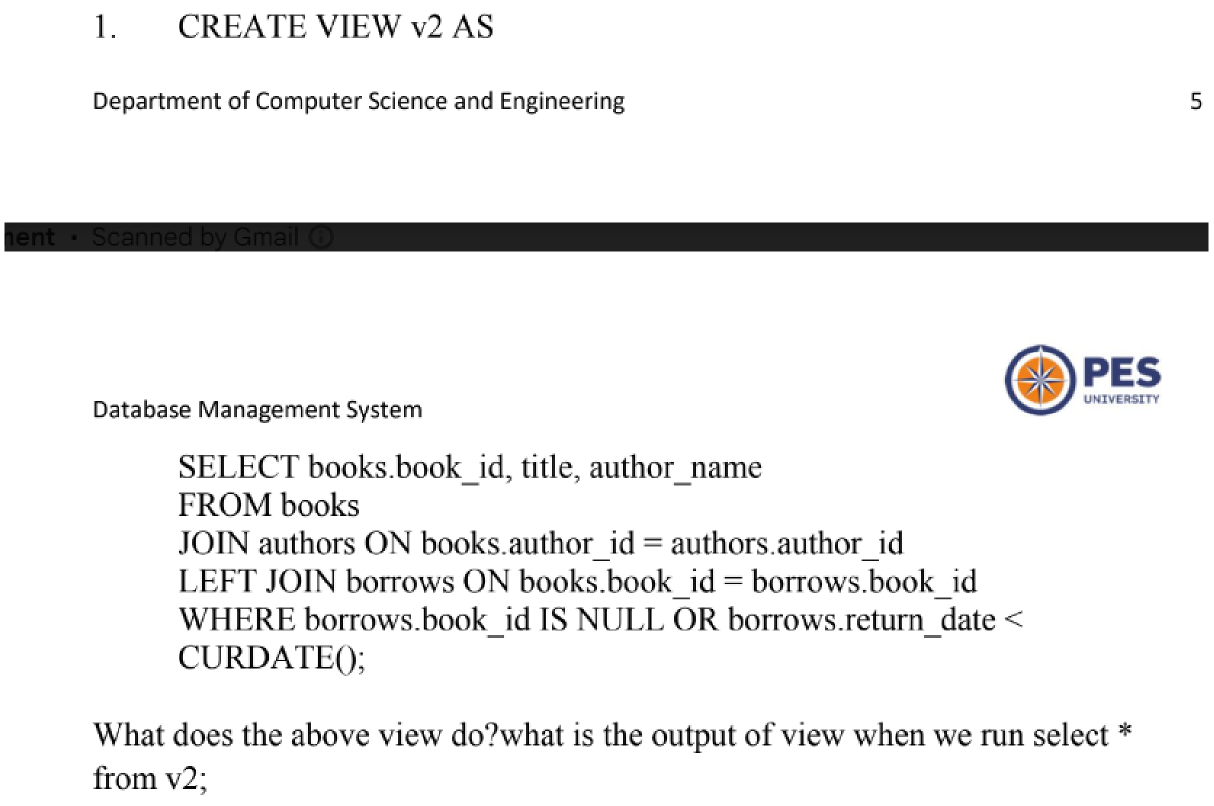
Lab5
DBMS
Procedures Functions and views

Art Gallery Management (AGM)

PRAMATH S

Task1



1. CREATE VIEW v2 AS

Department of Computer Science and Engineering 5

Database Management System

```
SELECT books.book_id, title, author_name
FROM books
JOIN authors ON books.author_id = authors.author_id
LEFT JOIN borrows ON books.book_id = borrows.book_id
WHERE borrows.book_id IS NULL OR borrows.return_date <
CURDATE();
```

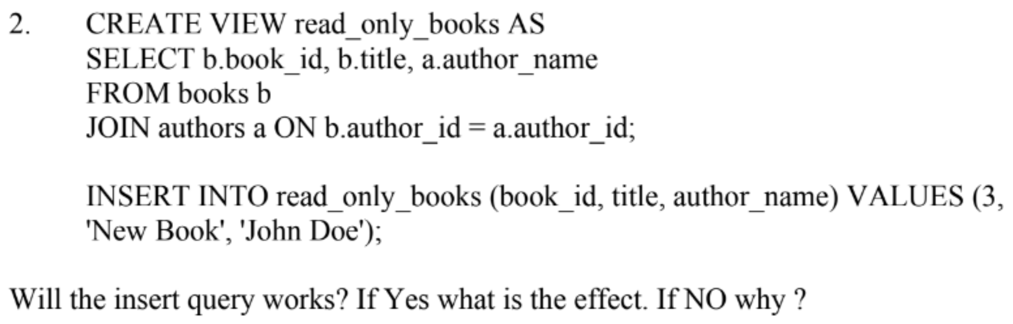
What does the above view do?what is the output of view when we run select * from v2;

The resulting view, when queried, will give a list of books book_id, title, and author_name that have either never been borrowed or were borrowed but are overdue.

The book_id, title, and author_name are displayed for each book in the result.

Book_id	title	Author_name
1	Harry Potter and the Sorcerer's Stone	J.K. Rowling
2	1984	George Orwell
4	Pride and Prejudice	J.K. Rowling
5	IT	George Orwell

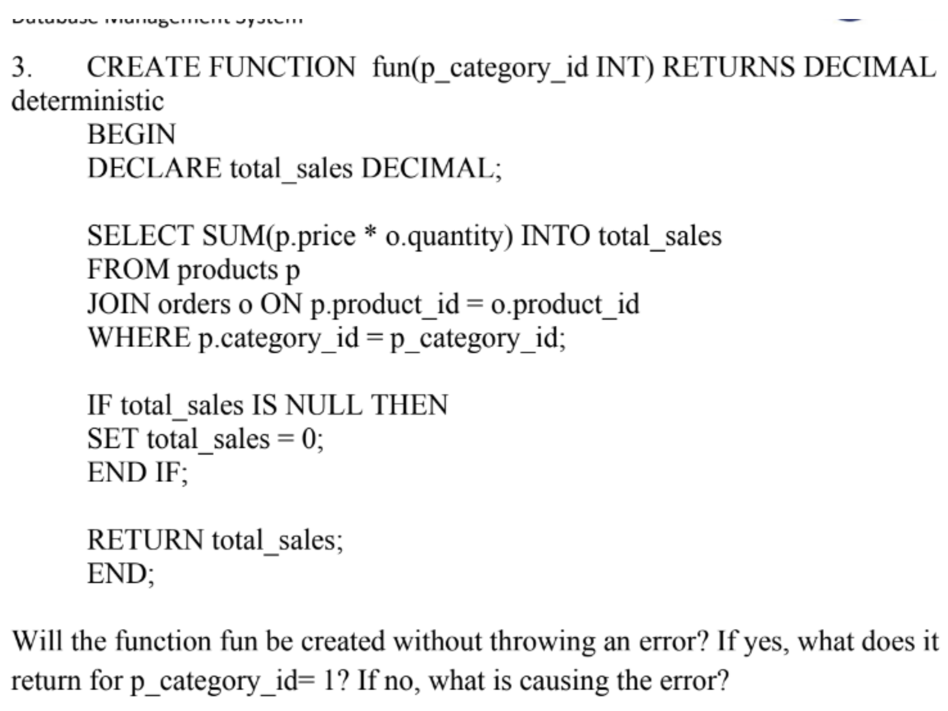
2)



2.
```
CREATE VIEW read_only_books AS
SELECT b.book_id, b.title, a.author_name
FROM books b
JOIN authors a ON b.author_id = a.author_id;

INSERT INTO read_only_books (book_id, title, author_name) VALUES (3,
'New Book', 'John Doe');
```

Will the insert query works? If Yes what is the effect. If NO why ?

