Referentail Integrity Validation Using PL/SQL

Here is a simple example of how you can perform referential integrity validation using PL/SQL in an Oracle database:

Let's say we have two tables: "Orders" and "Customers." The "Orders" table has a foreign key constraint referencing the "Customers" table to ensure that every order is associated with a valid customer.

Create the "Customers" table:

```
CREATE TABLE Customers (
  customer_id INT PRIMARY KEY,
  customer_name VARCHAR2(100)
);
```

Create the "Orders" table with a foreign key constraint:

```
CREATE TABLE Orders (
  order_id INT PRIMARY KEY,
  order_date DATE,
  customer_id INT,
  CONSTRAINT fk_orders_customers FOREIGN KEY (customer_id)
    REFERENCES Customers (customer_id)
);
```

Now, let's create a PL/SQL procedure to validate the referential integrity when inserting a new order:

```
CREATE OR REPLACE PROCEDURE Insert_Order(
  p_order_id INT,
```

```
  p_order_date DATE,
  p_customer_id INT
) AS
  v_count INT;
BEGIN
  -- Check if the customer exists
  SELECT COUNT(*) INTO v_count
  FROM Customers
  WHERE customer_id = p_customer_id;

  -- If the customer doesn't exist, raise an exception
  IF v_count = 0 THEN
    RAISE_APPLICATION_ERROR(-20001, 'Invalid customer ID');
  END IF;

  -- Insert the order
  INSERT INTO Orders (order_id, order_date, customer_id)
  VALUES (p_order_id, p_order_date, p_customer_id);

  COMMIT;
  DBMS_OUTPUT.PUT_LINE('Order inserted successfully.');
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
```

/

Test the procedure by inserting an order with a valid customer ID:

```
BEGIN
  Insert_Order(1, SYSDATE, 1001);
END;
/
```

Test the procedure by inserting an order with an invalid customer ID:

```
BEGIN
  Insert_Order(2, SYSDATE, 9999);
END;
/
```

In the second test, since the customer with ID 9999 doesn't exist in the "Customers" table, the procedure will raise an exception and rollback the transaction.

This is a basic example of how you can perform referential integrity validation using PL/SQL.