetched all available data from a cursor?	By using the CLOSE statement.	By using the OPEN statement.	By checking the cursor attribute %NOTFOUND.	Cursors automatically fetch all available data.	C
24	How do you handle database connections and transactions in PL/SQL?	Database connections and transactions are automatically managed by the PL/SQL engine.	Database connections and transactions are not supported in PL/SQL.	Database connections are established using the CONNECT statement, and transactions are managed using COMMIT and ROLLBACK statements.	Database connections are established using the DECLARE statement.	C
25	How do you handle exceptions in PL/SQL? Provide an example.	Exceptions are handled using the IF-ELSE statement.	Exceptions are handled using the TRY-CATCH block.	Exceptions are handled using the EXCEPTION block.	Exceptions are not supported in PL/SQL.	C
26	How do you handle exceptions that may occur when working with cursors that have parameters?	By using the FETCH statement.	By ignoring exceptions and proceeding with the cursor operations.	By using exception handling techniques such as WHEN OTHERS and specific exception handlers for cursor-related errors.	Cursors with parameters do not raise exceptions.	C
27	How do you name a cursor, and what are some best practices for naming conventions?	Cursors are named automatically.	Cursors can be named using any random string.	Cursors should have meaningful names following naming conventions such as prefixing with CUR_.	Cursors cannot have names in PL/SQL.	C
28	How do you open a cursor to make it ready for data retrieval?	Use the DECLARE CURSOR statement.	Use the OPEN CURSOR statement.	Use the FETCH statement.	Cursors are automatically opened in PL/SQL.	B
29	How do you pass values to the cursor parameters when opening the cursor?	Use the FETCH statement to provide parameter values.	Use the SET PARAMETER statement.	Use a separate ASSIGN statement to assign values to parameters before opening the cursor.	Cursor parameters do not require values when opening.	C
30	How does a Cursor FOR Loop handle exceptions compared to explicit cursor processing?	Cursor FOR Loops do not support exception handling.	Cursor FOR Loops handle exceptions more gracefully by providing built-in error handling mechanisms.	Exception handling in Cursor FOR Loops is the same as in explicit cursor processing.	Cursor FOR Loops handle exceptions less efficiently than explicit cursors.	B
31	How does a cursor with parameters differ from a cursor without parameters in terms of flexibility?	Cursors with parameters are less flexible.	Cursors with parameters are more flexible because they can retrieve data based on varying conditions.	There is no difference in flexibility between the two types of cursors.	Cursors with parameters are slower.	B

32	How is the declaration of a cursor different from a regular SQL query?	Cursors cannot be used to retrieve data.	Cursors have a SELECT statement, while regular SQL queries are standalone.	Regular SQL queries cannot be used in PL/SQL.	There is no difference; they are the same.	B
33	In a multi-user database system, what does optimistic concurrency control aim to achieve?	It aims to prevent transactions from running concurrently.	It aims to avoid blocking and allow transactions to proceed concurrently, only checking for conflicts at the end.	It aims to lock all records to avoid conflicts.	It aims to roll back all transactions.	B
34	In database recovery, what is the difference between forward recovery and backward recovery?	Forward recovery restores the database to a previous state, while backward recovery recovers the database to its current state.	Forward recovery is the same as database backup, while backward recovery restores the database to a previous state.	Forward recovery involves log analysis, while backward recovery involves restoring database backups.	There is no difference; the terms are used interchangeably.	C
35	In database recovery, what is the purpose of a database log file?	To store user data.	To record changes made to the database for recovery purposes.	To create database backups.	To store database metadata.	B
36	used?	View	Commit	Rollback	Flashback	C
37	In SQL, what is the role of the ROLLBACK statement?	To save pending changes.	To begin a new transaction.	To undo all changes made during the current transaction.	To release locks on database records.	C
38	Is it necessary to declare a cursor inside a PL/SQL block, or can it be declared globally in a package?	Cursors can only be declared globally.	Cursors can only be declared inside a PL/SQL block.	Cursors can be declared both globally and inside a PL/SQL block.	Cursors are not supported in PL/SQL.	C
39	What are database triggers, and when might you use them in PL/SQL?	Database triggers are used for hardware design.	Database triggers are used for declaring variables.	Database triggers are used to automatically respond to database events and can be used for auditing or enforcing business rules.	Database triggers are used for creating tables.	C

40	What are some common use cases for using cursor parameters in PL/SQL?	Cursor parameters are rarely used in practice.	Common use cases include generating reports with different filter criteria, processing data based on user inputs, and customizing data retrieval based on changing conditions.	Cursor parameters are mainly used for database administration tasks.	Cursor parameters are only used in triggers.	B
41	What are the advantages of using explicit cursors over implicit cursors in PL/SQL?	Explicit cursors are faster in performance.	Implicit cursors are more flexible.	Explicit cursors are easier to use and provide more control.	Implicit cursors are automatically managed by the database.	C
42	What are the benefits of using PL/SQL for database programming compared to using SQL alone?	PL/SQL allows for creating web applications.	PL/SQL provides procedural capabilities for better control and encapsulation of logic in the database.	PL/SQL is used for hardware design.	PL/SQL is primarily used for data modeling.	B
43	What does the isolation level READ COMMITTED mean?	Reads data as it was when the transaction started.	Reads uncommitted changes made by other transactions.	Prevents any reads until the transaction is committed.	Reads data from committed transactions only.	A
44	What does the SAVEPOINT statement do in SQL?	Marks a point in a transaction to be rolled back to later.	Commits the transaction.	Opens a new transaction.	Locks the database.	A
45	What happens when you fetch data from a cursor that has no more rows to retrieve?	An error occurs.	The cursor is automatically closed.	The cursor remains open and ready for the next fetch.	Cursors always have more rows to retrieve.	A
46	What is a cursor in PL/SQL, and why is it used?	A cursor is a database table.	A cursor is used for looping through query results.	A cursor is a data type in PL/SQL.	A cursor is used for creating triggers.	B
47	What is a database checkpoint?	A physical location where the database is stored.	A marker indicating the point in time up to which transactions are considered safe and can be recovered.	A log file containing SQL statements.	A password for accessing the database.	B
48	What is a database lock in the context of concurrency control?	A mechanism to block all database transactions.	A mechanism to prevent data corruption.	A mechanism to prevent multiple transactions from accessing the same data simultaneously.	A mechanism to unlock databases.	C
49	What is a database restore operation?	A process that erases all data from the database.	A process that removes the database log files.	A process that brings a database back to a previous state by applying database backups and log files.	A process that upgrades the database to a new version.	C
50	What is a database transaction?	A single SQL statement.	A sequence of related SQL statements that are executed as a unit.	A database schema.	A database table.	B

51	What is a deadlock in the context of concurrency control?	A situation where a transaction is rolled back.	A situation where two or more transactions are waiting for each other to release locks.	A situation where a transaction is terminated.	A situation where a transaction is committed.	B
52	What is a distributed transaction in database management?	A transaction that involves multiple databases.	A transaction that is committed automatically.	A transaction with a large number of SQL statements.	A transaction without a COMMIT.	A
53	What is a full database backup?	A backup that includes only a subset of the database.	A backup that includes all the data and structures in the database.	A backup that contains only log files.	A backup that is encrypted for security.	B
54	What is a nested transaction in SQL?	A transaction inside another transaction.	A transaction without any nested SQL statements.	A transaction that cannot be rolled back.	A transaction with a SAVEPOINT.	A
55	What is a PL/SQL function, and how does it differ from a procedure?	A function is used for controlling database transactions.	A function is used for encapsulating reusable logic and returns a value.	A procedure is used for data modeling.	A procedure is used for creating tables.	B
56	What is concurrency control in database systems?	Managing multiple database transactions simultaneously.	Controlling access to the database using passwords.	Rolling back transactions in case of errors.	Creating indexes for database tables.	A
57	What is cursor positioning, and how does it relate to fetching data?	Cursor positioning determines the cursor's name.	Cursor positioning is the process of opening a cursor.	Cursor positioning refers to the current position of the cursor relative to the result set, affecting the next fetch operation.	Cursor positioning is not relevant in PL/SQL.	C
58	What is database recovery in the context of database management systems?	Backing up the database to prevent data loss.	The process of restoring a database to a previous state after a failure.	Increasing the database size to accommodate more data.	Encrypting database files for security.	B
59	What is dynamic SQL, and why might you use it in PL/SQL?	Dynamic SQL is used for creating triggers in PL/SQL.	Dynamic SQL allows you to generate and execute SQL statements at runtime.	Dynamic SQL is used for web development.	Dynamic SQL is used for hardware design.	B
60	What is PL/SQL, and how does it differ from SQL?	PL/SQL is a markup language for web development.	PL/SQL is a procedural extension of SQL.	PL/SQL is a data modeling language.	PL/SQL is a hardware description language.	B
61	What is the ACID property in the context of database transactions?	Atomicity, Consistency, Isolation, Durability	Aggregation, Continuity, Integrity, Durability	Affinity, Consistency, Isolation, Durability	Atomicity, Cancellation, Isolation, Division	A
62	What is the advantage of using a Cursor FOR Loop over traditional cursor processing?	Cursor FOR Loops are slower than traditional cursors.	Cursor FOR Loops offer less control.	Cursor FOR Loops simplify cursor processing by handling cursor declaration, opening, fetching, and closing automatically.	Cursor FOR Loops are not recommended in PL/SQL.	C

63	What is the benefit of using cursor parameters when working with data retrieval?	Cursor parameters make cursor declaration simpler.	Cursor parameters allow for dynamic queries and customization of data retrieval based on varying conditions.	Cursor parameters improve cursor performance.	Cursor parameters are not useful in PL/SQL.	B
64	What is the default behavior of a Cursor FOR Loop if there are no rows to process?	It raises an error.	It skips the loop and continues with the next statement.	It automatically exits the loop.	It waits for rows to be available.	C
65	What is the difference between a PL/SQL procedure and a PL/SQL function?	A procedure returns a value, while a function does not.	A procedure does not return a value, while a function does.	A procedure and a function are the same.	A procedure and a function are not supported in PL/SQL.	B
66	What is the opposite of a COMMIT statement in SQL?	ROLLBACK	BEGIN TRANSACTION	SAVEPOINT	LOCK	A
67	What is the primary drawback of using pessimistic concurrency control in a database system?	It can lead to data inconsistency.	It can result in excessive locking and reduced concurrency.	It is slower than optimistic concurrency control.	It requires frequent COMMIT statements.	B
68	What is the purpose of a differential backup in database recovery?	To recover the database to a specific point in time.	To restore only the data that has changed since the last full backup, reducing the recovery time.	To make a copy of the entire database.	To compress the database backup files.	B
69	What is the purpose of declaring a cursor in PL/SQL?	To insert data into a table.	To define a variable in PL/SQL.	To retrieve and manipulate query results in a controlled manner.	To create a trigger in PL/SQL.	C
70	What is the purpose of fetching data from a cursor in PL/SQL?	To insert data into a table.	To define a variable in PL/SQL.	To retrieve and manipulate query results row by row.	To create a trigger in PL/SQL.	C
71	What is the purpose of the COMMIT statement in SQL?	To roll back a transaction.	To save all pending changes permanently to the database.	To lock database records.	To create a new transaction.	B
72	What is the purpose of the FOR loop in PL/SQL, and how is it used?	The FOR loop is used for declaring variables.	The FOR loop is used for defining exceptions.	The FOR loop is used for iterative processing.	The FOR loop is used for database connections.	C
73	What is the purpose of the LOCK TABLE statement in SQL?	To unlock a table.	To create a new table.	To specify the locking mode for a table explicitly.	To commit a transaction.	C
74	What is the purpose of the WHEN OTHERS exception handler in PL/SQL?	It is used for declaring variables.	It is used for defining custom exceptions.	It is used to catch and handle unexpected exceptions.	It is used for database connections.	C
75	What is the role of a database backup in recovery?	It serves as a temporary storage location.	It records changes made to the database.	It provides a copy of the database that can be used to restore data in case of data loss or corruption.	It ensures data consistency during concurrent transactions.	C
76	What is the role of the FETCH statement in cursor processing?	It defines the cursor's name.	It specifies the number of rows to fetch.	It retrieves rows from the cursor into variables or records.	It opens the cursor for data retrieval.	C

77	What is the significance of the %ROWTYPE attribute when declaring a cursor?	It defines the cursor's name.	It specifies the number of rows the cursor can fetch.	It defines the structure of the result set the cursor will hold.	It determines the data type of cursor variables.	C
78	What is the significance of the recovery point objective (RPO) in database recovery planning?	It defines the maximum number of recovery points allowed.	It specifies the desired point in time to which a database should be recovered after a failure.	It defines the number of database logs to retain.	It measures the database's performance.	B
79	Which ACID property ensures that a transaction is completed in its entirety or not at all?	Atomicity	Consistency	Isolation	Durability	A