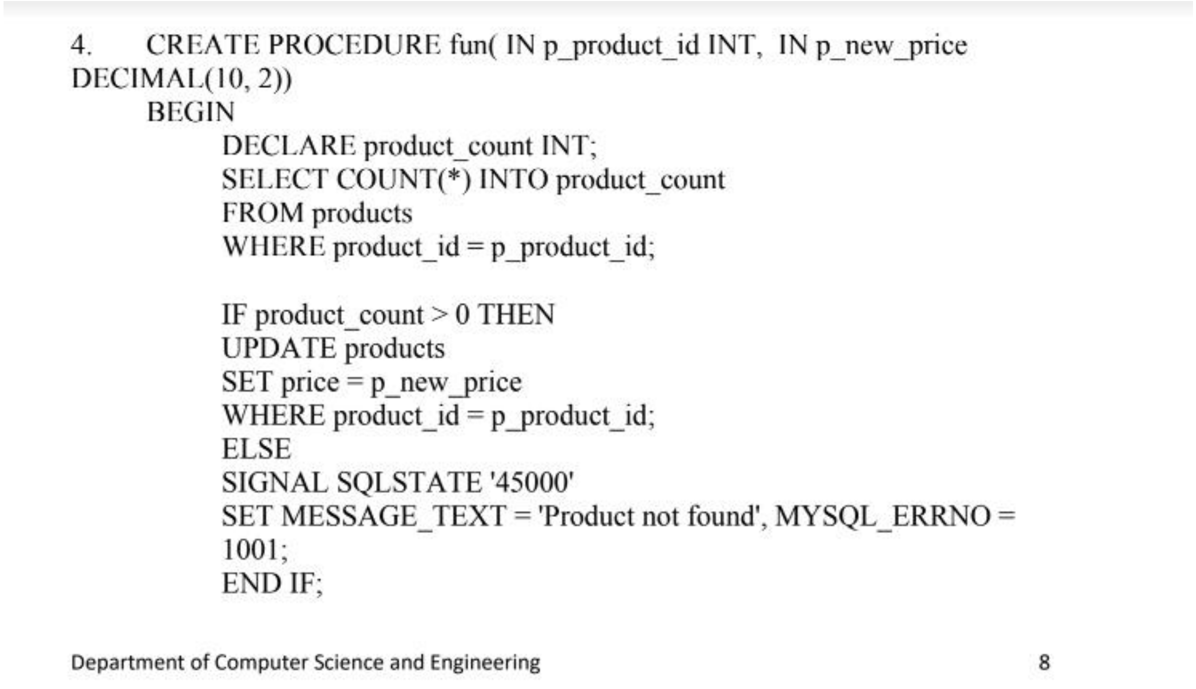
Here insert query won't work generally view is based on the multiple table joins its not updatable because there's ambiguity in determining how to propagate the changes to the underlying tables.



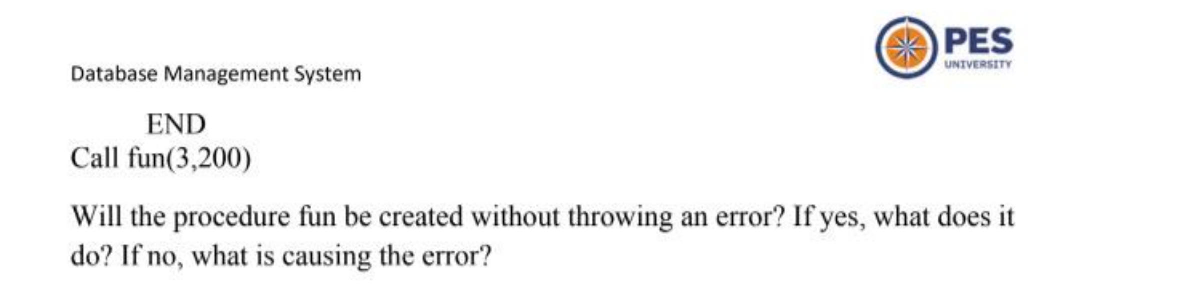
3.
```
CREATE FUNCTION  fun(p_category_id INT) RETURNS DECIMAL
deterministic
        BEGIN
        DECLARE total_sales DECIMAL;

        SELECT SUM(p.price * o.quantity) INTO total_sales
        FROM products p
        JOIN orders o ON p.product_id = o.product_id
        WHERE p.category_id = p_category_id;

        IF total_sales IS NULL THEN
        SET total_sales = 0;
        END IF;

        RETURN total_sales;
        END;
```

Will the function fun be created without throwing an error? If yes, what does it return for p_category_id= 1? If no, what is causing the error?

The query will execute without any error. For p_category_id = 1 it will return the total sales for this id 1 which will be 4300.



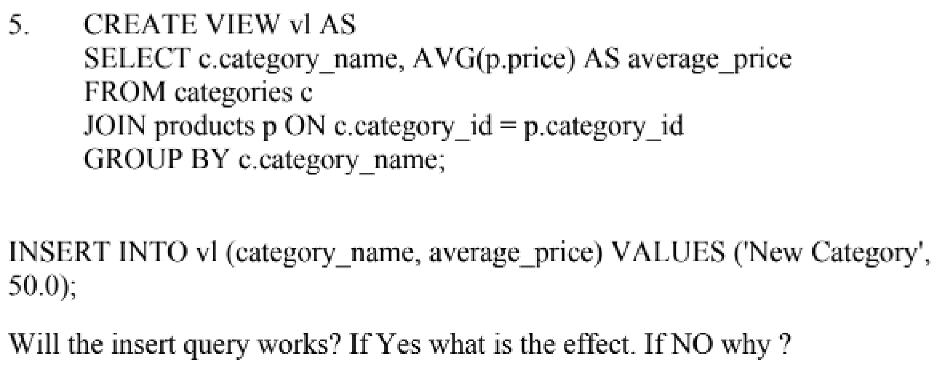
```
4.      CREATE PROCEDURE fun( IN p_product_id INT,  IN p_new_price
DECIMAL(10, 2))
        BEGIN
                DECLARE product_count INT;
                SELECT COUNT(*) INTO product_count
                FROM products
                WHERE product_id = p_product_id;

                IF product_count > 0 THEN
                UPDATE products
                SET price = p_new_price
                WHERE product_id = p_product_id;
                ELSE
                SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'Product not found', MYSQL_ERRNO =
                1001;
                END IF;
```



```
        END
Call fun(3,200)
```

Will the procedure fun be created without throwing an error? If yes, what does it do? If no, what is causing the error?

