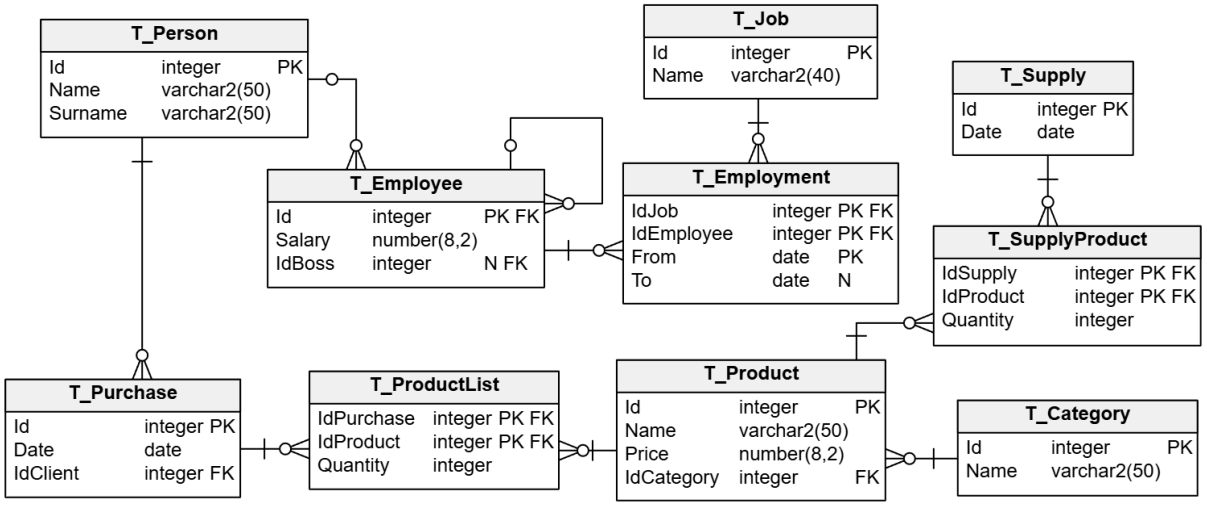
SBD Lab10
PL/SQL Triggers

Link to generate the database: link.
Link to drop the database: link.



Task 1

Create a simple trigger that will not allow us to delete records from the T_ProductList table. Use raise_application_error to print "You cannot delete records from T_ProductList",

Delete a record prior to creating a trigger as a test, it will work:

```
DELETE FROM T_ProductList
WHERE IdPurchase = 55 AND IdProduct = 4;
```

Then, after creating the trigger, try deleting another record:

```
DELETE FROM T_ProductList
WHERE IdPurchase = 55 AND IdProduct = 5;
```

Expected result:

```
[72000][20001]
ORA-20001: You cannot delete records from T_ProductList
ORA-06512: przy "PEUCESTAS.TR_PRODUCTLIST_DELETE", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.TR_PRODUCTLIST_DELETE'
Position: 12
```

Task 2

Transform the trigger from task 1 so that it also prints the following information: "Failed to delete record for purchase={IdPurchase} and product={IdProduct}." Use the :OLD reference for this purpose (the trigger must be FOR EACH ROW in order for this to work).

After updating the trigger, try deleting the record from the table again:

```
DELETE FROM T_ProductList
WHERE IdPurchase = 55 AND IdProduct = 5;
```

Expected result:

```
[72000][20001]
ORA-20001: You cannot delete records from T_ProductList.Failed to delete record for purchase=55 and product=5
ORA-06512: przy "PEUCESTAS.TR_PRODUCTLIST_DELETE", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.TR_PRODUCTLIST_DELETE'
Position: 12
```

Task 3

Drop the trigger from task 1 & 2, and then write an AFTER DELETE trigger for the T_Purchase table that will delete all records from T_ProductList associated with the purchase being deleted. Additionally, print the information: "Purchase with id = {id} was deleted".

Before creating the trigger, try deleting the purchase with id = 1, this will raise an error;

```
DELETE FROM T_Purchase WHERE id = 1;
```

Expected error:

```
[23000][2292] ORA-02292: naruszono więzy spójności (PEUCESTAS.T_PRODUCTLIST_T_PURCHASE) - znaleziono rekord podrzędny
Position: 0
```

After creating the trigger, try it again, now it should work;

```
DELETE FROM T_Purchase WHERE id = 1;
```

Expected result:

```
PEUCESTAS> DELETE FROM T_Purchase WHERE id = 1
[2024-12-12 22:16:18] 1 row affected in 80 ms
Purchase with id=1 was deleted
```

Task 4

Create a BEFORE trigger for the T_Employee table which will check whether the new salary (inserted or updated) is greater or less than 10,000. If it's greater than 10,000 then the trigger should report an error via raise_application_error and prevent the record from being inserted or updated.