80	Which concurrency control technique allows conflicts to be detected and resolved only at the commit time?	Validation-based protocol	Timestamp ordering	Two-phase locking	Three-phase locking	A
81	Which database recovery model allows for point-in-time recovery to any arbitrary moment?	Simple recovery model	Full recovery model	Bulk-logged recovery model	Incremental recovery model	B
82	Which isolation level in SQL provides the highest level of isolation but can lead to concurrency issues?	READ COMMITTED	SERIALIZABLE	READ UNCOMMITTED	REPEATABLE READ	B
83	Which of the following is not a concurrency control mechanism in DBMS?	Locking	Timestamp ordering	Multiversion concurrency control	Rollback and recovery	D
84	Which of the following recovery techniques is based on maintaining multiple copies of the database at different points in time?	Replication	Deferred update	Redo logging	Undo logging	A
85	Which SQL statement is used to set a SAVEPOINT within a transaction?	BEGIN SAVEPOINT	SAVEPOINT	SET SAVEPOINT	CREATE SAVEPOINT	B
86	Which technique allows concurrent transactions to access different parts of a database without conflicts?	Pessimistic concurrency control	Optimistic concurrency control	Exclusive locking	Distributed transactions	B
87	[ON table_name] specifies the name of the table associated with the trigger.	Yes	No	Can be yes or no	None of the above	A
88	some events occur.	Procedure	Triggers	Collection	Transaction	B
89	A _________ consists of a sequence of query and/or update statements.	Transaction	Commit	Rollback	Flashback	A
90	A __________ is a special kind of a store procedure that executes in response to certain action on the table like insertion, deletion or updation of data.	Procedures	Triggers	Functions	None of the mentioned	B
91	A stored procedure in SQL is a___________	Block of functions	Group of Transact-SQL statements compiled into a single execution plan.	Group of distinct SQL statements.	None of the mentioned	B
92	A view is actually a?	composition of a table	decomposition of a table	associated to a table	None of the above	A
93	All objects placed in the specification are called __________ objects.	private	protected	public	None of the above	B
94	Any subprogram not in the package specification but coded in the package body is called a _________ object.	protected	private	self	public	B
95	Boyce-Codd Normal Form (BCNF) is an extension of which normal form?	A) First Normal Form (1NF)	B) Second Normal Form (2NF)	C) Third Normal Form (3NF)	D) Fourth Normal Form (4NF)	B
96	Consider the following action: TRANSACTION..... Commit; ROLLBACK; What does Rollback do?	Undoes the transactions before commit	Clears all transactions	Redoes the transactions before commit	No action	D
97	Consider the following cursor declaration in SQL: DECLARE cursor1 CURSOR FOR SELECT FirstName, LastName FROM Employees WHERE Department = 'Sales' If you want to open and fetch rows from this cursor, what SQL statement should	FETCH NEXT FROM cursor1;	**OPEN cursor1;**	CLOSE cursor1;	DECLARE cursor1 CURSOR FOR ...	B
98	Consider the following database schedule with two transactions, T1 and T2 S = r2(X); r1(X); r2(Y); w1(X); r1(Y); w2(X); a1; a2 where ri(Z) denotes a read operation by transaction Ti on a variable Z, wi(Z) denotes a write operation by Ti on a variable Z and ai denotes an abort by transaction Ti . Which one of the following statements about the above schedule is TRUE?	S is non-recoverable	S is recoverable, but has a cascading abort	S does not have a cascading abort	S is strict	C
99	Consider the following stored procedure in SQL: CREATE PROCEDURE CalculateTotalPrice @ProductID INT, @Quantity INT AS BEGIN DECLARE @Price DECIMAL(10, 2) SELECT @Price = UnitPrice FROM Products WHERE ProductID = @ProductID PRINT 'Total Price: ' + CAST(@Price * @Quantity AS VARCHAR) END If you call this stored procedure with @ProductID = 101 and @Quantity = 5, what will be printed?	Total Price: 505	**Total Price: 25**	Total Price: 101	Total Price: 5	B
100	Create function dept count(dept_name varchar(20)) begin declare d count integer; select count(*) into d count from instructor where instructor.dept_name= dept_name return d count; end Find the error in the the above statement.	Return type missing	Dept_name is mismatched	Reference relation is not mentioned	All of the mentioned	A

101	CREATE OR REPLACE FUNCTION calculate_gpa(
	  student_id NUMBER
	) RETURN NUMBER IS
	  total_points NUMBER := 0;
	  total_credits NUMBER := 0;
	  gpa NUMBER;
	BEGIN
	  -- Calculate GPA for a student
	  -- Assume the grade scale: A=4, B=3, C=2, D=1, F=0
	  -- Credits per subject: 3 credits
	  -- GPA = (Total Points) / (Total Credits)
	  RETURN gpa;
	END;
		Deletes students by age.	Updates student's name.	Calculates a student's GPA based on their grades.	Retrieves students in a subject above average.	C

102	CREATE OR REPLACE FUNCTION calculate_student_gpa(
	  student_id NUMBER
	) RETURN NUMBER IS
	  gpa NUMBER;
	BEGIN
	  -- Calculate GPA for a student based on their grades and credit hours
	  RETURN gpa;
	END;
		Deletes students by age.	Updates student's name.	Calculates a student's GPA based on their grades and credit hours.	Transfers students from one batch to another.	C

103	CREATE OR REPLACE FUNCTION calculate_subject_average(
	  subject_name VARCHAR2
	) RETURN NUMBER IS
	  avg_grade NUMBER;
	BEGIN
	  SELECT AVG(grade) INTO avg_grade
	  FROM student
	  WHERE subject = subject_name;
	  RETURN avg_grade;
	END;
		Deletes students by age.	Updates student information.	Calculates the average grade in a specific subject.	Returns a list of students with above-average grades in a subject.	C

104	CREATE OR REPLACE FUNCTION calculate_total_students RETURN NUMBER IS
	  total_students NUMBER;
	BEGIN
	  SELECT COUNT(*) INTO total_students
	  FROM student;
	  RETURN total_students;
	END;
		Deletes students by name.	Updates student information.	Calculates the total number of students.	Returns a list of students by age range.	C

105	CREATE OR REPLACE FUNCTION count_students_by_age(
	age NUMBER
	) RETURN NUMBER IS
	student_count NUMBER;
	BEGIN
	SELECT COUNT(*) INTO student_count
	FROM student
	WHERE age = age;
	RETURN student_count;
	END;
		Retrieves students by age.	Returns the total count of students by age.	Enrolls students in multiple subjects.	Deletes students by subject.	B

106	CREATE OR REPLACE FUNCTION count_students_in_subject(
	subject_name VARCHAR2
	) RETURN NUMBER IS
	student_count NUMBER;
	BEGIN
	SELECT COUNT(*) INTO student_count
	FROM student
	WHERE subject = subject_name;
	RETURN student_count;
	END;
		Retrieves students by subject count.	Returns a list of students with the maximum number of subjects.	Deletes students by age.	Updates a student's batch.	A

107	SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE age = (SELECT MAX(age) FROM student);
	RETURN students_cursor;
	END;
		Retrieves students with the lowest age.	Returns a list of students with the highest age.	Deletes students by age.	Awards scholarships to deserving students.	B

108	SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM (
	SELECT student_id, COUNT(DISTINCT subject) AS subject_count
	FROM student
	GROUP BY student_id
	ORDER BY subject_count DESC
	)
	WHERE ROWNUM = 1;
	RETURN students_cursor;
	END;
		Retrieves students with the highest grades.	Returns a list of subjects with above-average grades.	Enrolls students in multiple subjects.	Deletes students by subject.	B

109	CREATE OR REPLACE FUNCTION find_students_with_subject_count(
	subject_count NUMBER
	) RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM (
	SELECT student_id, COUNT(DISTINCT subject) AS subject_count
	FROM student
	GROUP BY student_id
	)
	WHERE subject_count = subject_count;
	RETURN students_cursor;
	END;
		Retrieves students by subject count.	Returns a list of students with a specific subject count.	Deletes students by age.	Updates a student's subject and grade.	A

110	SYS_REFCURSOR IS
	subjects_cursor SYS_REFCURSOR;
	BEGIN
	OPEN subjects_cursor FOR
	SELECT subject
	FROM (
	SELECT subject, COUNT(*) AS student_count
	FROM student
	GROUP BY subject
	ORDER BY student_count ASC
	)
	WHERE ROWNUM = 1;
	RETURN subjects_cursor;
	END;
		Retrieves subjects with the highest grades.	Returns a list of subjects with the lowest number of students.	Enrolls students in multiple subjects.	Deletes students by subject.	B

111	CREATE OR REPLACE FUNCTION get_highest_grade_by_subject(
	subject_name VARCHAR2
	) RETURN NUMBER IS
	highest_grade NUMBER;
	BEGIN
	SELECT MAX(grade) INTO highest_grade
	FROM student
	WHERE subject = subject_name;
	RETURN highest_grade;
	END;
		Returns the highest grade of all students.	Enrolls students in a subject.	Deletes students by subject.	Returns the highest grade in a specific subject.	D

112	CREATE OR REPLACE FUNCTION get_student_count_by_subject(
	subject_name VARCHAR2
	) RETURN NUMBER IS
	student_count NUMBER;
	BEGIN
	SELECT COUNT(*) INTO student_count
	FROM student
	WHERE subject = subject_name;
	RETURN student_count;
	END;
		Deletes a student by subject.	Returns the count of students in a specific subject.	Enrolls a student in a subject.	Updates the student's subject.	B

113	subject_name VARCHAR2
	) RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE subject = subject_name AND grade > (SELECT AVG(grade) FROM student WHERE subject = subject_name);
	RETURN students_cursor;
	END;
		Retrieves all students.	Returns a list of students with above-average grades in a specific subject.	Enrolls students in a subject.	Deletes students by subject.	B

114	CREATE OR REPLACE FUNCTION get_students_by_age_range(
	min_age NUMBER,
	max_age NUMBER
	) RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE age BETWEEN min_age AND max_age;
	RETURN students_cursor;
	END;
		Retrieves students by age range.	Returns a list of students with above-average grades in a specific subject.	Enrolls students in multiple subjects.	Deletes students by subject.	A

115	CREATE OR REPLACE FUNCTION get_students_by_grade_range(
	min_grade NUMBER,
	max_grade NUMBER
	) RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE grade BETWEEN min_grade AND max_grade;
	RETURN students_cursor;
	END;
		Retrieves students by grade range.	Returns the total count of students by grade range.	Awards scholarships to deserving students.	Transfers students from one batch to another.	A

116	CREATE OR REPLACE FUNCTION get_students_by_subject_and_grade(
	subject_name VARCHAR2,
	min_grade NUMBER
	) RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE subject = subject_name AND grade >= min_grade;
	RETURN students_cursor;
	END;
		Retrieves students by age range.	Returns a list of students with a specific subject and minimum grade.	Enrolls students in multiple subjects.	Deletes students by subject.	B

117	IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE batch_id IS NOT NULL;
	RETURN students_cursor;
	END;
		Retrieves students in a specific batch.	Returns a list of students with the highest age.	Enrolls students in multiple subjects.	Deletes students by subject.	A

118	subject_name VARCHAR2
	) RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE subject = subject_name
	AND grade > (SELECT AVG(grade) FROM student WHERE subject = subject_name);
	RETURN students_cursor;
	END;
		Retrieves all students.	Returns a list of students with above-average grades in a specific subject.	Enrolls students in a subject.	Updates student information.	B

119	SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE grade = (SELECT MAX(grade) FROM student);
	RETURN students_cursor;
	END;
		Retrieves students with the lowest grade.	Returns students with the highest grade.	Enrolls students in a batch.	Deletes students by age.	B

120	SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM student
	WHERE grade = (SELECT MIN(grade) FROM student);
	RETURN students_cursor;
	END;
		Retrieves students with the highest grade.	Returns students with the lowest grade.	Enrolls students in a batch.	Updates student information.	B

121	RETURN SYS_REFCURSOR IS
	students_cursor SYS_REFCURSOR;
	BEGIN
	OPEN students_cursor FOR
	SELECT student_id
	FROM (
	SELECT student_id, MAX(grade) AS max_grade
	FROM student
	GROUP BY student_id
	);
	RETURN students_cursor;
	END;
		Retrieves students with the subject wise lowest grades .	Returns students with the subject wise highest grades.	Enrolls students in multiple subjects.	Deletes students by subject.	B

122	CREATE OR REPLACE FUNCTION get_subjects_by_student(
	student_id NUMBER
	) RETURN SYS_REFCURSOR IS
	subjects_cursor SYS_REFCURSOR;
	BEGIN
	OPEN subjects_cursor FOR
	SELECT DISTINCT subject
	FROM student
	WHERE student_id = student_id;
	RETURN subjects_cursor;
	END;
		Deletes a student record.	Updates a student's information.	Returns a list of distinct subjects for a specific student.	Enrolls students in a subject.	C

123	SYS_REFCURSOR IS
	subjects_cursor SYS_REFCURSOR;
	BEGIN
	OPEN subjects_cursor FOR
	SELECT subject
	FROM (
	SELECT subject, MAX(grade) AS max_grade
	FROM student
	GROUP BY subject
	);
	RETURN subjects_cursor;
	END;
		Retrieves subjects with the highest grades.	Returns a list of subjects with above-average grades.	Enrolls students in multiple subjects.	Deletes students by subject.	A

124	RETURN SYS_REFCURSOR IS
	subjects_cursor SYS_REFCURSOR;
	BEGIN
	OPEN subjects_cursor FOR
	SELECT subject
	FROM (
	SELECT subject, AVG(grade) AS avg_grade
	FROM student
	GROUP BY subject
	)
	WHERE avg_grade > (SELECT AVG(grade) FROM student);
	RETURN subjects_cursor;
	END;
		Retrieves subjects with students above average grades.	Returns a list of subjects with students below average grades.	Enrolls students in multiple subjects.	Deletes students by subject.	A

125	CREATE OR REPLACE PROCEDURE archive_student_records IS
	BEGIN
	-- Write logic to archive old student records
	END;
		Deletes students by age.	Updates student information.	Archives old student records.	Awards scholarships to deserving students.	C

126	CREATE OR REPLACE PROCEDURE assign_student_grade(student_id NUMBER, subject_name VARCHAR2, grade NUMBER) IS BEGIN -- Write logic to assign a specific grade to a student in a subject END;	Deletes students by age.	Updates a student's name.	Assigns a specific grade to a student.	Promotes students to the next grade.	C
127	CREATE OR REPLACE PROCEDURE assign_student_subjects_and_grades(student_id NUMBER, subject_grades SYS.ODCINUMBERLIST) IS BEGIN -- Write logic to assign subjects and grades to a student END;	Deletes students by age.	Updates student information.	Assigns subjects and grades to a student.	Promotes students to the next grade.	C