The procedure ‘fun’ will be created without throwing an error.
The procedure ‘fun(3, 200)’ updates the price of the product with ‘product_id = 3’ to the new price of 200. The product with ‘product_id = 3’ exists in the "products" table, which is T-shirts it will update its price from 20 to 200.

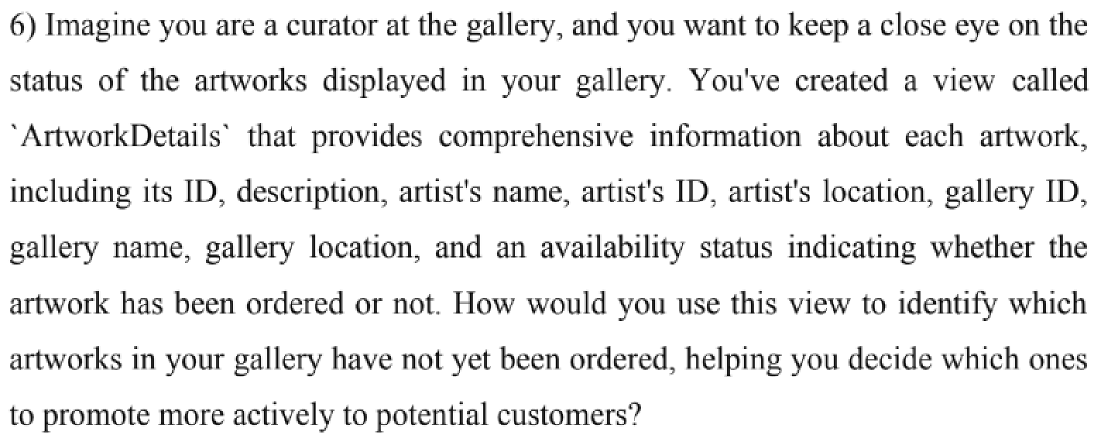


5.
```
CREATE VIEW vl AS
SELECT c.category_name, AVG(p.price) AS average_price
FROM categories c
JOIN products p ON c.category_id = p.category_id
GROUP BY c.category_name;
```

```
INSERT INTO vl (category_name, average_price) VALUES ('New Category',
50.0);
```

Will the insert query works? If Yes what is the effect. If NO why ?

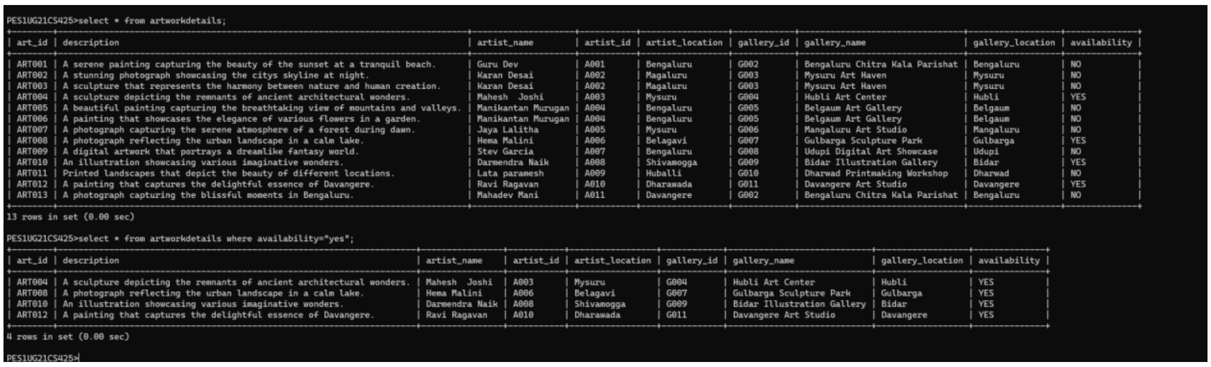
The insert query will not work, and it will produce an error.
Views are typically used for querying data and are often treated as read-only. We cannot directly perform any operation like INSERT, UPDATE, or DELETE operations on a view. Here "vl" view is a result of a SELECT statement that combines data from the categories" and products tables.



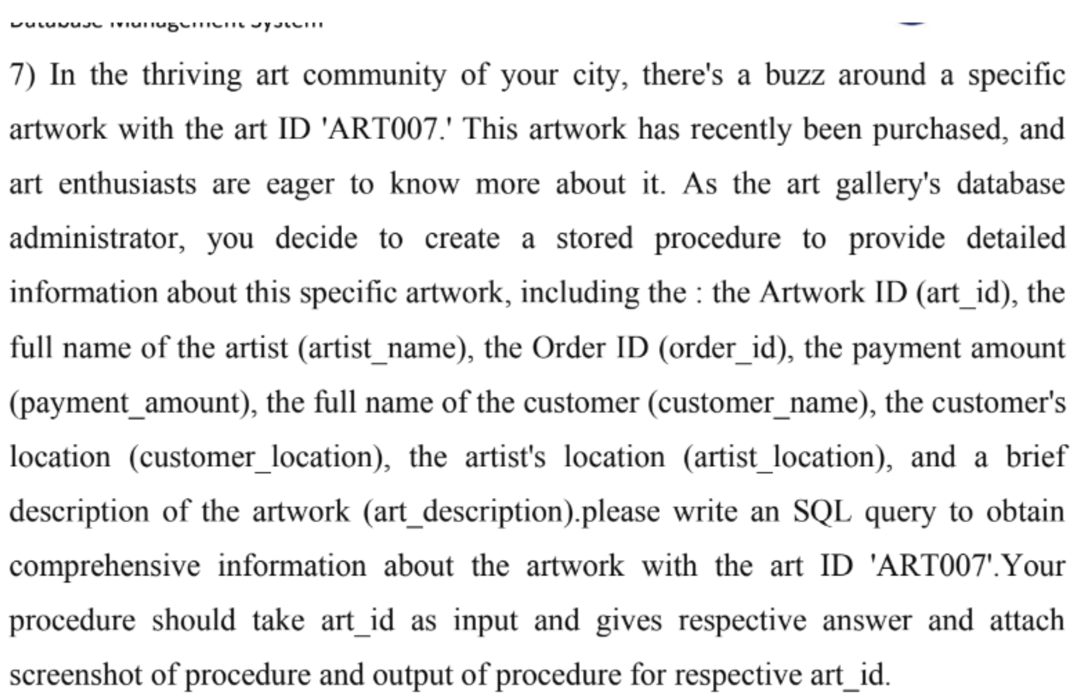
6) Imagine you are a curator at the gallery, and you want to keep a close eye on the status of the artworks displayed in your gallery. You've created a view called `ArtworkDetails` that provides comprehensive information about each artwork, including its ID, description, artist's name, artist's ID, artist's location, gallery ID, gallery name, gallery location, and an availability status indicating whether the artwork has been ordered or not. How would you use this view to identify which artworks in your gallery have not yet been ordered, helping you decide which ones to promote more actively to potential customers?

```
CREATE VIEW ArtworkDetails AS
SELECT
    a.art_id,
    a.art_description AS description,
    CONCAT(ar.f_name, ' ', ar.l_name) AS artist_name,
    ar.artist_id,
    ar.location AS artist_location,
    g.g_id AS gallery_id,
    g.g_name AS gallery_name,
    g.g_location AS gallery_location,
    a.availability
FROM
    art a
JOIN artist ar ON a.artist_id = ar.artist_id
JOIN gallery g ON a.gallery_id = g.g_id;
```

```
UPDATE art a
SET availability = 'NO'
WHERE EXISTS (
    SELECT 1 FROM purchase p WHERE p.art_id = a.art_id
);
```