Note: In this task, we use the trigger only for training purposes, because such functionality would be best implemented by creating a CHECK constraint on the salary column as follows:

```
ALTER TABLE T_Employee
ADD CONSTRAINT CHK_Salary CHECK (Salary <10000);
```

After creating the trigger, try updating the employee's salary to be greater than 10,000:

```
UPDATE T_Employee
SET salary = 40000
WHERE id =2;
```

Expected result:

```
[72000][20001]
ORA-20001: Salary exceeds the limit, DML operation failed.
ORA-06512: przy "PEUCESTAS.T_EMPLOYEE_BIUR", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.T_EMPLOYEE_BIUR'
Position: 7
```

After creating the trigger, try adding a new employee with a salary greater than 10,000:

```
INSERT INTO T_Employee(Id, salary) VALUES (7, 40000)
```

Expected result:

```
[72000][20001]
ORA-20001: Salary exceeds the limit, DML operation failed.
ORA-06512: przy "PEUCESTAS.T_EMPLOYEE_BIUR", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.T_EMPLOYEE_BIUR'
Position: 12
```

Task 5

Create a “BEFORE UPDATE OF price” trigger for T_Product table that will not allow you to lower the price. When we try to reduce the price we get an error: "The price cannot be reduced".

Once you've created your trigger, try reducing the price of one of your products:

```
UPDATE T_Product
SET price = 0.01
WHERE id = 2;
```

Expected result:

```
[72000][20001]
ORA-20001: The price cannot be reduced
ORA-06512: przy "PEUCESTAS.TR_PRODUCT_BUR", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.TR_PRODUCT_BUR'
Position: 7
```

Task 6

Drop the trigger from task 5, then write one trigger for T_Product that:
- will not allow adding a product with a price greater than 100 (in INSERT)
- will not allow the product price to be increased (in UPDATE)
- when deleting a product, it will delete all records for a given product from the T_ProductList table.

Task 7

Create a BEFORE INSERT trigger for the T_Person table that will not allow adding a new person if there is already a person with the same surname. If such a surname does not exist yet, then we add a new person with the information: "{name} has been successfully added."

After creating the trigger, try adding a person whose name already exists in T_Person:

```
INSERT INTO T_Person (id, name, surname) VALUES (11, 'Tim',
'Theramenes');
```

Expected result:

```
[72000][20001]
ORA-20001: Person with the give surname already exists
ORA-06512: przy "PEUCESTAS.TR_PERSON_BIR", linia 6
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.TR_PERSON_BIR'
Position: 12
```

After creating the trigger, try adding a person whose name does not exist in T_Person:

```
INSERT INTO T_Person (id, name, surname) VALUES (11, 'Tim',
'Thrasybulus');
```

Expected result:

```
[2024-12-13 09:53:48] 1 row affected in 105 ms
Thrasybulus was succesfully inserted.
```

After creating the trigger, try adding several people in one DML command and draw your conclusions:

```
INSERT ALL
   INTO T_Person (id, name, surname) VALUES (12, 'Liam', 'Thrasyllus')
   INTO T_Person (id, name, surname) VALUES (13, 'Keith', 'Conon')
   INTO T_Person (id, name, surname) VALUES (14, 'Reece',
'Callicratidas')
SELECT 1 FROM DUAL;
```

Expected result:

```
[42000][4091]
ORA-04091: tabela PEUCESTAS.T_PERSON ulega mutacji, wyzwalacz/funkcja może tego nie widzieć
ORA-06512: przy "PEUCESTAS.TR_PERSON_BIR", linia 4
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'PEUCESTAS.TR_PERSON_BIR'
Position: 21
```

Task 8

Create a BEFORE UPDATE trigger for the T_Employment table that will not allow you to update the values of any of the columns except the To column. Additionally, the To column can only be updated if its value is NULL and cannot be assigned a To date earlier than the From date.

Hint: use `IF UPDATING('IdEmployee') OR UPDATING('IdJob') OR UPDATING('From') THEN`

Task 9

Create a table T_SoldProducts with one "TotalValue" column that will store the value of all products sold and will always contain only one row. Create one trigger that will ensure that the value in the T_SoldProducts table is always up to date. For all operations updating the T_ProductList table (INSERT, UPDATE, DELETE), the trigger should update the value in the T_SoldProducts table.

Creating the T_SoldProducts table and assigning values to the "TotalValue" column:

```
CREATE TABLE T_SoldProducts(
   TotalValue number(8,2) not null
);

DECLARE
   v_value number(8,2);
BEGIN
   SELECT SUM(price * quantity) INTO v_value FROM T_ProductList pl JOIN
T_Product p ON pl.idproduct = p.id;
   INSERT INTO T_SoldProducts VALUES (v_value);
END;
```

Task 10

Create a View that includes the employee's name, salary and job. Then create an INSTEAD OF trigger for this View that will add a record to the database. If a person with the given name and surname does not exist, we add him to the T_Person table and proceed in the same way with the employee. If the employee already exists, we update his salary. If the

position does not exist, we create a new position. If the employee is not currently employed in a given position, we assign it to him, having previously removed him from his previous position with today's date (if he has previously held any position).

Creating a View:

```
CREATE VIEW V_Employee (Name, Surname, Salary, Job)
AS
SELECT p.name, p.surname, e.salary, j.name
FROM T_Person p JOIN T_Employee e ON p.id = e.id
JOIN T_Employment em ON em.idemployee = e.id
JOIN T_Job j ON j.id = em.idjob
WHERE em."To" IS NULL;
```