128	CREATE OR REPLACE PROCEDURE delete_student(
	student_id NUMBER
	) IS
	BEGIN
	DELETE FROM student
	WHERE student_id = student_id;
	COMMIT;
	END;
		Retrieve student details by ID.	Update student information.	Enroll a new student.	Delete a student record.	D

129	CREATE OR REPLACE PROCEDURE delete_students_by_age(
	max_age NUMBER
	) IS
	BEGIN
	DELETE FROM student
	WHERE age > max_age;
	COMMIT;
	END;
		Deletes students by name.	Updates student age.	Deletes students older than a specified age.	Calculates students' GPA.	C

130	CREATE OR REPLACE PROCEDURE delete_students_by_batch(
	batch_id NUMBER
	) IS
	BEGIN
	DELETE FROM student
	WHERE batch_id = batch_id;
	COMMIT;
	END;
		Deletes students by age.	Updates student information.	Deletes students by batch.	Assigns subjects and grades to a student.	C

131	CREATE OR REPLACE PROCEDURE delete_students_by_subject(
	subject_name VARCHAR2
	) IS
	BEGIN
	DELETE FROM student
	WHERE subject = subject_name;
	COMMIT;
	END;
		Deletes students by age.	Updates student information.	Deletes students by subject.	Enrolls students in multiple subjects.	C

132	student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) IS BEGIN INSERT INTO student(student_id, name, age, subject, grade) VALUES(student_sequence.NEXTVAL, student_name, student_age, subject, student_grade); COMMIT; END;	Deletes a student record.	Updates a student's information.	Enrolls a new student with provided details.	Returns the count of students in a specific subject.	C
133	CREATE OR REPLACE PROCEDURE enroll_student_in_multiple_subjects(student_name VARCHAR2, student_age NUMBER, subjects VARCHAR2, student_grades VARCHAR2) IS BEGIN -- Write logic to enroll a student in multiple subjects with corresponding grades END;	Retrieve student details by ID.	Update student information.	Enroll students in multiple subjects.	Delete students by subject.	C

134	CREATE OR REPLACE PROCEDURE increase_student_grades(
	grade_increase NUMBER
	) IS
	BEGIN
	UPDATE student
	SET grade = grade + grade_increase;
	COMMIT;
	END;
		Deletes students by age.	Updates student information.	Increases student grades.	Awards scholarships to deserving students.	C

135	CREATE OR REPLACE PROCEDURE print_student_details(
	student_id NUMBER
	) IS
	student_name VARCHAR2(100);
	student_age NUMBER;
	subject_name VARCHAR2(50);
	student_grade NUMBER;
	BEGIN
	SELECT name, age, subject, grade
	INTO student_name, student_age, subject_name, student_grade
	FROM student
	WHERE student_id = student_id;
	
	DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_id);
	DBMS_OUTPUT.PUT_LINE('Name: ' || student_name);
	DBMS_OUTPUT.PUT_LINE('Age: ' || student_age);
		Deletes a student record.	Updates student information.	Prints details of a student by ID.	Returns the count of students in a specific subject.	C

136	CREATE OR REPLACE PROCEDURE print_student_transcript(
	student_id NUMBER
	) IS
	BEGIN
	-- Write logic to print the student's transcript
	END;
		Deletes students by age.	Updates student information.	Prints a student's transcript.	Returns students with the lowest grade.	C

137	student_id NUMBER
	) IS
	BEGIN
	FOR rec IN (SELECT subject, grade FROM student WHERE student_id = student_id) LOOP
	DBMS_OUTPUT.PUT_LINE('Subject: ' || rec.subject || ', Grade: ' || rec.grade);
	END LOOP;
	END;
		Deletes a student record.	Updates student information.	Prints subjects and grades of a student by ID.	Returns a list of subjects in which a student is enrolled.	C

138	CREATE OR REPLACE PROCEDURE transfer_student(student_id NUMBER, new_batch_id NUMBER) IS BEGIN -- Write logic to transfer a student to a new batch END;	Deletes students by age.	Updates student information.	Transfers a student to a new batch.	Calculates the average grade in a subject.	C

139	CREATE OR REPLACE PROCEDURE update_student_age(
	student_id NUMBER,
	new_age NUMBER
	) IS
	BEGIN
	UPDATE student
	SET age = new_age
	WHERE student_id = student_id;
	COMMIT;
	END;
	/
		Deletes a student record.	Updates a student's age.	Enrolls a new student.	Returns the count of students in a specific subject.	B

140	CREATE OR REPLACE PROCEDURE update_student_name(
	student_id NUMBER,
	new_name VARCHAR2
	) IS
	BEGIN
	UPDATE student
	SET name = new_name
	WHERE student_id = student_id;
	COMMIT;
	END;
		Deletes students by age.	Updates a student's name.	Enrolls a new student.	Returns the count of students in a specific subject.	B

141	CREATE OR REPLACE PROCEDURE update_student_subject_and_grade(student_id NUMBER, subject_name VARCHAR2, new_grade NUMBER) IS BEGIN -- Write logic to update a student's subject and grade END;	Deletes a student's subject and grade.	Updates a student's subject and grade.	Enrolls a new student.	Returns a student's subject and grade.	B

142	Create procedure dept_count proc(in dept name varchar(20),
	out d count integer)
	begin
	select count(*) into d count
	from instructor
	where instructor.dept name= dept count proc.dept name
	end
	Which of the following is used to call the procedure given above ?
		Declare d_count integer;	Declare d_count integer;
			call dept_count proc('Physics', d_count);
				Declare d_count integer;
				call dept_count proc('Physics');
					Declare d_count;
					call dept_count proc('Physics', d_count);
						B

143	Declare out of classroom seats condition DECLARE exit handler FOR OUT OF classroom seats BEGIN SEQUENCE OF statements END The above statements are used for	Calling procedures	Handling Exception	Handling procedures	All of the mentioned	B
144	internal database error.	Yes	No	1 or 2	All of the above	A

145	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to calculate the average grade for each subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		FOR subject_rec IN subject_cursor LOOP
		-- Calculate and print the average grade for each subject
		END LOOP;
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			avg_grade NUMBER;
			BEGIN
			FOR subject_rec IN subject_cursor LOOP
			-- Calculate and store the average grade for each subject in avg_grade
			
			DBMS_OUTPUT.PUT_LINE('Subject: ' || subject_rec.subject || ', Avg Grade: ' || avg_grade);
			END LOOP;
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				FOR subject_rec IN subject_cursor LOOP
				-- Calculate and print the average grade for each subject
				END LOOP;
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					FOR subject_rec IN subject_cursor LOOP
					-- Calculate and print the average grade for each subject
					END LOOP;
					END;
						B

146	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to fetch all students who have a grade greater than or equal to 90 in the subject "Mathematics." Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR high_math_grades IS
		SELECT student_id
		FROM student
		WHERE subject = 'Mathematics' AND grade >= 90;
		BEGIN
		FOR rec IN high_math_grades LOOP
		
		DBMS_OUTPUT.PUT_LINE('Student ID: ' || rec.student_id);
		END LOOP;
		END;
			DECLARE
			CURSOR high_math_grades IS
			SELECT student_id
			FROM student
			WHERE subject = 'Mathematics' AND grade >= 90;
			student_recordstudent%ROWTYPE;
			BEGIN
			OPEN high_math_grades;
			LOOP
			FETCH high_math_grades INTO student_record;
			EXIT WHEN high_math_grades%NOTFOUND;
			
			DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_record.student_id);
			END LOOP;
			CLOSE high_math_grades;
			END;
				DECLARE
				CURSOR high_math_grades IS
				SELECT student_id
				FROM student
				WHERE subject = 'Mathematics' AND grade >= 90;
				student_id NUMBER;
				BEGIN
				OPEN high_math_grades;
				LOOP
				FETCH high_math_grades INTO student_id;
				EXIT WHEN high_math_grades%NOTFOUND;
				
				DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_id);
				END LOOP;
				CLOSE high_math_grades;
				END;
					DECLARE
					CURSOR high_math_grades(student_id NUMBER) IS
					SELECT student_id
					FROM student
					WHERE subject = 'Mathematics' AND grade >= 90;
					BEGIN
					FOR rec IN high_math_grades(90) LOOP
					DBMS_OUTPUT.PUT_LINE('Student ID: ' || rec.student_id);
					END LOOP;
					END;
						B

147	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the student with the highest grade for each subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		FOR subject_rec IN subject_cursor LOOP
		-- Find and print the student with the highest grade for each subject
		END LOOP;
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			highest_grade NUMBER;
			BEGIN
			FOR subject_rec IN subject_cursor LOOP
			-- Calculate and store the highest grade for each subject in highest_grade
			
			DBMS_OUTPUT.PUT_LINE('Subject: ' || subject_rec.subject || ', Highest Grade: ' || highest_grade);
			END LOOP;
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				FOR subject_rec IN subject_cursor LOOP
				-- Find and print the student with the highest grade for each subject
				END LOOP;
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					FOR subject_rec IN subject_cursor LOOP
					-- Find and print the student with the highest grade for each subject
					END LOOP;
					END;
						A

148	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have improved their grades in at least one subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR student_cursor IS
		SELECT DISTINCT student_id
		FROM student;
		BEGIN
		-- Find and print students who have improved their grades
		END;
			DECLARE
			CURSOR student_cursor IS
			SELECT DISTINCT student_id
			FROM student;
			improved_students VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of improved students in improved_students
			
			DBMS_OUTPUT.PUT_LINE('Improved Students: ' || improved_students);
			END;
				DECLARE
				CURSOR student_cursor IS
				SELECT DISTINCT student_id
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students who have improved their grades
				END;
					DECLARE
					CURSOR student_cursor IS
					SELECT DISTINCT student_id
					FROM student;
					BEGIN
					-- Find and print students who have improved their grades
					END;
						B

149	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR student_cursor IS
		    SELECT DISTINCT student_id
		    FROM student;
		BEGIN
		  -- Find and print students who have not improved their grades
		END;
			DECLARE
			  CURSOR student_cursor IS
			    SELECT DISTINCT student_id
			    FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				  CURSOR student_cursor IS
				    SELECT DISTINCT student_id
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students who have not improved their grades
				END;
					DECLARE
					  CURSOR student_cursor IS
					    SELECT DISTINCT student_id
					    FROM student;
					BEGIN
					  -- Find and print students who have not improved their grades
					END;
						B

150	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR student_cursor IS
		    SELECT DISTINCT student_id
		    FROM student;
		BEGIN
		  -- Find and print students who have not improved their grades
		END;
			DECLARE
			  CURSOR student_cursor IS
			    SELECT DISTINCT student_id
			    FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				  CURSOR student_cursor IS
				    SELECT DISTINCT student_id
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students who have not improved their grades
				END;
					DECLARE
					  CURSOR student_cursor IS
					    SELECT DISTINCT student_id
					    FROM student;
					BEGIN
					  -- Find and print students who have not improved their grades
					END;
						B

151	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR student_cursor IS
		    SELECT DISTINCT student_id
		    FROM student;
		BEGIN
		  -- Find and print students who have not improved their grades
		END;
			DECLARE
			  CURSOR student_cursor IS
			    SELECT DISTINCT student_id
			    FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				  CURSOR student_cursor IS
				    SELECT DISTINCT student_id
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students who have not improved their grades
				END;
					DECLARE
					  CURSOR student_cursor IS
					    SELECT DISTINCT student_id
					    FROM student;
					BEGIN
					  -- Find and print students who have not improved their grades
					END;
						B

152	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR student_cursor IS
		    SELECT DISTINCT student_id
		    FROM student;
		BEGIN
		  -- Find and print students who have not improved their grades
		END;
			DECLARE
			  CURSOR student_cursor IS
			    SELECT DISTINCT student_id
			    FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				  CURSOR student_cursor IS
				    SELECT DISTINCT student_id
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students who have not improved their grades
				END;
					DECLARE
					  CURSOR student_cursor IS
					    SELECT DISTINCT student_id
					    FROM student;
					BEGIN
					  -- Find and print students who have not improved their grades
					END;
						B

153	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR student_cursor IS
		SELECT DISTINCT student_id
		FROM student;
		BEGIN
		-- Find and print students who have not improved their grades
		END;
			DECLARE
			CURSOR student_cursor IS
			SELECT DISTINCT student_id
			FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				CURSOR student_cursor IS
				SELECT DISTINCT student_id
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students who have not improved their grades
				END;
					DECLARE
					CURSOR student_cursor IS
					SELECT DISTINCT student_id
					FROM student;
					BEGIN
					-- Find and print students who have not improved their grades
					END;
						B

154	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR student_cursor IS
		SELECT DISTINCT student_id
		FROM student;
		BEGIN
		-- Find and print students who have not improved their grades
		END;
			DECLARE
			CURSOR student_cursor IS
			SELECT DISTINCT student_id
			FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				CURSOR student_cursor IS
				SELECT DISTINCT student_id
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students who have not improved their grades
				END;
					DECLARE
					CURSOR student_cursor IS
					SELECT DISTINCT student_id
					FROM student;
					BEGIN
					-- Find and print students who have not improved their grades
					END;
						B

155	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have not improved their grades in any subject compared to the previous year. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR student_cursor IS
		SELECT DISTINCT student_id
		FROM student;
		BEGIN
		-- Find and print students who have not improved their grades
		END;
			DECLARE
			CURSOR student_cursor IS
			SELECT DISTINCT student_id
			FROM student;
			not_improved_students VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of students who have not improved their grades in not_improved_students
			
