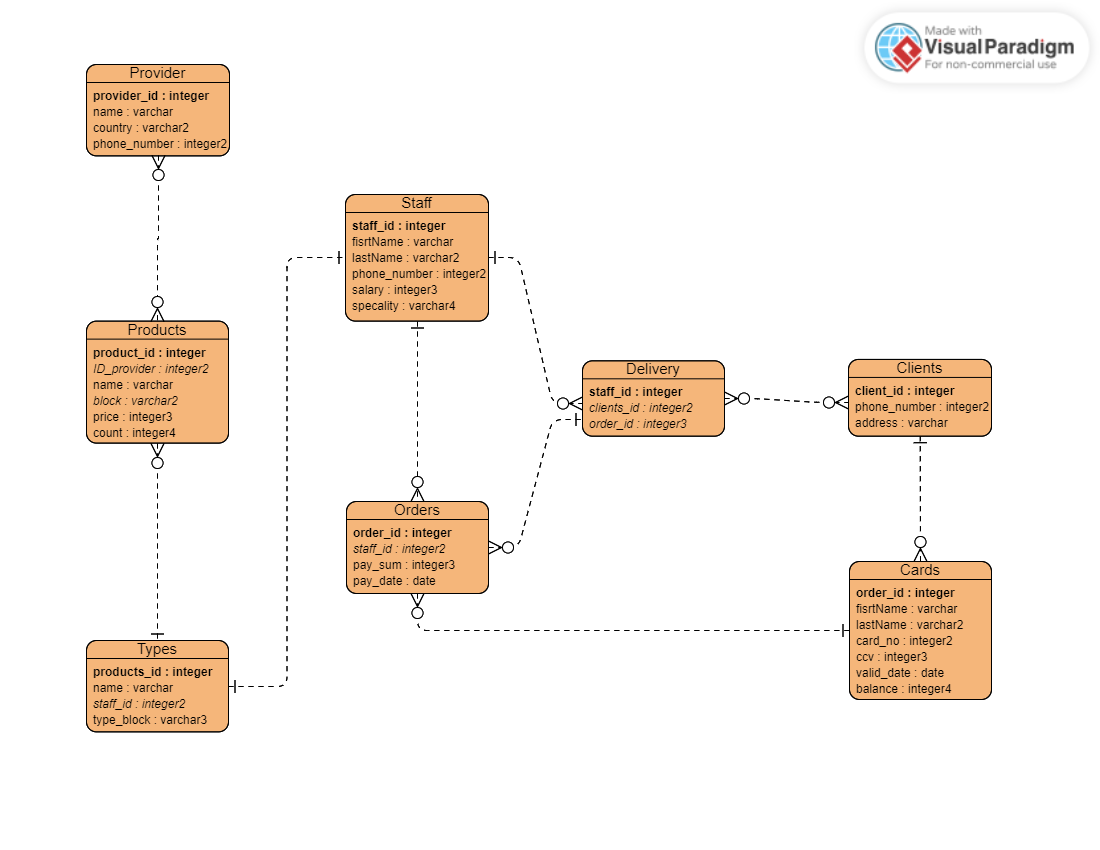
**Records**

DBMS EndTerm

**ER Diagram:**

****

**Table:**

**Cards :**

* **client\_id: an integer column that serves as the primary key for the table. This column is likely used to uniquely identify each client who has a credit card.**
* **cards\_firstname: a string column with a maximum length of 50 characters that stores the first name of the cardholder.**
* **cards\_lastname: a string column with a maximum length of 50 characters that stores the last name of the cardholder.**
* **cards\_no: an integer column that stores the card number of the credit card. The length of this column is not specified, so it is unclear how many digits are allowed for card numbers.**
* **cards\_ccv: an integer column that stores the CCV code of the credit card. This code is typically a 3- or 4-digit number used for security purposes.**
* **cards\_valid: a date column that stores the expiration date of the credit card.**
* **cards\_balance: an integer column that stores the current balance of the credit card. This column likely represents the amount of credit available to the cardholder.**

**Delivery:**

* **staff\_id: an integer column that likely stores the ID of the staff member responsible for delivering a product to a client. This column may be a foreign key referencing a staff table that contains more detailed information about the staff member.**
* **client\_id: an integer column that likely stores the ID of the client who placed the order for delivery. This column may be a foreign key referencing a client table that contains more detailed information about the client.**
* **payment\_id: an integer column that likely stores the ID of the payment associated with the delivery. This column may be a foreign key referencing a payment table that contains more detailed information about the payment.**

**Onlineclient:**

* **client\_id: an integer column that likely serves as the primary key for the table, uniquely identifying each online client.**
* **client\_username: a string column that likely stores the username that each client uses to log in to the online system. This column may have a uniqueness constraint to ensure that usernames are unique across all clients.**
* **client\_password: a string column that likely stores the password that each client uses to log in to the online system. This column may have some form of encryption or hashing applied to protect the client's password.**
* **client\_phone: a string column that likely stores the phone number associated with the client's account.**
* **client\_address: a string column that likely stores the address associated with the client's account.**