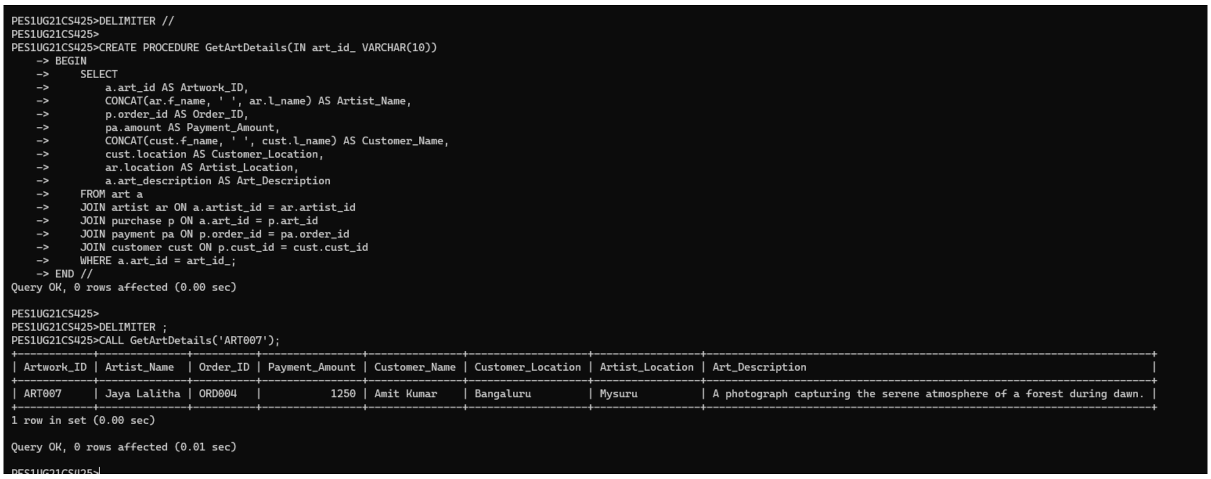


```
PES1UG21CS425>select * from artworkdetails;
+--------+--------------------------------------------------------------------------------+--------------------+-----------+-----------------+------------+--------------------------------+------------------+--------------+
| art_id | description                                                                    | artist_name        | artist_id | artist_location | gallery_id | gallery_name                   | gallery_location | availability |
+--------+--------------------------------------------------------------------------------+--------------------+-----------+-----------------+------------+--------------------------------+------------------+--------------+
| ART001 | A serene painting capturing the beauty of the sunset at a tranquil beach.      | Guru Dev           | A001      | Bengaluru       | G002       | Bengaluru Chitra Kala Parishat | Bengaluru        | NO           |
| ART002 | A stunning photograph showcasing the citys skyline at night.                   | Karan Desai        | A002      | Magaluru        | G003       | Mysuru Art Haven               | Mysuru           | NO           |
| ART003 | A sculpture that represents the harmony between nature and human creation.     | Karan Desai        | A002      | Magaluru        | G003       | Mysuru Art Haven               | Mysuru           | NO           |
| ART004 | A sculpture depicting the remnants of ancient architectural wonders.           | Mahesh  Joshi      | A003      | Mysuru          | G004       | Hubli Art Center               | Hubli            | YES          |
| ART005 | A beautiful painting capturing the breathtaking view of mountains and valleys. | Manikantan Murugan | A004      | Bengaluru       | G005       | Belgaum Art Gallery            | Belgaum          | NO           |
| ART006 | A painting that showcases the elegance of various flowers in a garden.         | Manikantan Murugan | A004      | Bengaluru       | G005       | Belgaum Art Gallery            | Belgaum          | NO           |
| ART007 | A photograph capturing the serene atmosphere of a forest during dawn.          | Jaya Lalitha       | A005      | Mysuru          | G006       | Mangaluru Art Studio           | Mangaluru        | NO           |
| ART008 | A photograph reflecting the urban landscape in a calm lake.                    | Hema Malini        | A006      | Belagavi        | G007       | Gulbarga Sculpture Park        | Gulbarga         | YES          |
| ART009 | A digital artwork that portrays a dreamlike fantasy world.                     | Stev Garcia        | A007      | Bengaluru       | G008       | Udupi Digital Art Showcase     | Udupi            | NO           |
| ART010 | An illustration showcasing various imaginative wonders.                        | Darmendra Naik     | A008      | Shivamogga      | G009       | Bidar Illustration Gallery     | Bidar            | YES          |
| ART011 | Printed landscapes that depict the beauty of different locations.              | Lata paramesh      | A009      | Huballi         | G010       | Dharwad Printmaking Workshop   | Dharwad          | NO           |
| ART012 | A painting that captures the delightful essence of Davangere.                  | Ravi Ragavan       | A010      | Dharawada       | G011       | Davangere Art Studio           | Davangere        | YES          |
| ART013 | A photograph capturing the blissful moments in Bengaluru.                      | Mahadev Mani       | A011      | Davangere       | G002       | Bengaluru Chitra Kala Parishat | Bengaluru        | NO           |
+--------+--------------------------------------------------------------------------------+--------------------+-----------+-----------------+------------+--------------------------------+------------------+--------------+
13 rows in set (0.00 sec)

PES1UG21CS425>select * from artworkdetails where availability="yes";
+--------+----------------------------------------------------------------------+----------------+-----------+-----------------+------------+----------------------------+------------------+--------------+
| art_id | description                                                          | artist_name    | artist_id | artist_location | gallery_id | gallery_name               | gallery_location | availability |
+--------+----------------------------------------------------------------------+----------------+-----------+-----------------+------------+----------------------------+------------------+--------------+
| ART004 | A sculpture depicting the remnants of ancient architectural wonders. | Mahesh  Joshi  | A003      | Mysuru          | G004       | Hubli Art Center           | Hubli            | YES          |
| ART008 | A photograph reflecting the urban landscape in a calm lake.          | Hema Malini    | A006      | Belagavi        | G007       | Gulbarga Sculpture Park    | Gulbarga         | YES          |
| ART010 | An illustration showcasing various imaginative wonders.              | Darmendra Naik | A008      | Shivamogga      | G009       | Bidar Illustration Gallery | Bidar            | YES          |
| ART012 | A painting that captures the delightful essence of Davangere.        | Ravi Ragavan   | A010      | Dharawada       | G011       | Davangere Art Studio       | Davangere        | YES          |
+--------+----------------------------------------------------------------------+----------------+-----------+-----------------+------------+----------------------------+------------------+--------------+
4 rows in set (0.00 sec)

PES1UG21CS425>
```