			DBMS_OUTPUT.PUT_LINE('Students Who Have Not Improved Their Grades: ' || not_improved_students);
			END;
				DECLARE
				CURSOR student_cursor IS
				SELECT DISTINCT student_id
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students who have not improved their grades
				END;
					DECLARE
					CURSOR student_cursor IS
					SELECT DISTINCT student_id
					FROM student;
					BEGIN
					-- Find and print students who have not improved their grades
					END;
						B

156	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the highest grade in at least one subject and the lowest grade in at least one subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR student_cursor IS
		SELECT DISTINCT student_id
		FROM student;
		BEGIN
		-- Find and print students with highest and lowest grades in subjects
		END;
			DECLARE
			CURSOR student_cursor IS
			SELECT DISTINCT student_id
			FROM student;
			students_with_extreme_grades VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of students with extreme grades in students_with_extreme_grades
			
			DBMS_OUTPUT.PUT_LINE('Students with Extreme Grades: ' || students_with_extreme_grades);
			END;
				DECLARE
				CURSOR student_cursor IS
				SELECT DISTINCT student_id
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students with highest and lowest grades in subjects
				END;
					DECLARE
					CURSOR student_cursor IS
					SELECT DISTINCT student_id
					FROM student;
					BEGIN
					-- Find and print students with highest and lowest grades in subjects
					END;
						B

157	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the highest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print students with the highest grade in each subject
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			highest_grade NUMBER;
			BEGIN
			-- Calculate and store the highest grade for each subject in highest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Highest Grade in Each Subject: ' || highest_grade);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students with the highest grade in each subject
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print students with the highest grade in each subject
					END;
						B

158	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the highest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print students with the highest grade in each subject
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			highest_grade NUMBER;
			BEGIN
			-- Calculate and store the highest grade for each subject in highest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Highest Grade in Each Subject: ' || highest_grade);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students with the highest grade in each subject
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print students with the highest grade in each subject
					END;
						B

159	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the highest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print students with the highest grade in each subject
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			highest_grade NUMBER;
			BEGIN
			-- Calculate and store the highest grade for each subject in highest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Highest Grade in Each Subject: ' || highest_grade);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students with the highest grade in each subject
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print students with the highest grade in each subject
					END;
						B

160	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the highest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print students with the highest grade in each subject
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			highest_grade NUMBER;
			BEGIN
			-- Calculate and store the highest grade for each subject in highest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Highest Grade in Each Subject: ' || highest_grade);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students with the highest grade in each subject
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print students with the highest grade in each subject
					END;
						B

161	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the highest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print students with the highest grade in each subject
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			highest_grade NUMBER;
			BEGIN
			  -- Calculate and store the highest grade for each subject in highest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Highest Grade in Each Subject: ' || highest_grade);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students with the highest grade in each subject
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print students with the highest grade in each subject
					END;
						B

162	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the lowest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print students with the lowest grade in each subject
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			lowest_grade NUMBER;
			BEGIN
			  -- Calculate and store the lowest grade for each subject in lowest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Lowest Grade in Each Subject: ' || lowest_grade);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students with the lowest grade in each subject
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print students with the lowest grade in each subject
					END;
						B

163	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the lowest grade in each subject. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print students with the lowest grade in each subject
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			lowest_grade NUMBER;
			BEGIN
			  -- Calculate and store the lowest grade for each subject in lowest_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Lowest Grade in Each Subject: ' || lowest_grade);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students with the lowest grade in each subject
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print students with the lowest grade in each subject
					END;
						B

164	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the same grade in all subjects. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR student_cursor IS
		    SELECT DISTINCT student_id
		    FROM student;
		BEGIN
		  -- Find and print students who have scored the same grade in all subjects
		END;
			DECLARE
			  CURSOR student_cursor IS
			    SELECT DISTINCT student_id
			    FROM student;
			students_with_same_grade VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of students with the same grade in all subjects in students_with_same_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Same Grade in All Subjects: ' || students_with_same_grade);
			END;
				DECLARE
				  CURSOR student_cursor IS
				    SELECT DISTINCT student_id
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students who have scored the same grade in all subjects
				END;
					DECLARE
					  CURSOR student_cursor IS
					    SELECT DISTINCT student_id
					    FROM student;
					BEGIN
					  -- Find and print students who have scored the same grade in all subjects
					END;
						B

165	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the students who have scored the same grade in all subjects. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR student_cursor IS
		SELECT DISTINCT student_id
		FROM student;
		BEGIN
		-- Find and print students who have scored the same grade in all subjects
		END;
			DECLARE
			CURSOR student_cursor IS
			SELECT DISTINCT student_id
			FROM student;
			students_with_same_grade VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of students with the same grade in all subjects in students_with_same_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Same Grade in All Subjects: ' || students_with_same_grade);
			END;
				DECLARE
				CURSOR student_cursor IS
				SELECT DISTINCT student_id
				FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print students who have scored the same grade in all subjects
				END;
					DECLARE
					CURSOR student_cursor IS
					SELECT DISTINCT student_id
					FROM student;
					BEGIN
					-- Find and print students who have scored the same grade in all subjects
					END;
						B

166	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subject with the highest overall grades (sum of grades for all students). Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print the subject with the highest overall grades
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			highest_subject VARCHAR2(100);
			BEGIN
			-- Calculate and store the subject with the highest overall grades in highest_subject
			
			DBMS_OUTPUT.PUT_LINE('Subject with Highest Overall Grades: ' || highest_subject);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print the subject with the highest overall grades
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print the subject with the highest overall grades
					END;
						B

167	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subject with the lowest average grade. Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print the subject with the lowest average grade
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			lowest_avg_grade NUMBER;
			BEGIN
			-- Calculate and store the lowest average grade for a subject in lowest_avg_grade
			
			DBMS_OUTPUT.PUT_LINE('Subject with Lowest Average Grade: ' || lowest_avg_grade);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print the subject
				
				with the lowest average grade
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print the subject with the lowest average grade
					END;
						B

168	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which all students have scored above a specified threshold (e.g., 60). Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print subjects with all scores above the threshold
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT
			
			subject
			FROM student;
			subjects_with_all_high_scores VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of subjects with all high scores in subjects_with_all_high_scores
			
			DBMS_OUTPUT.PUT_LINE('Subjects with All High Scores: ' || subjects_with_all_high_scores);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print subjects with all scores above the threshold
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print subjects with all scores above the threshold
					END;
						B

<table>
<tr><td>169</td><td>Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which all students have scored above a specified threshold (e.g., 70). Which of the following code snippets accomplishes this task?</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with all scores above the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subjects_with_all_high_scores VARCHAR2(4000);
BEGIN
-- Calculate and store the list of subjects with all high scores in subjects_with_all_high_scores

DBMS_OUTPUT.PUT_LINE('Subjects with All High Scores: ' || subjects_with_all_high_scores);
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subject_recordstudent%ROWTYPE;
BEGIN
-- Find and print subjects with all scores above the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with all scores above the threshold
END;</td><td>B</td></tr>
<tr><td>170</td><td>Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which at least one student has scored below a specified threshold (e.g., 40). Which of the following code snippets accomplishes this task?</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subjects_with_low_scores VARCHAR2(4000);
BEGIN
-- Calculate and store the list of subjects with at least one low score in subjects_with_low_scores

DBMS_OUTPUT.PUT_LINE('Subjects with Low Scores: ' || subjects_with_low_scores);
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subject_recordstudent%ROWTYPE;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>A</td></tr>
<tr><td>171</td><td>Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which at least one student has scored below a specified threshold (e.g., 50). Which of the following code snippets accomplishes this task?</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subjects_with_low_scores VARCHAR2(4000);
BEGIN
-- Calculate and store the list of subjects with at least one low score in subjects_with_low_scores

DBMS_OUTPUT.PUT_LINE('Subjects with Low Scores: ' || subjects_with_low_scores);
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subject_recordstudent%ROWTYPE;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>B</td></tr>
<tr><td>172</td><td>Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which at least one student has scored below a specified threshold (e.g., 50). Which of the following code snippets accomplishes this task?</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subjects_with_low_scores VARCHAR2(4000);
BEGIN
-- Calculate and store the list of subjects with at least one low score in subjects_with_low_scores

DBMS_OUTPUT.PUT_LINE('Subjects with Low Scores: ' || subjects_with_low_scores);
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
subject_recordstudent%ROWTYPE;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>DECLARE
CURSOR subject_cursor IS
SELECT DISTINCT subject
FROM student;
BEGIN
-- Find and print subjects with at least one score below the threshold
END;</td><td>A</td></tr>
</table>

173	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which at least one student has scored below a specified threshold (e.g., 55). Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print subjects with at least one score below the threshold
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			subjects_with_low_scores VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of subjects with at least one low score in subjects_with_low_scores
			
			DBMS_OUTPUT.PUT_LINE('Subjects with Low Scores: ' || subjects_with_low_scores);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print subjects with at least one score below the threshold
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print subjects with at least one score below the threshold
					END;
						B

174	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which at least one student has scored below a specified threshold (e.g., 60). Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print subjects with at least one score below the threshold
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			subjects_with_low_scores VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of subjects with at least one low score in subjects_with_low_scores
			
			DBMS_OUTPUT.PUT_LINE('Subjects with Low Scores: ' || subjects_with_low_scores);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print subjects with at least one score below the threshold
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print subjects with at least one score below the threshold
					END;
						B

175	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which every student has scored above a specified threshold (e.g., 85). Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print subjects with all scores above the threshold
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			subjects_with_all_high_scores VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of subjects with all high scores in subjects_with_all_high_scores
			
			DBMS_OUTPUT.PUT_LINE('Subjects with All High Scores: ' || subjects_with_all_high_scores);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print subjects with all scores above the threshold
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print subjects with all scores above the threshold
					END;
						B

176	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which every student has scored above a specified threshold (e.g., 90). Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print subjects with all scores above the threshold
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			subjects_with_all_high_scores VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of subjects with all high scores in subjects_with_all_high_scores
			
			DBMS_OUTPUT.PUT_LINE('Subjects with All High Scores: ' || subjects_with_all_high_scores);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print subjects with all scores above the threshold
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print subjects with all scores above the threshold
					END;
						B

177	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which no student has scored below a specified threshold (e.g., 60). Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print subjects with no scores below the threshold
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			subjects_with_no_low_scores VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of subjects with no low scores in subjects_with_no_low_scores
			
			DBMS_OUTPUT.PUT_LINE('Subjects with No Low Scores: ' || subjects_with_no_low_scores);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print subjects with no scores below the threshold
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print subjects with no scores below the threshold
					END;
						B

178	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which the average grade is above a specified threshold (e.g., 75). Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print subjects with an average grade above the threshold
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			subjects_above_threshold VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of subjects with an average grade above the threshold in subjects_above_threshold
			
			DBMS_OUTPUT.PUT_LINE('Subjects with Average Grade Above Threshold: ' || subjects_above_threshold);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print subjects with an average grade above the threshold
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print subjects with an average grade above the threshold
					END;
						B

179	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which the average grade is above a specified threshold (e.g., 75). Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print subjects with an average grade above the threshold
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			subjects_above_threshold VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of subjects with an average grade above the threshold in subjects_above_threshold
			
			DBMS_OUTPUT.PUT_LINE('Subjects with Average Grade Above Threshold: ' || subjects_above_threshold);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print subjects with an average grade above the threshold
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print subjects with an average grade above the threshold
					END;
						B

180	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which the average grade is above a specified threshold (e.g., 80). Which of the following code snippets accomplishes this task?	DECLARE
		CURSOR subject_cursor IS
		SELECT DISTINCT subject
		FROM student;
		BEGIN
		-- Find and print subjects with an average grade above the threshold
		END;
			DECLARE
			CURSOR subject_cursor IS
			SELECT DISTINCT subject
			FROM student;
			subjects_above_threshold VARCHAR2(4000);
			BEGIN
			-- Calculate and store the list of subjects with an average grade above the threshold in subjects_above_threshold
			
			DBMS_OUTPUT.PUT_LINE('Subjects with Average Grade Above Threshold: ' || subjects_above_threshold);
			END;
				DECLARE
				CURSOR subject_cursor IS
				SELECT DISTINCT subject
				FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				-- Find and print subjects with an average grade above the threshold
				END;
					DECLARE
					CURSOR subject_cursor IS
					SELECT DISTINCT subject
					FROM student;
					BEGIN
					-- Find and print subjects with an average grade above the threshold
					END;
						B

181	Given a "student" table with columns `student_id`, `subject`, and `grade`, write a PL/SQL block that uses a cursor to find and print the subjects in which the average grade is below a specified threshold (e.g., 70). Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR subject_cursor IS
		    SELECT DISTINCT subject
		    FROM student;
		BEGIN
		  -- Find and print subjects with an average grade below the threshold
		END;
			DECLARE
			  CURSOR subject_cursor IS
			    SELECT DISTINCT subject
			    FROM student;
			subjects_below_threshold VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of subjects with an average grade below the threshold in subjects_below_threshold
			
			DBMS_OUTPUT.PUT_LINE('Subjects with Average Grade Below Threshold: ' || subjects_below_threshold);
			END;
				DECLARE
				  CURSOR subject_cursor IS
				    SELECT DISTINCT subject
				    FROM student;
				subject_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print subjects with an average grade below the threshold
				END;
					DECLARE
					  CURSOR subject_cursor IS
					    SELECT DISTINCT subject
					    FROM student;
					BEGIN
					  -- Find and print subjects with an average grade below the threshold
					END;
						B