**Order:**

* **order\_id: an integer column that serves as the primary key for the table, uniquely identifying each order.**
* **client\_id: an integer column that likely stores the ID of the client who placed the order. This column may have a foreign key constraint referencing the client\_id column of the "onlineclient" table.**
* **staff\_id: an integer column that likely stores the ID of the staff member responsible for fulfilling the order. This column may have a foreign key constraint referencing the staff\_id column of another table.**
* **pay\_type: a string column that likely stores the payment type used for the order (e.g., "cash", "credit card", "PayPal", etc.).**
* **payment\_sum: an integer column that likely stores the total cost of the order.**
* **payment\_date: a date column that likely stores the date when the order was placed.**

**Product:**

* product\_id**: unique identifier for each product**
* product\_name**: name of the product**
* product\_type**: type or category of the product**
* product\_price**: price of the product**
* provider\_id**: identifier of the provider who supplies the product**

**Provider:**

* provider\_id**: unique identifier for each provider**
* provider\_name**: name of the provider**
* provider\_country**: country where the provider is located**
* provider\_phone**: contact phone number of the provider**

**Staff**:

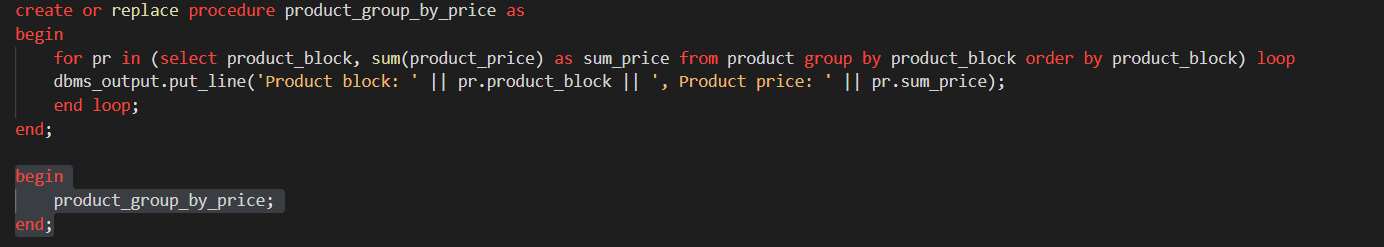
* staff\_id: unique identifier for each staff member
* first\_name: first name of the staff member
* last\_name: last name of the staff member
* salary: salary of the staff member
* phone\_number: contact phone number of the staff member
* speciality: area of expertise of the staff member

**Type:**

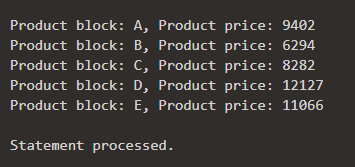
* type\_id**: a unique identifier for each product type.**
* type\_name**: the name of the product type.**
* staff\_id**: the ID of the staff member who manages the product type.**
* type\_block**: a categorization of the product type.**

**Procedure:**

This is a PL/SQL procedure that generates a report of the total price for each product block (category) in the "product" table. It uses a cursor to select the distinct product blocks from the table, then sums the prices of all products in each block using the "sum" aggregate function. The resulting totals are then output to the console using the "dbms\_output.put\_line" function, with each row containing the product block and its corresponding total price.

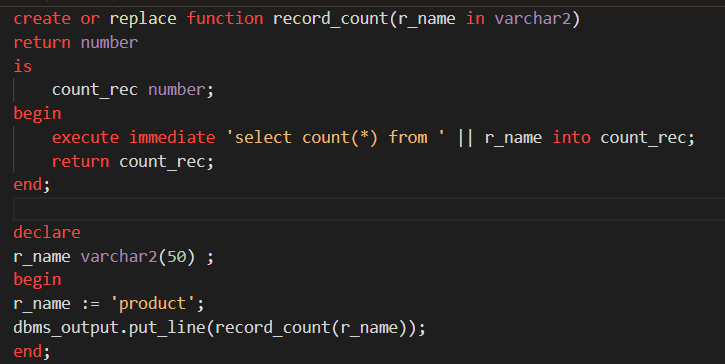
****

**Answer:**

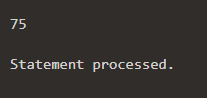
****

**Function:**

**The function record\_count takes a table name as input and returns the number of records in the table. It uses dynamic SQL to execute a count query on the specified table and stores the result in a variable before returning it. This function can be useful when you need to check the number of records in a table programmatically.**