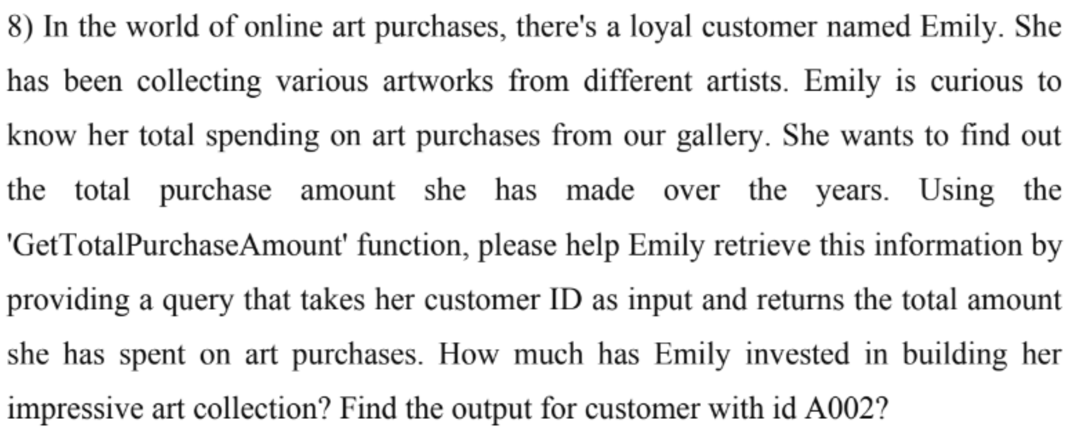


7) In the thriving art community of your city, there's a buzz around a specific artwork with the art ID 'ART007.' This artwork has recently been purchased, and art enthusiasts are eager to know more about it. As the art gallery's database administrator, you decide to create a stored procedure to provide detailed information about this specific artwork, including the : the Artwork ID (art_id), the full name of the artist (artist_name), the Order ID (order_id), the payment amount (payment_amount), the full name of the customer (customer_name), the customer's location (customer_location), the artist's location (artist_location), and a brief description of the artwork (art_description).please write an SQL query to obtain comprehensive information about the artwork with the art ID 'ART007'.Your procedure should take art_id as input and gives respective answer and attach screenshot of procedure and output of procedure for respective art_id.

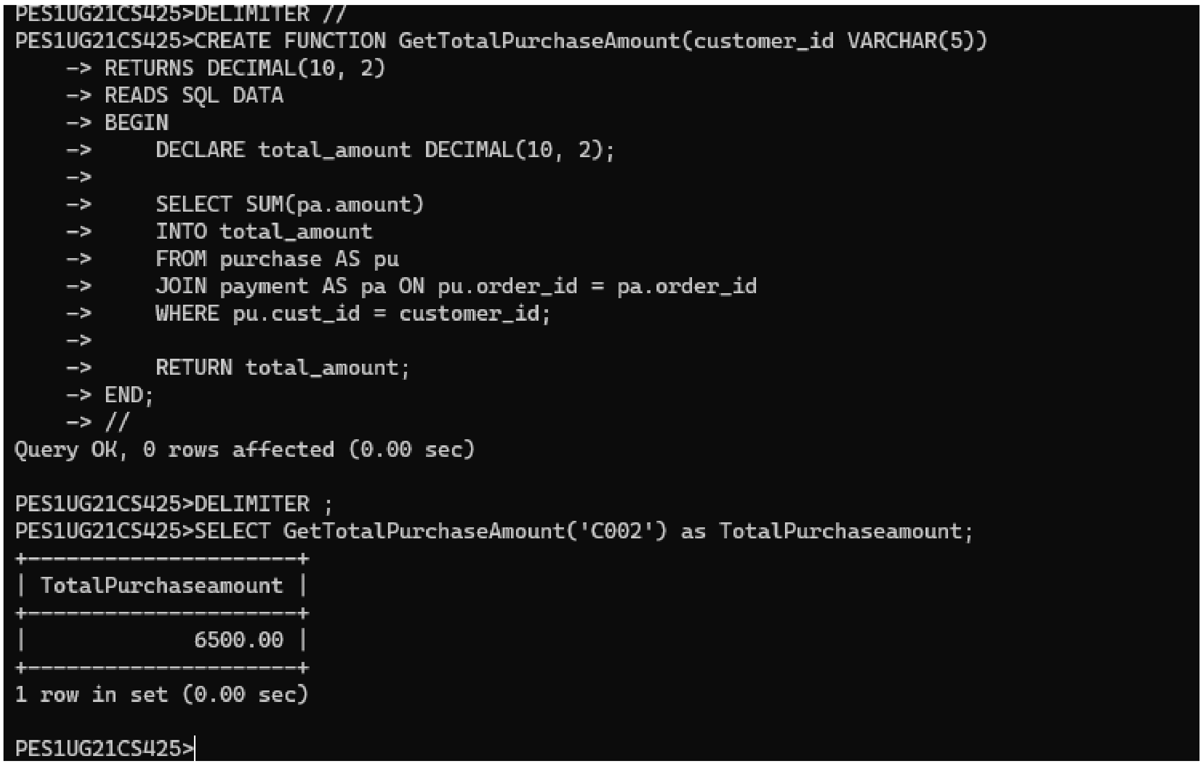
```
DELIMITER //

CREATE PROCEDURE GetArtDetails(IN art_id_ VARCHAR(10))
BEGIN
    SELECT
        a.art_id AS Artwork_ID,
        CONCAT(ar.f_name, ' ', ar.l_name) AS Artist_Name,
        p.order_id AS Order_ID,
        pa.amount AS Payment_Amount,
        CONCAT(cust.f_name, ' ', cust.l_name) AS Customer_Name,
        cust.location AS Customer_Location,
        ar.location AS Artist_Location,
        a.art_description AS Art_Description
    FROM art a
    JOIN artist ar ON a.artist_id = ar.artist_id
    JOIN purchase p ON a.art_id = p.art_id
    JOIN payment pa ON p.order_id = pa.order_id
    JOIN customer cust ON p.cust_id = cust.cust_id
    WHERE a.art_id = art_id_;
END //

DELIMITER ;

CALL GetArtDetails('ART007');
```