182	Given a "student" table with columns `student_id`, `subject`, and `grade`,write a PL/SQL block that uses a cursor to find and print the students who have scored the same grade in all subjects. Which of the following code snippets accomplishes this task?	DECLARE
		  CURSOR student_cursor IS
		    SELECT DISTINCT student_id
		    FROM student;
		BEGIN
		  -- Find and print students who have scored the same grade in all subjects
		END;
			DECLARE
			  CURSOR student_cursor IS
			    SELECT DISTINCT student_id
			    FROM student;
			students_with_same_grade VARCHAR2(4000);
			BEGIN
			  -- Calculate and store the list of students with the same grade in all subjects in students_with_same_grade
			
			DBMS_OUTPUT.PUT_LINE('Students with Same Grade in All Subjects: ' || students_with_same_grade);
			END;
				DECLARE
				  CURSOR student_cursor IS
				    SELECT DISTINCT student_id
				    FROM student;
				student_recordstudent%ROWTYPE;
				BEGIN
				  -- Find and print students who have scored the same grade in all subjects
				END;
					DECLARE
					  CURSOR student_cursor IS
					    SELECT DISTINCT student_id
					    FROM student;
					BEGIN
					  -- Find and print students who have scored the same grade in all subjects
					END;
						B

183	How can concurrent access to shared data lead to data inconsistency in a DBMS?	A) By preventing data updates	B) By enforcing data integrity rules	C) By allowing simultaneous updates	D) By reducing query performance	C
184	How can you remove a package from the database in PL/SQL?	By using the DROP PACKAGE statement	By removing all the procedures and functions from the package body	By using the DELETE PACKAGE statement	By using the TRUNCATE PACKAGE statement	A
185	How many mandatory parts packages has?	1	2	3	4	B
186	In a DBMS, a package is typically used to:	Store and organize tables and views	Group related procedures, functions, and variables	Define user roles and permissions	Execute ad-hoc SQL queries	B
187	In a DBMS, what is a benefit of using stored procedures for business logic?	It allows for dynamic SQL execution.	**It centralizes and secures the business logic.**	It simplifies data retrieval operations.	It eliminates the need for data validation.	B
188	In a DBMS, what is a database trigger?	A procedure that runs when the database is created	A statement that rolls back database changes	A set of rules for data validation	**A piece of code that automatically executes in response to a specific event**	D
189	In a DBMS, what is the main purpose of a database trigger?	To enforce referential integrity constraints	To encapsulate business logic for data processing	To automatically generate primary keys	To record changes to data in response to events	D
190	In a DBMS, what is the primary purpose of a "SERIALIZABLE" isolation level?	To maximize concurrency	To prevent transactions from acquiring locks	To ensure that transactions execute in a specific order	**To provide the highest level of data consistency and isolation**	D
191	In a DBMS, what is the primary purpose of a database package body?	A) To define package variables	B) To provide information about the package's procedures and functions	C) To specify package triggers	D) To implement the actual code for package procedures	D
192	In a DBMS, what is the primary purpose of using "SELECT FOR UPDATE"?	To retrieve data for reporting purposes	**To lock rows, preventing other transactions from modifying them**	To retrieve data without acquiring locks	To roll back a transaction	B

193	In a DBMS, what is the primary purpose of using a "READ COMMITTED" isolation level?	To minimize data redundancy	To ensure that transactions execute in a specific order	To prevent transactions from acquiring locks	**To balance data consistency and concurrency**	D
194	In a DBMS, what is the primary purpose of using an "AFTER UPDATE" trigger?	To prevent any updates from occurring	To execute before any update operation occurs	**To log changes made to a table after an update**	To execute after a row is updated in a table	C
195	In a DBMS, what is the primary purpose of using triggers?	To simplify data retrieval operations	**To enforce data integrity constraints**	To create temporary tables	To improve query performance	B
196	In a DBMS, what is the purpose of a "SAVE TRANSACTION" statement?	To save a transaction for later execution	To create a new transaction	**To save the current state of a transaction as a savepoint**	To commit the transaction	C
197	In a DBMS, what is the purpose of a savepoint?	To commit a transaction	To roll back a transaction	**To create a point within a transaction to which you can later roll back**	To lock a table temporarily	C
198	In a DBMS, what is the purpose of the "COMMIT" statement?	To start a new transaction	**To save all changes made within the current transaction**	To roll back all changes made within the current transaction	To create a new savepoint	B
199	In a DBMS, what is the purpose of the "ROLLBACK TO SAVEPOINT" statement?	To roll back an entire transaction	To create a new savepoint	**To roll back to a specific savepoint within a transaction**	To commit a transaction	C
200	In a DBMS, what is the purpose of the "SET TRANSACTION" statement?	**To define transaction isolation levels**	To start a new transaction	To commit a transaction	To roll back a transaction	A
201	In a DBMS, what is the purpose of the procedure header?	A) To declare the procedure's input and output variables	B) To specify the operations to be performed by the procedure	C) To implement the actual code for the procedure	D) To define the package specification	A
202	In a DBMS, what is the term for a package that only contains the package specification without a package body?	A) A minimal package	B) A comprehensive package	C) A bodiless package	D) A complete package	C
203	In a DBMS, what is the typical process of developing a package?	A) Write the package body first, then the specification	B) Write the package specification first, then the body	C) Develop procedures and triggers independently of packages	D) Develop triggers before procedures	B
204	In a DBMS, which clause is used to specify the action to be taken when a trigger event occurs?	EXECUTE	FOR EACH ROW	**WHEN**	INSTEAD OF	C
205	In a DBMS, which type of cursor is used to fetch and process one row at a time?	Static cursor	Dynamic cursor	**Forward-only cursor**	Scrollable cursor	C
206	In a locking-based concurrency control system, what does a "lock" prevent other transactions from doing?	A) Accessing the locked data	B) Aborting the transaction	C) Executing queries	D) Creating database triggers	A
207	In a multi-user DBMS, what issue can occur without proper concurrency control?	A) Faster query execution	B) Data consistency problems	C) Reduced code duplication	D) Increased code modularity	B
208	In a multi-user DBMS, what problem can arise when transactions are executed concurrently without control?	A) Faster data retrieval	B) Enhanced data consistency	C) Reduced code duplication	D) Increased code modularity	B
209	In Oracle PL/SQL, can a package contain both procedures and functions?	Yes, but they must all have the same name.	No, a package can only contain either procedures or functions, not both.	**Yes, a package can contain a mix of procedures and functions.**	Yes, but they must all be stored in separate packages.	C
210	In Oracle PL/SQL, what is a benefit of using packages over standalone procedures?	Packages are easier to write and debug.	**Packages allow you to encapsulate related code and data.**	Packages execute faster than standalone procedures.	Packages are not dependent on any database schema.	B
211	In Oracle PL/SQL, which statement is used to raise an exception explicitly within a stored procedure?	RAISE EXCEPTION	SIGNAL	**RAISE_APPLICATION_ERROR**	THROW	C

212	In PL/SQL, the CREATE TABLESPACE is used	To create a place in the database for storage of scheme objects, rollback segments, and naming the data files to comprise the table-space	To create a database trigger	To add/rename data files, to change storage	All of the above	A
213	In PL/SQL, which of the following statements accurately describes a view?	A view is a virtual table based on the result of a SELECT query.	A view is a physical table that stores data permanently.	A view is a temporary table used for transactional purposes.	A view is a table used only for indexing purposes.	A
214	In PL/SQL, which trigger event is fired when a column value is updated to NULL?	AFTER UPDATE NULL	BEFORE UPDATE NULL	AFTER UPDATE OF column_name	BEFORE UPDATE OF column_name	C
215	In the context of a DBMS, what is the purpose of a stored procedure?	To store data in the database	To define the structure of a table	To encapsulate a series of SQL statements for reuse	To retrieve data from external sources	C
216	In the context of database transactions, what is a deadlock?	**A situation where two or more transactions are waiting for each other to release locks, preventing progress.**	A situation where a transaction reads uncommitted data.	A situation where a transaction violates integrity constraints.	A situation where a transaction is aborted due to a rollback request.	A
217	In the context of isolation levels in a DBMS, what does the "Read Uncommitted" isolation level allow?	Transactions can read uncommitted changes made by other transactions.	Transactions cannot read any data until all changes are committed.	Transactions can only read committed data.	Transactions can read their own uncommitted changes.	A
218	In the context of isolation levels in a DBMS, what does the "Read Uncommitted" isolation level allow?	**Transactions can read uncommitted changes made by other transactions.**	Transactions cannot read any data until all changes are committed.	Transactions can only read committed data.	Transactions can read their own uncommitted changes.	A
219	In the context of transactions, what does "serializability" mean?	A) The ability to serialize data	B) The ability to execute transactions concurrently	C) The ability to recover from failures	D) The ability to perform queries efficiently	A
220	OLD and NEW references are not available for table-level triggers.	TRUE	FALSE	Can be true or false	None of the above	A
221	and subprograms.	Yes	No	Can be yes or no	none of the above	A
222	PL/SQL controls the context area through a cursor.	TRUE	FALSE	Can be true or false	All of the above	A
223	the program	Try	Throw	Catch	Exception	D
224	Repeat sequence of statements; _________________ end repeat Fill in the correct option :	While Condition	Until variable	Until boolean expression	Until 0	C
225	Repeat sequence of statements; _________________ end repeat Fill in the correct option :	While Condition	Until variable	Until boolean expression	Until 0	C
226	Suppose a database system crashes again while recovering from a previous crash. Assume checkpointing is not done by the database either during the transactions or during recovery. Which of the following statements is/are correct?	The same undo and redo list will be used while recovering again.	The database will become inconsistent.	All the transactions that are already undone and redone will not be recovered again.	The system cannot recover any further.	A
227	Temporary stored procedures are stored in _________ database.	Master	Model	User specific	Tempdb	D
228	The __________ Statement is used for creating the package body.	CREATE	CREATE PACKAGE	CREATE BODY	CREATE PACKAGE BODY	D
229	The __________ Statement is used for creating the package body.	CREATE PACKAGE BODY	CREATE	CREAT BODY	CREATE BODY	A
230	The constructs of a procedure, function or a package are ________ .	Variables and Constants	Cursors	Exceptions	All of the above	D
231	The CREATE TRIGGER statement is used to create the trigger. THE _____ clause specifies the table name on which the trigger is to be attached. The ______ specifies that this is an AFTER INSERT trigger.	for insert, on	On, for insert	For, insert	None of the mentioned	B
232	specifies the table name on which the trigger is to be attached. The ______ specifies that this is an AFTER INSERT trigger.	for insert, on	On, for insert	For, insert	None of the mentioned	A
233	The format for compound statement is	Begin ……. end	Begin atomic……. end	Begin ……. repeat	Both Begin ……. end and Begin atomic……. end	D
234	The package specification is the interface to the package.	TRUE	FALSE	Nither TRUE NOR FALSE	none of the above	A

235	The parameters can be passed as default also to the procedures and the functions.	TRUE	FALSE	Nither TRUE NOR FALSE	None of he above	A
236	The property of a schedule that states that the result of executing concurrent transactions is the same as executing them serially is known as:	Consistency	Atomicity	Serializability	Durability	C
237	The technique used to detect and resolve conflicts among concurrent transactions is called:	Two-phase locking	Timestamp ordering	Deadlock detection	Deadlock prevention	C
238	Triggers can be defined on the?	table	view	schema	All of the above	D
239	Triggers can be defined on the?	DDL	DML	Database Operation	All of the above	D
240	What does "recoverability" encompass in the context of transactions and concurrency control?	A) The ability to recover from system failures	B) The ability to execute transactions concurrently	C) The ability to lock database tables	D) The ability to perform efficient queries	A
241	What does DBA stand for in the context of databases?	A) Database Backup Administrator	B) Data Business Analyst	C) Database Architect	D) Database Administrator	D
242	What does the concept of "recoverability" in concurrency control refer to?	A) The ability to lock data	B) The ability to recover from system failures	C) The ability to perform database recovery	D) The ability to execute queries efficiently	B
243	What does the concept of "serializability" in concurrency control refer to?	A) The ability to lock data	B) The ability to execute transactions in parallel	C) The ability to perform database recovery	D) The ability to execute queries efficiently	B
244	What does the term "serializability" imply in the context of transaction execution?	A) Transactions occur in sequence	B) Transactions can execute concurrently	C) Transactions are aborted	D) Transactions are isolated	A
245	What does the term "transaction isolation" refer to in the context of concurrency control?	A) A transaction's lifespan	B) A transaction's ability to update data	C) A transaction's isolation level	D) A transaction's recovery	C
246	What is a "bodiless" package in a DBMS context?	A) A package without a body	B) A package with excessive code	C) A package with only triggers	D) A package with minimal documentation	A
247	What is a "bodiless" package in a DBMS context?	A) A package without a body	B) A package with excessive code	C) A package with only triggers	D) A package with minimal documentation	A
248	What is a "transaction" in the context of a database management system (DBMS)?	A) A data dictionary	B) A single unit of work	C) A database schema	D) A database connection	B
249	What is a common approach to resolving deadlocks in a DBMS?	Rolling back one of the transactions involved in the deadlock	Killing all transactions to release locks	Preventing transactions from acquiring locks	Using deadlock detection and resolution algorithms	D
250	What is a common drawback of "pessimistic" locking in concurrency control systems?	A) Increased code modularity	B) Reduced data consistency	C) Reduced query performance	D) Optimized query execution	C
251	What is a common use of a database trigger in a DBMS?	A) To define package specifications	B) To encapsulate related procedures and functions	C) To monitor and respond to database events	D) To create database packages	C
252	What is a database schema?	A) A collection of tables in a database	B) A diagram representing the structure of a database	C) A set of rules that define the database structure	D) A description of the database structure, including tables, fields, and relationships	D
253	What is a package body in a DBMS?	A) A part of a package that contains package variables	B) A part of a package that specifies the package's procedures and functions	C) A part of a package that defines package triggers	D) A part of a package that implements the actual code for package procedures	D
254	What is a package specification in a DBMS?	A) A part of a package that contains package variables	B) A part of a package that specifies the package's procedures and functions	C) A part of a package that defines package triggers	D) A part of a package that implements the actual code for package procedures	D
255	What is a parameterized stored procedure in a DBMS?	**A procedure that accepts parameters and returns a result set**	A procedure that uses a parameter as its name	A procedure that cannot accept any input parameters	A procedure that can only accept integer parameters	A
256	What is a transaction in a DBMS?	A database schema	A series of SQL statements	**A logical unit of work that is either fully completed or fully undone**	A data dictionary	C