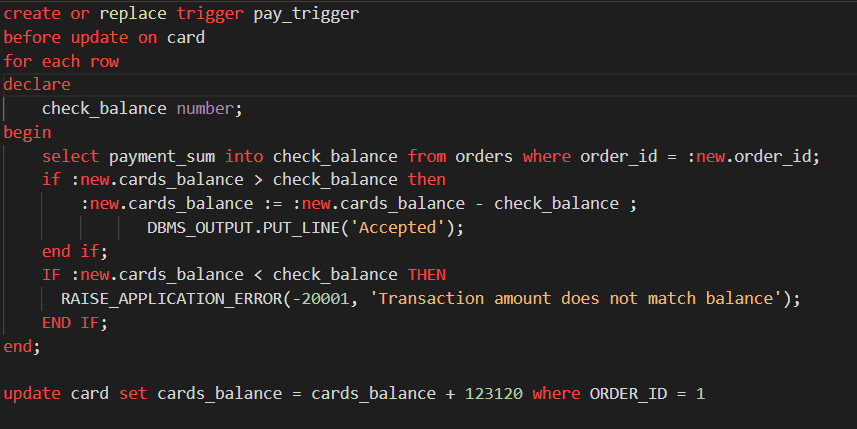
****

**Answer:**

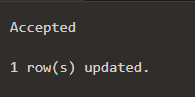
****

**Trigger:**

This trigger is designed to prevent updates to the "cards" table where the new balance is less than the payment amount associated with the payment ID in the same row. If the payment amount is greater than the current balance on the card, the trigger raises an application error with the message "Transaction amount does not match balance". If the payment amount is less than or equal to the current balance on the card, the trigger subtracts the payment amount from the card balance and outputs "Accepted" using DBMS\_OUTPUT.PUT\_LINE.

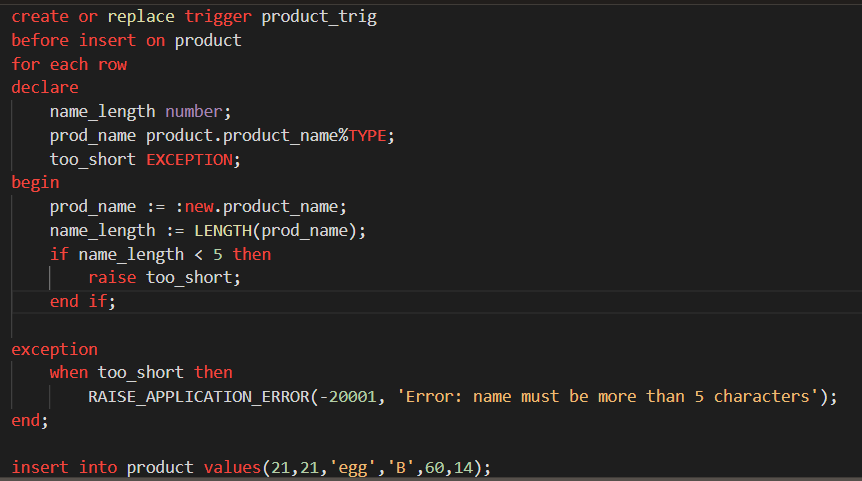
****

Answer:

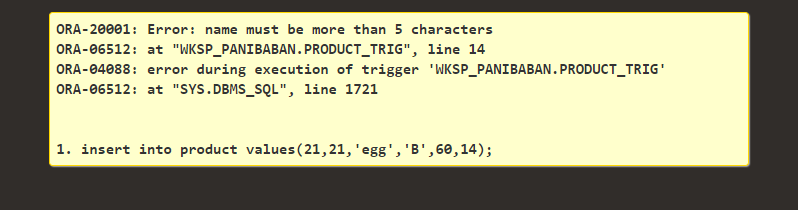


**Exception:**

**This trigger ensures that the product\_name column in the product table is not less than 5 characters before inserting a new row. If the length is less than 5, it raises a custom exception too\_short and then raises an application error with the message 'Error: name must be more than 5 characters'.**

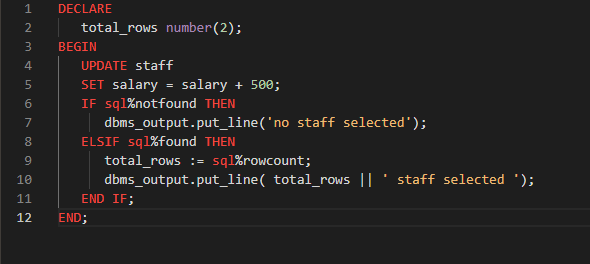
****

**Answer:**

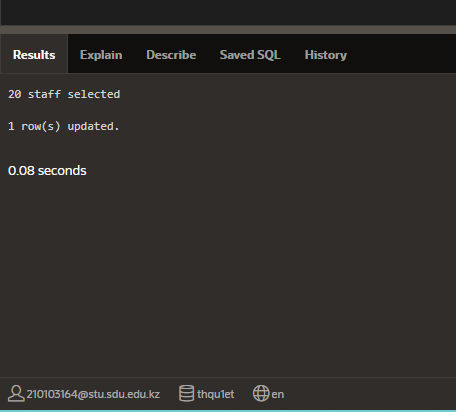
****

**Cursor:**

**This is a PL/SQL anonymous block that updates the salary of the staff in the "staff" table by adding 500 to their existing salary. The code then checks whether any rows were affected by the update using the SQL%NOTFOUND and SQL%FOUND attributes. If no rows were updated, it will output 'no staff selected', and if any rows were updated, it will output the total number of updated rows followed by 'staff selected'.**



**Answer:**

****

**By:**

**Askar Abylkhaiyrov ID:210103164 ;**

**Nagmetova Assem 210103109;**

**Zhaparova Aigerim ID:210103073;**

**Uzhanov Mars 210103187;**