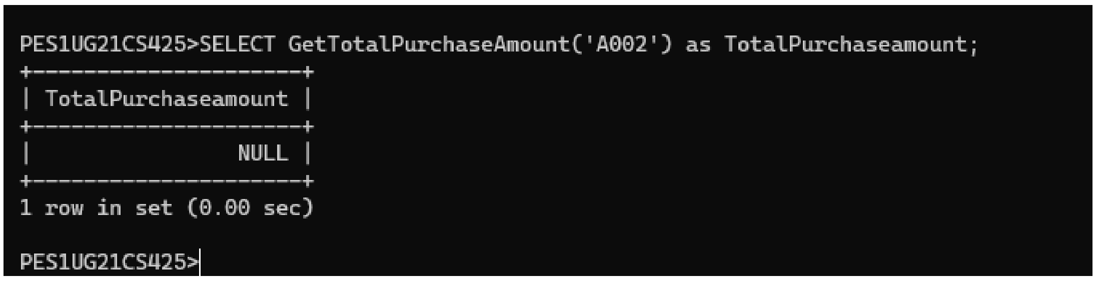
```
PES1UG21CS425>DELIMITER //
PES1UG21CS425>
PES1UG21CS425>CREATE PROCEDURE GetArtDetails(IN art_id_ VARCHAR(10))
    -> BEGIN
    ->     SELECT
    ->         a.art_id AS Artwork_ID,
    ->         CONCAT(ar.f_name, ' ', ar.l_name) AS Artist_Name,
    ->         p.order_id AS Order_ID,
    ->         pa.amount AS Payment_Amount,
    ->         CONCAT(cust.f_name, ' ', cust.l_name) AS Customer_Name,
    ->         cust.location AS Customer_Location,
    ->         ar.location AS Artist_Location,
    ->         a.art_description AS Art_Description
    ->     FROM art a
    ->     JOIN artist ar ON a.artist_id = ar.artist_id
    ->     JOIN purchase p ON a.art_id = p.art_id
    ->     JOIN payment pa ON p.order_id = pa.order_id
    ->     JOIN customer cust ON p.cust_id = cust.cust_id
    ->     WHERE a.art_id = art_id_;
    -> END //
Query OK, 0 rows affected (0.00 sec)

PES1UG21CS425>
PES1UG21CS425>DELIMITER ;
PES1UG21CS425>CALL GetArtDetails('ART007');
+------------+--------------+----------+----------------+---------------+-------------------+-----------------+------------------------------------------------------------------------+
| Artwork_ID | Artist_Name  | Order_ID | Payment_Amount | Customer_Name | Customer_Location | Artist_Location | Art_Description                                                        |
+------------+--------------+----------+----------------+---------------+-------------------+-----------------+------------------------------------------------------------------------+
| ART007     | Jaya Lalitha | ORD004   |           1250 | Amit Kumar    | Bangaluru         | Mysuru          | A photograph capturing the serene atmosphere of a forest during dawn. |
+------------+--------------+----------+----------------+---------------+-------------------+-----------------+------------------------------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

PES1UG21CS425>
```



8) In the world of online art purchases, there's a loyal customer named Emily. She has been collecting various artworks from different artists. Emily is curious to know her total spending on art purchases from our gallery. She wants to find out the total purchase amount she has made over the years. Using the 'GetTotalPurchaseAmount' function, please help Emily retrieve this information by providing a query that takes her customer ID as input and returns the total amount she has spent on art purchases. How much has Emily invested in building her impressive art collection? Find the output for customer with id A002?



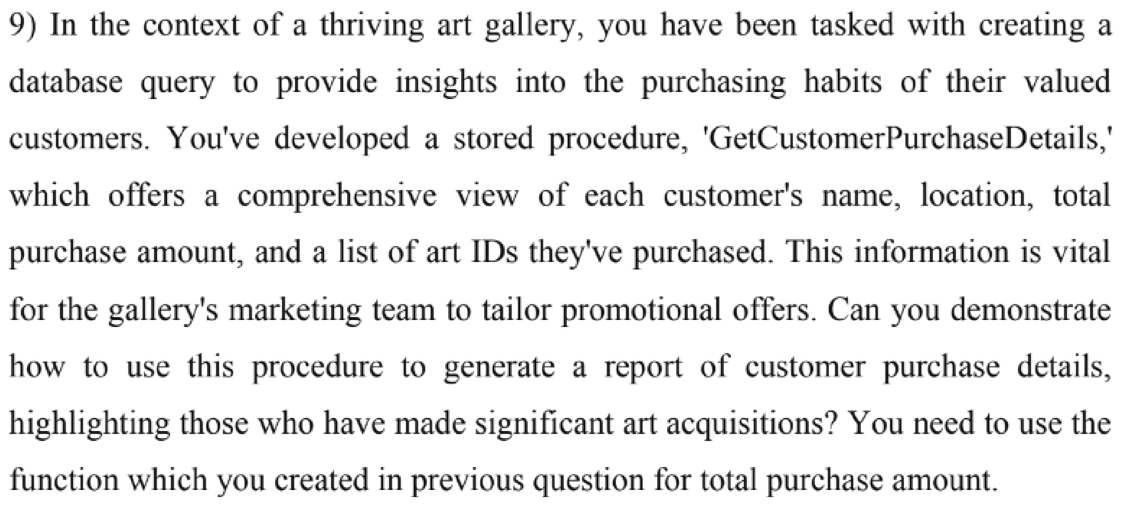
```
PES1UG21CS425>DELIMITER //
PES1UG21CS425>CREATE FUNCTION GetTotalPurchaseAmount(customer_id VARCHAR(5))
    -> RETURNS DECIMAL(10, 2)
    -> READS SQL DATA
    -> BEGIN
    ->     DECLARE total_amount DECIMAL(10, 2);
    ->
    ->     SELECT SUM(pa.amount)
    ->     INTO total_amount
    ->     FROM purchase AS pu
    ->     JOIN payment AS pa ON pu.order_id = pa.order_id
    ->     WHERE pu.cust_id = customer_id;
    ->
    ->     RETURN total_amount;
    -> END;
    -> //
Query OK, 0 rows affected (0.00 sec)

PES1UG21CS425>DELIMITER ;
PES1UG21CS425>SELECT GetTotalPurchaseAmount('C002') as TotalPurchaseamount;
+---------------------+
| TotalPurchaseamount |
+---------------------+
|             6500.00 |
+---------------------+
1 row in set (0.00 sec)

PES1UG21CS425>
```



```
PES1UG21CS425>SELECT GetTotalPurchaseAmount('A002') as TotalPurchaseamount;
+---------------------+
| TotalPurchaseamount |
+---------------------+
|                NULL |
+---------------------+
1 row in set (0.00 sec)

PES1UG21CS425>
```

```
DELIMITER //

CREATE FUNCTION GetTotalPurchaseAmount(customer_id VARCHAR(5))

RETURNS DECIMAL(10, 2)

READS SQL DATA

BEGIN

   DECLARE total_amount DECIMAL(10, 2);


   SELECT SUM(pa.amount)

   INTO total_amount

   FROM purchase AS pu

   JOIN payment AS pa ON pu.order_id = pa.order_id
```