257	What is correct a PL/SQL program that create Trigger to update the "salary" of an employee to 80000 if the "department" is
	changed to 'Management'
		CREATE OR REPLACE TRIGGER trg_department_update
		BEFORE UPDATE OF department ON employee
		FOR EACH ROW
		BEGIN
		IF :NEW.department = 'Management' THEN
		:NEW.salary := 80000;
		END IF;
		END;
		/
			CREATE OR REPLACE TRIGGER trg_department_update
			BEFORE UPDATE OF department ON employee
			FOR EACH ROW
			IF :NEW.department = 'Management' THEN
			:NEW.salary := 80000;
			END IF;
			END;
			/
				CREATE OR REPLACE TRI
				trg_department_update
				BEFORE UPDATE OF department ON employee
				FOR EACH
				BEGIN
				IF :NEW.department = 'Management' THEN
				:NEW.salary := 80000;
				END IF;
				END;
				/
					CREATE OR REPLACE TRIGGER trg_department_update
					UPDATE OF department ON employee
					ROW
					BEGIN
					IF :NEW.department = 'Management' THEN
					:NEW.salary := 80000;
					END IF;
					END;
					/
						A

258	What is correct a procedure that calculates and displays the total salary of employees in a given department.
	The department name is an optional parameter with a default value of 'HR'.
		CREATE OR REPLACE PROCEDURE total_salary_by_department(p_department_name IN VARCHAR2 DEFAULT 'HR') AS
		v_total_salary NUMBER;
		BEGIN
		SELECT SUM(salary) INTO v_total_salary
		FROM employee
		WHERE department = p_department_name;
		
		DBMS_OUTPUT.PUT_LINE('Total Salary for Department ' || p_department_name || ': ' || v_total_salary);
		EXCEPTION
		WHEN OTHERS THEN
		DBMS_OUTPUT.PUT_LINE('An error occurred.');
		END;
		/
			CREATE OR REPLACE PROCEDURE total_salary_by_department(p_department_name IN VARCHAR2 ') AS
			v_total_salary NUMBER;
			BEGIN
			SELECT SUM(salary) INTO v_total_salary
			FROM employee
			WHERE department = p_department_name;
			
			DBMS_OUTPUT.PUT_LINE('Total Salary for Department ' || p_department_name || ': ' || v_total_salary);
			EXCEPTION
			WHEN OTHERS THEN
			DBMS_OUTPUT.PUT_LINE('An error occurred.');
			END;
			/
				CREATE OR REPLACE PROCEDURE total_salary_by_department(p_department_name IN VARCHAR2 DEFAULT 'HR') AS
				v_total_salary NUMBER;
				SELECT SUM(salary) INTO v_total_salary
				FROM employee
				WHERE department = p_department_name;
				
				DBMS_OUTPUT.PUT_LINE('Total Salary for Department ' || p_department_name || ': ' || v_total_salary);
				EXCEPTION
				WHEN OTHERS THEN
				DBMS_OUTPUT.PUT_LINE('An error occurred.');
				END;
				/
					CREATE OR REPLACE PROCEDURE total_salary_by_department(DEFAULT 'HR') AS
					v_total_salary NUMBER;
					BEGIN
					SELCT SUM(salary) INTO v_total_salary
					FROM employee
					WHERE department = p_department_name;
					
					DBMS_OUTPUT.PUT_LINE('Total Salary for Department ' || p_department_name || ': ' || v_total_salary);
					EXCEPTION
					WHEN OTHERS THEN
					DBMS_OUTPUT.PUT_LINE('An error occurred.');
					END;
					/
						A

259	What is one of the advantages of using procedures in a DBMS?	A) Increased code duplication	B) Slower query performance	C) Enhanced security vulnerabilities	D) Reduced code redundancy	D
260	What is one of the advantages of using triggers in a DBMS?	A) Increased code modularity	B) Reduced control over data changes	C) Enhanced query performance	D) Automated enforcement of data integrity rules	D

261	What is Output-- Insert sample records into the "employee" table
	INSERT INTO employee (employee_id, first_name, last_name, department, salary)
	VALUES (1, 'John', 'Doe', 'HR', 50000);
	
	INSERT INTO employee (employee_id, first_name, last_name, department, salary)
	VALUES (2, 'Jane', 'Smith', 'Finance', 60000);
	
	INSERT INTO employee (employee_id, first_name, last_name, department, salary)
	VALUES (3, 'Michael', 'Johnson', 'IT', 70000);
	
	INSERT INTO employee (employee_id, first_name, last_name, department, salary)
	VALUES (4, 'Merry', 'Agarwal', 'IT', 50000);
	
	CREATE OR REPLACE PROCEDURE delete_employee_by_id(p_employee_id NUMBER) AS
	BEGIN
	DELETE FROM employee
	WHERE employee_id = p_employee_id;
	
	IF SQL%ROWCOUNT > 0 THEN
	DBMS_OUTPUT.PUT_LINE('Employee deleted successfully.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('Employee ID not found. No employee deleted.');
	END IF;
	EXCEPTION
	WHEN OTHERS THEN
	DBMS_OUTPUT.PUT_LINE('An error occurred.');
	END;
	/
	EXECUTE delete_employee_by_id(6);
		An error occurred.	Employee ID not found. No employee deleted.	Employee deleted successfully.	EXECUTE delete_employee_by_id(2);	B

262	What is the ACID property that ensures that transactions are performed correctly and completely?	A) Atomicity	B) Consistency	C) Isolation	D) Durability	A
263	What is the key difference between stored procedures and functions in DBMS?	Stored procedures can return multiple values, while functions can only return a single value	Stored procedures can be executed by users, while functions can only be executed by the database administrator	Stored procedures are used for data manipulation, while functions are used for data retrieval	Stored procedures can be called from within other procedures, while functions cannot	A
264	What is the Oracle Error Code for ACCESS_INTO_NULL?	6592	6531	1722	6530	D

265	What is the output of the following program?
	DECLARE
	A NUMBER :=2;
	BEGIN
	FOR I IN 1..3 LOOP
	A := A*2;
	END LOOP;
	DBMS_OUTPUT.PUT_LINE(A);
	END;
		4	8	16	32	D

266	What is the primary advantage of using packages in a DBMS?	Improved query performance	Enhanced data security	Better code organization and reusability	Simplified database design	D

267	What is the primary difference between a row-level trigger and a statement-level trigger?	Row-level triggers are executed before statement-level triggers.	Row-level triggers are fired once for each affected row, while statement-level triggers are fired once for each SQL statement.	Statement-level triggers can be defined on tables, whereas row-level triggers cannot.	Row-level triggers can be recursive, while statement-level triggers cannot.	B
268	What is the primary drawback of "optimistic" concurrency control in a DBMS?	A) Increased code modularity	B) Slower query performance	C) Risk of transaction conflicts	D) Enhanced data integrity	C
269	What is the primary goal of concurrency control in a DBMS?	To maximize data redundancy	To minimize database access	To ensure data consistency and integrity in a multi-user environment	To eliminate the need for indexing	C
270	What is the primary goal of concurrency control in a DBMS?	To maximize data redundancy	To minimize database access	**To ensure data consistency and integrity in a multi-user environment**	To eliminate the need for indexing	C
271	What is the primary purpose of "deadlock detection" mechanisms in a DBMS that uses locking for concurrency control?	A) To prevent transaction conflicts	B) To optimize query performance	C) To eliminate transactions	D) To create database triggers	C
272	What is the primary purpose of a "BEFORE DELETE" trigger in a DBMS?	**To execute before any delete operation occurs**	To execute after a row is deleted from a table	To prevent any delete operations from taking place	To execute only if a delete operation fails	A
273	What is the primary purpose of a cursor in a stored procedure?	To store the results of a query	**To iterate through the records returned by a query**	To create a temporary table	To enforce data integrity constraints	B
274	What is the primary purpose of a database package in a DBMS?	A) To define database triggers	B) To encapsulate and group related procedures, functions, and variables	C) To establish database connections	D) To optimize query performance	B
275	What is the primary purpose of a stored procedure in DBMS?	To store and organize data in a database	To retrieve data from the database	To define the structure of the database	To encapsulate a series of database operations	D
276	What is the primary purpose of an "AFTER INSERT" trigger in a DBMS?	To execute before any insert operation occurs	**To execute after a new row is inserted into a table**	To prevent any insert operations from taking place	To execute only if an insert operation fails	B
277	What is the primary purpose of locking in concurrency control?	A) To eliminate transactions	B) To optimize query performance	C) To manage data access	D) To create database triggers	C
278	What is the primary purpose of transaction management in a database system?	A) To optimize query performance	B) To ensure data consistency	C) To define database triggers	D) To establish database connections	B
279	What is the primary purpose of using packages in a DBMS?	To store data in tables	**To organize related procedures, functions, and variables**	To manage user permissions	To enforce data integrity constraints	B
280	What is the primary syntax for creating a trigger in a DBMS?	A) CREATE PROCEDURE	B) CREATE FUNCTION	C) CREATE TRIGGER	D) CREATE TABLE	C
281	What is the purpose of "recoverability" in the context of database transactions?	A) To ensure all transactions recover	B) To prevent data recovery issues	C) To recover from system failures	D) To lock database tables	C
282	What is the purpose of declaring a cursor in SQL?	To define a new table in the database	To specify the database connection string	**To define a result set for query execution**	To create a new user account	C
283	What is the purpose of the "ROLLBACK" statement in a stored procedure?	To commit all changes made within the procedure	**To undo all changes made within the procedure**	To restart the execution of the procedure from the beginning	To create a new savepoint within the procedure	B

284	What is the purpose of the internal schema in a database system?	A) To define the logical view of the database for users	B) To specify the access controls and security settings for the database	C) To represent the physical storage structure of the database	D) To define the user views and queries for the database	C
285	What is the purpose of the JOIN operation in a relational database?	A) To add new records to a table	B) To remove records from a table	C) To combine data from multiple tables based on a related column	D) To modify existing records in a table	C
286	What is the purpose of the SAVEPOINT statement in DBMS?	To define the start of a transaction	To create a temporary table	To define a point within a transaction to which you can roll back	To release a lock on a database object	C
287	What is the role of a "transaction log" in a DBMS with respect to concurrency control?	A) To manage database locks	B) To record transaction history	C) To optimize query performance	D) To create database triggers	B
288	What is the role of a "transaction manager" in a DBMS?	A) To design the database	B) To manage database connections	C) To coordinate transaction execution	D) To create database triggers	C
289	What is the role of a package specification in a database package?	A) To define package variables	B) To provide information about the package's procedures and functions	C) To specify package triggers	D) To establish database connections	B
290	What is the role of the FETCH statement in SQL cursor operations?	It declares a new cursor.	**It retrieves rows from the result set and moves the cursor to the next row.**	It closes an open cursor.	It defines the structure of a cursor.	B
291	What is the role of the FETCH statement in SQL cursor operations?	It declares a new cursor.	**It retrieves rows from the result set and moves the cursor to the next row.**	It closes an open cursor.	It defines the structure of a cursor.	B
292	What is the syntax of User-defined exceptions?	DECLARE my-exception EXCEPTION;	DECLARE EXCEPTION;	DECLARE my-exception;	EXCEPTION;	A
293	What is the typical sequence of steps for developing a package in a DBMS?	A) Develop triggers first, then procedures	B) Develop procedures and package specification simultaneously	C) Write the package specification first, then the package body	D) Write the package body first, then the specification	C
294	What is the typical sequence of steps for developing a package in a DBMS?	A) Develop triggers first, then procedures	B) Develop procedures and package specification simultaneously	C) Write the package specification first, then the package body	D) Write the package body first, then the specification	C
295	What part of a procedure in a DBMS is responsible for declaring the input and output variables?	A) Procedure header	B) Procedure specification	C) Procedure body	D) Procedure parameters	B
296	What type of trigger is executed automatically after the triggering event?	A) After Trigger	B) Before Trigger	C) Instead of Trigger	D) Compound Trigger	A
297	What type of trigger is executed automatically before a specific event, such as an INSERT or UPDATE operation?	A) After Trigger	B) Before Trigger	C) Instead of Trigger	D) Compound Trigger	B
298	When executing a stored procedure, what keyword is commonly used to return a result set to the calling application?	RESULT	RESULTSET	**OUTPUT**	RETURN	C
299	and the data doesn't get ___ back, another transaction tries to access the updated database item.	Rolled	Committed	Aborted	None	A
300	Which ACID property ensures that a transaction's effects on the database are permanent?	A) Atomicity	B) Consistency	C) Isolation	D) Durability	D
301	Which attribute is used to raise exception?	Open	Select	Raise	Try	C
302	Which attribute returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows?	%NOTFOUND	%ISOPEN	%ROWCOUNT	%FOUND	D
303	Which clause is used to create trigger on a view?	BEFORE	INSTEAD OF	AFTER	None of the above	B
304	Which component of a package in DBMS defines the interface and public entities?	Package body	Package Signature	Package Constructor	Package specification	D
305	Which concurrency control technique allows multiple transactions to read data simultaneously but enforces write locks to prevent data conflicts?	Two-Phase Locking (2PL)	Time-stamp Ordering	Multi-Version Concurrency Control (MVCC)	Optimistic Concurrency Control	A
306	Which concurrency control technique allows multiple transactions to read data simultaneously but enforces write locks to prevent data conflicts?	**Two-Phase Locking (2PL)**	Time-stamp Ordering	Multi-Version Concurrency Control (MVCC)	Optimistic Concurrency Control	A
307	Which cursor attribute can be used to determine the total number of rows returned by a cursor in PL/SQL?	%ROWCOUNT	%FOUND	%ISOPEN	%NOTFOUND	A

308	Which database system component is responsible for managing transactions and ensuring data integrity?	A) Database schema	B) Data dictionary	C) Concurrency control manager	D) Query optimizer	C
309	Which isolation level allows only committed data to be read?	Read Uncommitted	Read committed	Serializable	Read Update	B
310	Which isolation level allows the highest concurrency but may result in non-repeatable reads?	Read Uncommitted	Read Committed	Repeatable Read	Serializable	A
311	Which isolation level provides the highest level of data consistency but can lead to reduced concurrency?	Read Uncommitted	Read Committed	Repeatable Read	Serializable	D
312	Which isolation level provides the highest level of data consistency but can lead to reduced concurrency?	Read Uncommitted	Read Committed	Repeatable Read	**Serializable**	D
313	Which keyword is used to create a new package body in PL/SQL?	BODY	IMPLEMENTATION	DEFINE	CREATE	A
314	Which keyword is used to define an exception handler in PL/SQL?	EXCEPTION	CATCH	TRY	HANDLE	A
315	Which of the following actions can be performed by a "BEFORE INSERT" trigger in a DBMS?	Rollback the entire transaction	**Modify data before it is inserted**	Create a new table	Terminate the DBMS session	B
316	Which of the following are benefits of Triggers?	Generating some derived column values automatically	Enforcing referential integrity	Event logging and storing information on table access	All of the above	D
317	Which of the following are the advantages of PL/SQL Packages?	Modularity	Easier Application Design	Information Hiding	All of the above	D
318	Which of the following clause does not comes in the syntax while raising an exception?	DECLARE	WHEN	CLOSE	END	C
319	Which of the following database languages is used to define the structure and organization of a database?	A) Data Manipulation Language (DML)	B) Data Definition Language (DDL)	C) Data Control Language (DCL)	D) Data Query Language (DQL)	B
320	Which of the following describes the ACID properties of transactions?	Atomicity, Consistency, Isolation, Durability	Atomicity, Consistency, Isolation, Dileang	Atomicity, Consistency, Isolate, Durability	Atomic, Consistency, Isolation, Durability	A
321	Which of the following is a benefit of using procedures in a DBMS?	A) Increased code duplication	B) Reduced security	C) Improved code organization and maintenance	D) Limited code reuse	C
322	Which of the following is a benefit of using the "SAVEPOINT" statement in a DBMS?	It allows you to commit a transaction.	It allows you to roll back the entire database.	It enables you to create nested transactions.	**It provides a way to roll back to a specific point within a transaction.**	D
323	Which of the following is a benefit of using the "SAVEPOINT" statement in a DBMS?	It allows you to commit a transaction.	It allows you to roll back the entire database.	It enables you to create nested transactions.	**It provides a way to roll back to a specific point within a transaction.**	D
324	Which of the following is a characteristic of an "AUTOCOMMIT" transaction mode in a DBMS?	Each SQL statement is treated as a separate transaction.	**Transactions are automatically committed after each SQL statement.**	Transactions cannot be rolled back.	It is a read-only mode.	B
325	Which of the following is a commonly used technique for concurrency control in a DBMS?	A) Optimistic concurrency control	B) Serializability control	C) Slower query performance	D) Increased code duplication	B
326	Which of the following is a drawback of strict two-phase locking (S2PL)	Increased concurrency	Increased deadlock probability	Reduced consistency	Reduced durability	B
327	Which of the following is a property of a transaction in a database system?	A) Increased code duplication	B) Slower query performance	C) Atomicity	D) Reduced code redundancy	C
328	Which of the following is an advantage of using packages in a DBMS?	A) Limited code organization	B) Increased code redundancy	C) Enhanced code isolation	D) Improved code modularity and reusability	D
329	Which of the following is an advantage of using procedures in a DBMS?	A) Increased code redundancy	B) Slower execution of queries	C) Improved security vulnerabilities	D) Code reusability and maintainability	D
330	Which of the following is an advantage of using stored procedures in a DBMS?	Reduced security	Increased data redundancy	**Improved performance**	Decreased data integrity	C
331	Which of the following is an advantage of using triggers in a DBMS?	A) Limited code organization	B) Increased code redundancy	C) Enhanced code isolation	D) Improved code modularity and reusability	D
332	Which of the following is an example of a parameterized trigger in a DBMS?	A trigger that fires after any insert operation	A trigger that fires after a specific date and time	A trigger that fires when a specific condition is met	**A trigger that accepts parameters passed from the calling application**	D
333	Which of the following is an example of an optimistic concurrency control technique in a DBMS?	Two-Phase Locking (2PL)	**Multi-Version Concurrency Control (MVCC)**	Time-stamp Ordering	Rollback Segments	B
334	Which of the following is not a level of data abstraction in a database system?	A) Physical level	B) Logical level	C) External level	D) Semantic level	D
335	Which of the following is NOT a property of a transaction in DBMS?	Atomicity	Consistency	Durability	Isolation	B
336	Which of the following is not a type of trigger in DBMS?	Insert trigger	Update trigger	Delete trigger	Search trigger	D

337	Which of the following is not an advantage of trigger?	Various column values are automatically generated by triggers	Maintains the integrity of referential	Tables are replicated asynchronously	Validating transactions and preventing them from being invalid	C
338	Which of the following is NOT an Oracle-supported trigger?	BEFORE	DURING	AFTER	INSTEAD OF	B
339	Which of the following is the correct format for if statement?	If boolean expression then statement or compound statement elseif boolean expression then statement or compound statement else statement or compound statement end if	If boolean expression then statement or compound statement elsif boolean expression then statement or compound statement else statement or compound statement end if	If boolean expression then statement or compound statement elif boolean expression then statement or compound statement else statement or compound statement end if	If boolean expression then statement or compound statement else statement or compound statement else statement or compound statement end if	A
340	Which of the following is true about compound triggers?	They can only be defined for tables, not views	They are fired once for each row affected by the triggering event	They cannot contain any SQL statements	They are not supported in DBMS	B
341	Which of the following is true about recursive triggers?	They are triggered by other triggers	They can only be fired once per event	They can cause an infinite loop if not handled properly	They are not supported in DBMS	C
342	Which of the following is true about stored procedures?	They can only return a single scalar value.	They can contain control-of-flow statements like IF and LOOP.	They cannot accept input parameters.	They are always automatically executed when the database starts.	B
343	Which of the following is TRUE about User-defined exceptions?	Users can explicitly raise an exception by using a RAISE statement	RAISE_APPLICATION_ERROR can be used to raise a user-defined exception explicitly	both 1 and 2	None of the above	C
344	Which of the following is used to input the entry and give the result in a variable in a procedure?	Put and get	Get and put	Out and In	In and out	D
345	Which of the following makes the transaction permanent in the database?	View	Commit	Rollback	Flashback	B
346	Which of the following specifies when the trigger will be executed?	BEFORE	AFTER	INSTEAD OF	All of the above	D
347	Which of the following statements best defines database recovery in DBMS?	The process of restoring data from backup tapes	The process of ensuring that the database remains secure	The process of restoring the database to a consistent state after a failure	The process of recovering deleted data from the Recycle Bin	C
348	Which of the following statements is true about First Normal Form (1NF)?	A) It allows for multivalued dependencies.	B) It allows for partial dependencies.	C) It eliminates repeating groups and ensures atomicity of data.	D) It enforces referential integrity constraints.	C
349	Which of the following statements is true about stored procedures?	Stored procedures cannot have input parameters	Stored procedures cannot return values	Stored procedures can be reused and shared by multiple applications	Stored procedures can only be executed by the database administrator	C
350	Which of the following statements is true about the Two-tier architecture?	A) It allows for better scalability than the Three-tier architecture.	B) It is easier to maintain and modify compared to the Three-tier architecture.	C) It requires less network traffic than the Three-tier architecture.	D) It provides better security and data isolation compared to the Three-tier architecture.	C
351	Which of the following statements is true regarding stored procedures?	Stored procedures always return a single value.	Stored procedures are not allowed to contain conditional statements.	**Stored procedures are precompiled and stored in the database for reuse.**	Stored procedures can only be executed by database administrators.	C
352	Which package lets PL/SQL programs read and write operating system (OS) text files?	UTL_HTTP	UTL_FILE	UTL_SMTP	UTL_FMT	B
353	Which part of a procedure in a DBMS is responsible for specifying the operations to be performed?	A) Procedure header	B) Procedure specification	C) Procedure body	D) Procedure parameters	C

354	Which property of a transaction ensures that it does not interfere with other transactions while executing?	A) Atomicity	B) Consistency	C) Isolation	D) Durability	C
355	Which property of a transaction ensures that it does not interfere with other transactions while executing?	A) Atomicity	B) Consistency	C) Isolation	D) Durability	C
356	Which property of a transaction ensures that it does not violate integrity constraints?	Isolation	Atomicity	Consistency	Durability	C
357	Which property of a transaction ensures that it either completes in its entirety or has no effect at all?	A) Atomicity	B) Optimistic concurrency control	C) Slower query performance	D) Data redundancy	A
358	Which property of a transaction ensures that the database remains in a consistent state after transaction execution?	A) Atomicity	B) Consistency	C) Isolation	D) Durability	B
359	Which recovery technique uses backward recovery to undo the changes made by a failed transaction?	Undo logging	Redo logging	Deferred update	Immediate update	A
360	Which specifies the column name that will be updated?	For col_name	ON col_name	OF col_name	WHEN col_name	C
361	Which type of database constraint ensures that a foreign key value matches a primary key value in another table?	A) Unique constraint	B) Primary key constraint	C) Foreign key constraint	D) Not null constraint	C
362	Which type of database trigger in SQL is executed before the triggering event occurs?	AFTER trigger	INSTEAD OF trigger	BEFORE trigger	FOR EACH ROW trigger	C
363	Which type of error occurs when the database crashes while a transaction is being executed?	System error	Media error	Transaction error	Operator error	A
364	Which type of trigger in a DBMS can be used to prevent changes to a table?	BEFORE trigger	AFTER trigger	**INSTEAD OF trigger**	FOR EACH ROW trigger	C
365	Which type of trigger in a DBMS is fired after a triggering event and can be used for auditing purposes?	BEFORE trigger	**AFTER trigger**	INSTEAD OF trigger	FOR EACH ROW trigger	B
366	Which type of view in PL/SQL allows you to update data directly through the view?	Materialized View	Read-Only View	Updatable View	Join View	C
367	Why is "concurrency control" important in a multi-user database environment?	A) To increase query performance	B) To ensure data consistency	C) To eliminate transactions	D) To optimize database storage	B
368	Why is "concurrency" a concern in a multi-user DBMS environment?	A) To simplify data retrieval	B) To ensure data consistency	C) To reduce query performance	D) To create redundant data	B
369	Why is concurrency control needed in a database management system (DBMS)?	A) To increase data redundancy	B) To slow down query execution	C) To ensure data consistency	D) To reduce code duplication	C

370	Write a PL/SQL function named `get_student_average_grade` that takes a student ID as input and returns the average grade of the specified student across all subjects. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_average_grade(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_average_grade(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_average_grade(student_id NUMBER, avg_grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_average_grade(student_id NUMBER) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

371	Write a PL/SQL function named `get_student_count_by_subject` that takes a subject as input and returns the count of students enrolled in that subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_count_by_subject(subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE FUNCTION get_student_count_by_subject(subject VARCHAR2) RETURN NUMBER IS
			student_count NUMBER;
			BEGIN
			  -- Function logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_count_by_subject(subject VARCHAR2, student_count OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE PROCEDURE get_student_count_by_subject(subject VARCHAR2) IS
					BEGIN
					  -- Procedure logic here
					END;
						a

372	Write a PL/SQL function named `get_student_grade_in_subject` that takes a student ID and a subject as input and returns the grade of the specified student for the given subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2, grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

373	Write a PL/SQL function named `get_student_grade_in_subject` that takes a student ID and a subject as input and returns the grade of the specified student for the given subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2, grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

374	Write a PL/SQL function named `get_student_grade_in_subject` that takes a student ID and a subject as input and returns the grade of the specified student for the given subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2, grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

375	Write a PL/SQL function named `get_student_grade_in_subject` that takes a student ID and a subject as input and returns the grade of the specified student for the given subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2, grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

376	Write a PL/SQL function named `get_student_grade_in_subject` that takes a student ID and a subject as input and returns the grade of the specified student for the given subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2, grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_grade_in_subject(student_id NUMBER, subject VARCHAR2) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

377	Write a PL/SQL function named `get_student_grade` that takes a student ID and a subject as input and returns the grade of the specified student for the given subject. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_grade(student_id NUMBER, subject VARCHAR2) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_grade(student_id NUMBER, subject VARCHAR2) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_grade(student_id NUMBER, subject VARCHAR2, grade OUT NUMBER) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_student_grade(student_id NUMBER, subject VARCHAR2) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

378	Write a PL/SQL function named `get_student_info` that takes a student ID as input and returns the student's name, age, and the number of subjects they are enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_info(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_info(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_info(
				student_id NUMBER,
				student_name OUT VARCHAR2,
				student_age OUT NUMBER,
				subject_count OUT NUMBER
				) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION
					get_student_info(student_id NUMBER) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						C