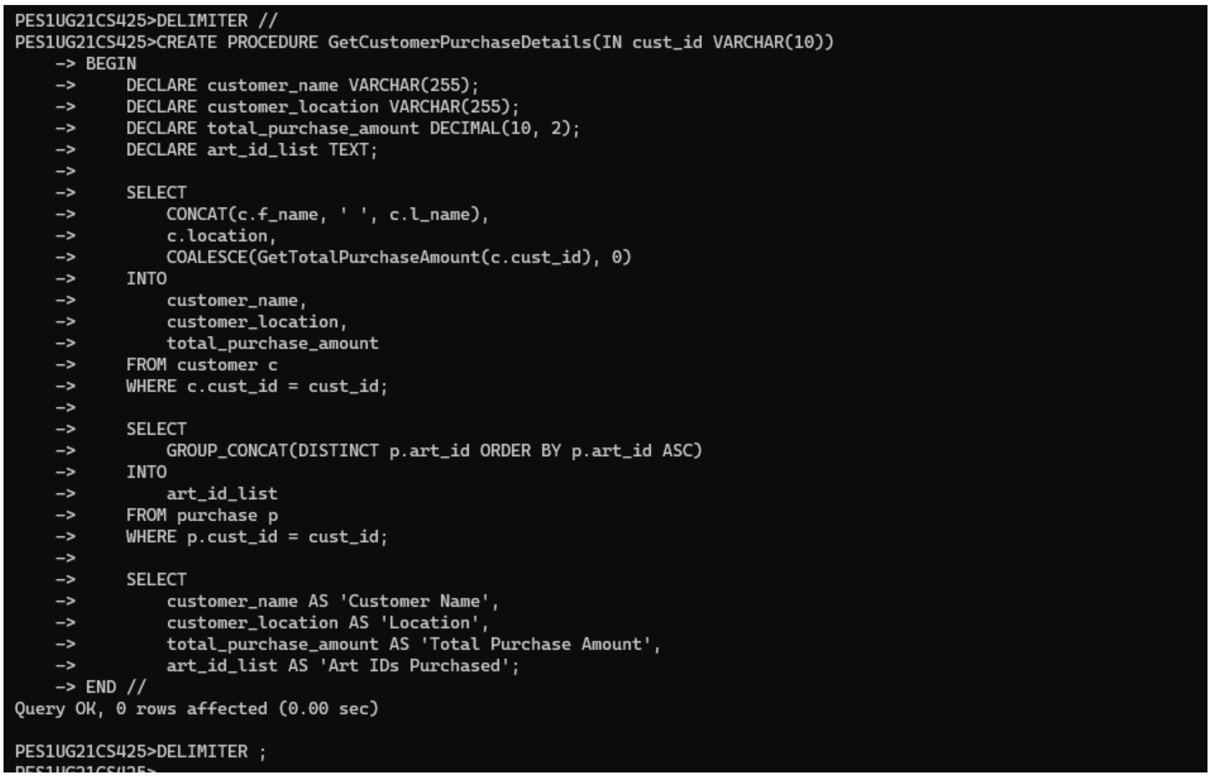
```
    WHERE pu.cust_id = customer_id;

    RETURN total_amount;
END;
//
DELIMITER ;
PES1UG21CS425>SELECT GetTotalPurchaseAmount('C002') as
TotalPurchaseamount;
```



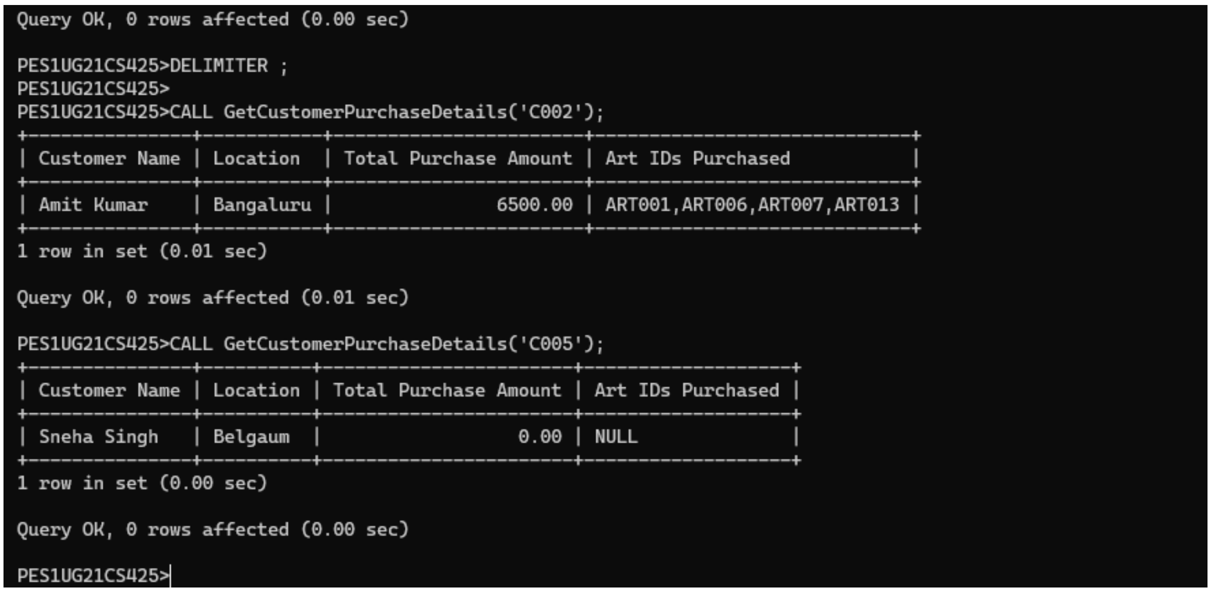
9) In the context of a thriving art gallery, you have been tasked with creating a database query to provide insights into the purchasing habits of their valued customers. You've developed a stored procedure, 'GetCustomerPurchaseDetails,' which offers a comprehensive view of each customer's name, location, total purchase amount, and a list of art IDs they've purchased. This information is vital for the gallery's marketing team to tailor promotional offers. Can you demonstrate how to use this procedure to generate a report of customer purchase details, highlighting those who have made significant art acquisitions? You need to use the function which you created in previous question for total purchase amount.

```
DELIMITER //
CREATE PROCEDURE GetCustomerPurchaseDetails(IN cust_id VARCHAR(10))
BEGIN
    DECLARE customer_name VARCHAR(255);
    DECLARE customer_location VARCHAR(255);
    DECLARE total_purchase_amount DECIMAL(10, 2);
    DECLARE art_id_list TEXT;

    SELECT
```

```
        CONCAT(c.f_name, ' ', c.l_name),
        c.location,
        COALESCE(GetTotalPurchaseAmount(c.cust_id), 0)
    INTO
        customer_name,
        customer_location,
        total_purchase_amount
    FROM customer c
    WHERE c.cust_id = cust_id;

    SELECT
        GROUP_CONCAT(DISTINCT p.art_id ORDER BY p.art_id ASC)
    INTO
        art_id_list
    FROM purchase p
    WHERE p.cust_id = cust_id;

    SELECT
        customer_name AS 'Customer Name',
        customer_location AS 'Location',
        total_purchase_amount AS 'Total Purchase Amount',
        art_id_list AS 'Art IDs Purchased';
END //
DELIMITER ;

CALL GetCustomerPurchaseDetails('C005');
```