379	Write a PL/SQL function named `get_student_info` that takes a student ID as input and returns the student's name, age, and the number of subjects they are enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_info(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_info(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_info(
				student_id NUMBER,
				student_name OUT VARCHAR2,
				student_age OUT NUMBER,
				subject_count OUT NUMBER
				) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION
					get_student_info(student_id NUMBER) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						C

380	Write a PL/SQL function named `get_student_info` that takes a student ID as input and returns the student's name, age, and the number of subjects they are enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_info(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_info(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_info(
				student_id NUMBER,
				student_name OUT VARCHAR2,
				student_age OUT NUMBER,
				subject_count OUT NUMBER
				) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION
					get_student_info(student_id NUMBER) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						C

381	Write a PL/SQL function named `get_student_info` that takes a student ID as input and returns the student's name, age, and the number of subjects they are enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_info(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_info(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_info(
				student_id NUMBER,
				student_name OUT VARCHAR2,
				student_age OUT NUMBER,
				subject_count OUT NUMBER
				) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION
					get_student_info(student_id NUMBER) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						C

382	Write a PL/SQL function named `get_student_info` that takes a student ID as input and returns the student's name, age, and the number of subjects they are enrolled in. Which of the following code snippets correctly
	defines this function?
		CREATE OR REPLACE FUNCTION get_student_info(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		-- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_info(student_id NUMBER) IS
			BEGIN
			-- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_info(
				student_id NUMBER,
				student_name OUT VARCHAR2,
				student_age OUT NUMBER,
				subject_count OUT NUMBER
				) IS
				BEGIN
				-- Function logic here
				END;
					CREATE OR REPLACE FUNCTION
					get_student_info(student_id NUMBER) RETURN TABLE IS
					BEGIN
					-- Function logic here
					END;
						C

383	Write a PL/SQL function named `get_student_subject_scores` that takes a student ID as input and returns a cursor containing the subject and grade for all subjects in which the student is enrolled. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_subject_scores(student_id NUMBER) RETURN CURSOR IS
		BEGIN
		-- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_student_subject_scores(student_id NUMBER) IS
			BEGIN
			-- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_student_subject_scores(
				student_id NUMBER,
				subject_scores OUT SYS_REFCURSOR
				) IS
				BEGIN
				-- Function logic here
				END;
					CREATE OR REPLACE FUNCTION
					get_student_subject_scores(student_id NUMBER) RETURN TABLE IS
					BEGIN
					-- Function logic here
					END;
						C

384	Write a PL/SQL function named `get_student_subjects` that takes a student ID as input and returns a list of subjects that the student is enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		-- Function logic here
		END;
			CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
			BEGIN
			-- Function logic here
			END;
				CREATE OR REPLACE PROCEDURE get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
				BEGIN
				-- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE
					get_student_subjects(student_id NUMBER) IS
					BEGIN
					-- Procedure logic here
					END;
						A

385	Write a PL/SQL function named `get_student_subjects` that takes a student ID as input and returns a list of subjects that the student is enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		-- Function logic here
		END;
			CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
			BEGIN
			-- Function logic here
			END;
				CREATE OR REPLACE PROCEDURE get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
				BEGIN
				-- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE
					get_student_subjects(student_id NUMBER) IS
					BEGIN
					-- Procedure logic here
					END;
						A

386	Write a PL/SQL function named `get_student_subjects` that takes a student ID as input and returns a list of subjects that the student is enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		-- Function logic here
		END;
			CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
			BEGIN
			-- Function logic here
			END;
				CREATE OR REPLACE PROCEDURE get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
				BEGIN
				-- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE
					get_student_subjects(student_id NUMBER) IS
					BEGIN
					-- Procedure logic here
					END;
						A

387	Write a PL/SQL function named `get_student_subjects` that takes a student ID as input and returns a list of subjects that the student is enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		-- Function logic here
		END;
			CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
			BEGIN
			-- Function logic here
			END;
				CREATE OR REPLACE PROCEDURE get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
				BEGIN
				-- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE
					get_student_subjects(student_id NUMBER) IS
					BEGIN
					-- Procedure logic here
					END;
						A

388	Write a PL/SQL function named `get_student_subjects` that takes a student ID as input and returns a list of subjects that the student is enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE FUNCTION get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
			BEGIN
			  -- Function logic here
			END;
				CREATE OR REPLACE PROCEDURE get_student_subjects(student_id NUMBER, subjects OUT VARCHAR2) IS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE get_student_subjects(student_id NUMBER) IS
					BEGIN
					  -- Procedure logic here
					END;
						A

389	Write a PL/SQL function named `get_subjects_by_student` that takes a student ID as input and returns a list of subjects that the student is enrolled in. Which of the following code snippets correctly defines this function?	CREATE OR REPLACE FUNCTION get_subjects_by_student(student_id NUMBER) RETURN VARCHAR2 IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE get_subjects_by_student(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION get_subjects_by_student(student_id NUMBER, subjects OUT VARCHAR2) IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION get_subjects_by_student(student_id NUMBER) RETURN TABLE IS
					BEGIN
					  -- Function logic here
					END;
						A

390	Write a PL/SQL procedure named `add_student_subject` that takes a student ID, subject, and grade as input and adds a new subject enrollment record for the specified student in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) IS -- Declare variables here BEGIN -- Procedure logic here END;	B
391	Write a PL/SQL procedure named `add_student_subject` that takes a student ID, subject, and grade as input and adds a new subject enrollment record for the specified student in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE add_student_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) IS -- Declare variables here BEGIN -- Procedure logic here END;	B

392	Write a PL/SQL procedure named `calculate_avg_grade` that takes a student ID as input and calculates the average grade of that student across all subjects. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE PROCEDURE calculate_avg_grade(student_id NUMBER) IS
		BEGIN
		  -- Procedure logic here
		END;
			CREATE OR REPLACE PROCEDURE calculate_avg_grade(student_id NUMBER, avg_grade OUT NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE calculate_avg_grade(student_id NUMBER) AS
				avg_grade NUMBER;
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE FUNCTION calculate_avg_grade(student_id NUMBER) RETURN NUMBER IS
					BEGIN
					  -- Function logic here
					END;
						B

393	Write a PL/SQL procedure named `calculate_student_average` that calculates the average grade for a specific student identified by their `student_id` and stores it in a variable. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE PROCEDURE calculate_student_average(student_id NUMBER) AS
		BEGIN
		  -- Procedure logic here
		END;
			CREATE OR REPLACE PROCEDURE calculate_student_average(student_id NUMBER, avg_grade OUT NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE FUNCTION calculate_student_average(student_id NUMBER) RETURN NUMBER IS
				BEGIN
				  -- Function logic here
				END;
					CREATE OR REPLACE FUNCTION calculate_student_average(student_id NUMBER, avg_grade OUT NUMBER) RETURN NUMBER IS
					BEGIN
					  -- Function logic here
					END;
						B

394	Write a PL/SQL procedure named `delete_student_record` that takes a student ID as input and deletes the corresponding student record from the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION delete_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

395	Write a PL/SQL procedure named `delete_student_record` that takes a student ID as input and deletes the corresponding student record from the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION delete_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

396	Write a PL/SQL procedure named `delete_student_record` that takes a student ID as input and deletes the corresponding student record from the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION delete_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

397	Write a PL/SQL procedure named `delete_student_record` that takes a student ID as input and deletes the corresponding student record from the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION delete_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

398	Write a PL/SQL procedure named `delete_student_record` that takes a student ID as input and deletes the corresponding student record from the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION delete_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE delete_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

399	Write a PL/SQL procedure named `delete_student` that takes a student ID as input and deletes the corresponding student record from the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION delete_student(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE delete_student(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE delete_student(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE delete_student(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

400	Write a PL/SQL procedure named `enroll_student_in_subject` that takes a student ID, subject, and grade as input and inserts a new enrollment record for the specified student in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION enroll_student_in_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE enroll_student_in_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE enroll_student_in_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE enroll_student_in_subject(student_id NUMBER, subject VARCHAR2, grade NUMBER) IS -- Declare variables here BEGIN -- Procedure logic here END;	B

401	Write a PL/SQL procedure named `enroll_student` that takes student details (name, age, subject, grade) as input and inserts a new record into the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION enroll_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE enroll_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE enroll_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE enroll_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) IS -- Declare variables here BEGIN -- Procedure logic here END;	b
402	Write a PL/SQL procedure named `insert_student_record` that takes student details (name, age, subject, grade) as input and inserts a new record into the "student" table. Additionally, it should return the newly generated student ID. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER, student_id OUT NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER, student_id OUT NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE FUNCTION insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS student_id NUMBER; BEGIN -- Function logic here END;	B
403	Write a PL/SQL procedure named `insert_student_record` that takes student details (name, age, subject, grade) as input and inserts a new record into the "student" table. Additionally, it should return the newly generated student ID. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER, student_id OUT NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER, student_id OUT NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE FUNCTION insert_student_record(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS student_id NUMBER; BEGIN -- Function logic here END;	B
404	Write a PL/SQL procedure named `insert_student` that takes student details (name, age, subject, grade) as input and inserts a new record into the "student" table. Additionally, it should return the newly generated student ID. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION insert_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE insert_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER, student_id OUT NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE insert_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER, student_id OUT NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE FUNCTION insert_student(student_name VARCHAR2, student_age NUMBER, subject VARCHAR2, student_grade NUMBER) RETURN NUMBER IS student_id NUMBER; BEGIN -- Function logic here END;	B

405	Write a PL/SQL procedure named `update_student_grade` that takes a student ID, subject, and a new grade as input and updates the grade of the specified student for the given subject. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION update_student_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE update_student_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE update_student_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE update_student_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) IS -- Declare variables here BEGIN -- Procedure logic here END;	B

406	Write a PL/SQL procedure named `update_student_record` that takes a student ID as input and updates the student's name, age, and grade in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION update_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

407	Write a PL/SQL procedure named `update_student_record` that takes a student ID as input and updates the student's name, age, and grade in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION update_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

408	Write a PL/SQL procedure named `update_student_record` that takes a student ID as input and updates the student's name, age, and grade in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION update_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

409	Write a PL/SQL procedure named `update_student_record` that takes a student ID as input and updates the student's name, age, and grade in the "student" table. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION update_student_record(student_id NUMBER) RETURN NUMBER IS
		BEGIN
		  -- Function logic here
		END;
			CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
			BEGIN
			  -- Procedure logic here
			END;
				CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) AS
				BEGIN
				  -- Procedure logic here
				END;
					CREATE OR REPLACE PROCEDURE update_student_record(student_id NUMBER) IS
					  -- Declare variables here
					BEGIN
					  -- Procedure logic here
					END;
						B

410	Write a PL/SQL procedure named `update_student_subject_grade` that takes a student ID, subject, and a new grade as input and updates the grade of the specified student for the given subject. Which of the following code snippets correctly defines this procedure?	CREATE OR REPLACE FUNCTION update_student_subject_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) RETURN NUMBER IS BEGIN -- Function logic here END;	CREATE OR REPLACE PROCEDURE update_student_subject_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) IS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE update_student_subject_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) AS BEGIN -- Procedure logic here END;	CREATE OR REPLACE PROCEDURE update_student_subject_grade(student_id NUMBER, subject VARCHAR2, new_grade NUMBER) IS -- Declare variables here BEGIN -- Procedure logic here END;	B
411	You have a stored procedure named UpdateEmployeeSalary that accepts an employee ID and a salary value as parameters and updates the employee's salary in the database. Which SQL statement would you use to execute this stored procedure with employee ID 101 and a new salary of 55000?	**EXEC UpdateEmployeeSalary 101, 55000;**	CALL UpdateEmployeeSalary(101, 55000);	RUN UpdateEmployeeSalary(101, 55000);	UPDATE EmployeeSalary(101, 55000);	A

412	You have a stored procedure that calculates the average salary of employees in a specific department. Which SQL statement do you use to execute this stored procedure and retrieve the result?	**EXEC GetAverageSalaryForDepartment 101;**	CALL GetAverageSalaryForDepartment(101);	EXEC GetAverageSalaryForDepartment @DepartmentID = 101;	EXEC GetAverageSalaryForDepartment @DepartmentID = 101, @Result = OUTPUT;	A
413	want to create a trigger that updates the "LastPurchaseDate" column to the current date whenever a new purchase is made by a customer. What type of trigger should you create?	**AFTER INSERT**	BEFORE INSERT	AFTER UPDATE	INSTEAD OF INSERT	A
414	You have a table named "Inventory" with a "Quantity" column. You want to create a trigger that automatically updates the "Quantity" column to zero when a product is marked as "Out of Stock." What type of trigger should you create?	**AFTER UPDATE**	BEFORE UPDATE	AFTER INSERT	INSTEAD OF UPDATE	A

415	You want to create a stored procedure that inserts a new customer record into the "Customers" table. The customer's name, email, and phone number will be passed as parameters. Which SQL statement creates this stored procedure?	**CREATE PROCEDURE InsertCustomer**
		@Name VARCHAR(50),
		@Email VARCHAR(100),
		@Phone VARCHAR(20)
		AS
		BEGIN
		INSERT INTO Customers (Name, Email, Phone)
		VALUES (@Name, @Email, @Phone)
		END
			CREATE PROCEDURE InsertCustomer AS BEGIN INSERT INTO Customers (Name, Email, Phone) VALUES (@Name, @Email, @Phone) END	CREATE PROCEDURE InsertCustomer @CustomerData VARCHAR(MAX) AS BEGIN INSERT INTO Customers (Name, Email, Phone) VALUES (@CustomerData) END	CREATE PROCEDURE InsertCustomer @Name VARCHAR(50), @Email VARCHAR(100), @Phone VARCHAR(20) AS BEGIN INSERT INTO Customers (Name, Email, Phone) VALUES (@Name, @Email, @Phone) END	A