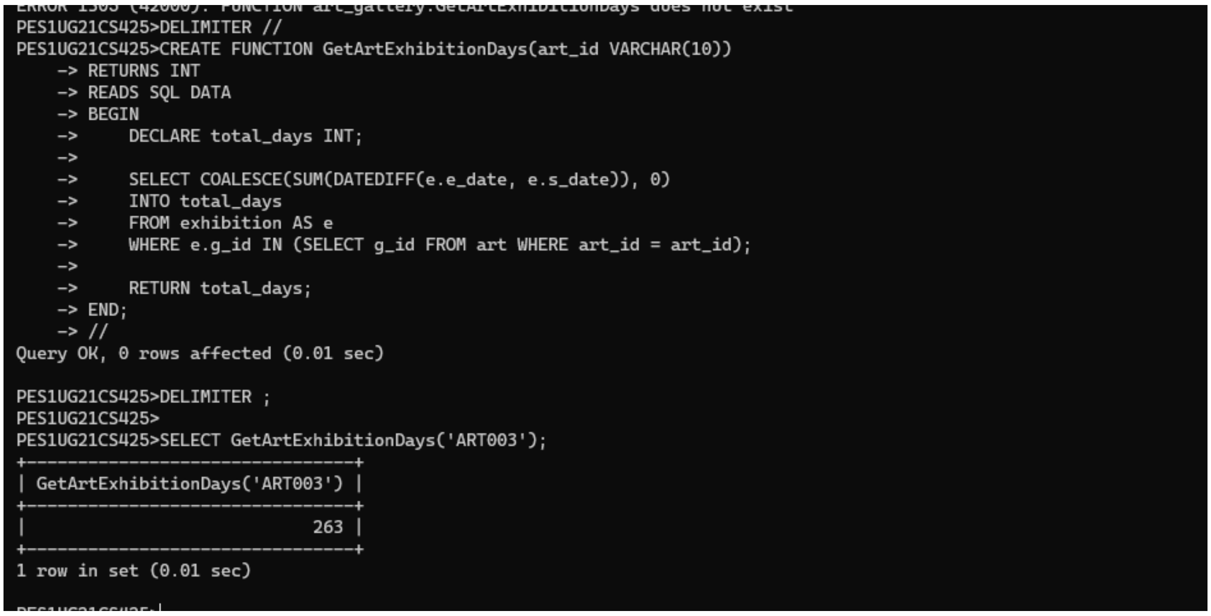
```
PES1UG21CS425>DELIMITER //
PES1UG21CS425>CREATE PROCEDURE GetCustomerPurchaseDetails(IN cust_id VARCHAR(10))
    -> BEGIN
    ->     DECLARE customer_name VARCHAR(255);
    ->     DECLARE customer_location VARCHAR(255);
    ->     DECLARE total_purchase_amount DECIMAL(10, 2);
    ->     DECLARE art_id_list TEXT;
    ->
    ->     SELECT
    ->         CONCAT(c.f_name, ' ', c.l_name),
    ->         c.location,
    ->         COALESCE(GetTotalPurchaseAmount(c.cust_id), 0)
    ->     INTO
    ->         customer_name,
    ->         customer_location,
    ->         total_purchase_amount
    ->     FROM customer c
    ->     WHERE c.cust_id = cust_id;
    ->
    ->     SELECT
    ->         GROUP_CONCAT(DISTINCT p.art_id ORDER BY p.art_id ASC)
    ->     INTO
    ->         art_id_list
    ->     FROM purchase p
    ->     WHERE p.cust_id = cust_id;
    ->
    ->     SELECT
    ->         customer_name AS 'Customer Name',
    ->         customer_location AS 'Location',
    ->         total_purchase_amount AS 'Total Purchase Amount',
    ->         art_id_list AS 'Art IDs Purchased';
    -> END //
Query OK, 0 rows affected (0.00 sec)

PES1UG21CS425>DELIMITER ;
PES1UG21CS425>
```



```
Query OK, 0 rows affected (0.00 sec)

PES1UG21CS425>DELIMITER ;
PES1UG21CS425>
PES1UG21CS425>CALL GetCustomerPurchaseDetails('C002');
+---------------+-----------+-----------------------+-----------------------------+
| Customer Name | Location  | Total Purchase Amount | Art IDs Purchased           |
+---------------+-----------+-----------------------+-----------------------------+
| Amit Kumar    | Bangaluru |               6500.00 | ART001,ART006,ART007,ART013 |
+---------------+-----------+-----------------------+-----------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

PES1UG21CS425>CALL GetCustomerPurchaseDetails('C005');
+---------------+----------+-----------------------+-------------------+
| Customer Name | Location | Total Purchase Amount | Art IDs Purchased |
+---------------+----------+-----------------------+-------------------+
| Sneha Singh   | Belgaum  |                  0.00 | NULL              |
+---------------+----------+-----------------------+-------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

PES1UG21CS425>
```

10) In the dynamic world of art appreciation, the gallery management is keen on recognizing the impact of each artwork. They want to understand the cumulative number of days an artwork has spent being showcased across various exhibitions. For this purpose, you've been assigned the task of creating a function. This function, named `GetArtExhibitionDays`, takes an art ID as input and returns the total number of days the artwork has been exhibited.Execute the function for 'ART003' to unveil the intriguing story of its exhibition journey.

```
DELIMITER //
CREATE FUNCTION GetArtExhibitionDays(art_id VARCHAR(10))
RETURNS INT
READS SQL DATA
BEGIN
   DECLARE total_days INT;

   SELECT COALESCE(SUM(DATEDIFF(e.e_date, e.s_date)), 0)
   INTO total_days
   FROM exhibition AS e
   WHERE e.g_id IN (SELECT g_id FROM art WHERE art_id = art_id);

   RETURN total_days;
END;
//
DELIMITER ;

SELECT GetArtExhibitionDays('ART003');
```



```
ERROR 1305 (42000): FUNCTION art_gallery.GetArtExhibitionDays does not exist
PES1UG21CS425>DELIMITER //
PES1UG21CS425>CREATE FUNCTION GetArtExhibitionDays(art_id VARCHAR(10))
    -> RETURNS INT
    -> READS SQL DATA
    -> BEGIN
    ->     DECLARE total_days INT;
    ->
    ->     SELECT COALESCE(SUM(DATEDIFF(e.e_date, e.s_date)), 0)
    ->     INTO total_days
    ->     FROM exhibition AS e
    ->     WHERE e.g_id IN (SELECT g_id FROM art WHERE art_id = art_id);
    ->
    ->     RETURN total_days;
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)

PES1UG21CS425>DELIMITER ;
PES1UG21CS425>
PES1UG21CS425>SELECT GetArtExhibitionDays('ART003');
+--------------------------------+
| GetArtExhibitionDays('ART003') |
+--------------------------------+
|                            263 |
+--------------------------------+
1 row in set (0.01 sec)

PES1UG21CS425>
```