PL/SQL Triggers

In this session, you will learn:

- Overview of PL/SQL Trigger
- How to create a Trigger
- How to manage triggers

Overview of PL/SQL Trigger

- Triggers are stored programs, which are automatically executed or fired when some event occurs.
- It is executed or fired whenever an event associated with a table occurs e.g., DML(insert, update or delete)

Why to use Triggers?

- ✓ provide an alternative way to check the integrity of data.
- ✓ can catch errors in business logic in the database layer.
- ✓ useful to audit the changes of data in tables.

How to Create a Trigger

Syntax

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
Declaration-statements
BEGIN
Executable-statements
EXCEPTION
Exception-handling-statements
END;
```

Triggers – Example 1

Example

```
CREATE OR REPLACE TRIGGER display_price_changes
BEFORE DELETE OR INSERT OR UPDATE ON Product
FOR EACH ROW
WHEN (NEW.Product_Id > 0)
DECLARE
price_diff number;
BEGIN
price_diff := :NEW.Price - :OLD.Price;
INSERT INTO Product_Log VALUES(:OLD.Price, :NEW.Price, price_diff);
END;
/
```

Trigger Created

How to invoke a trigger

Product

Product_Id	Price	Pdt_Type
300	5000	Electronics
301	2500	Books
310	12000	Accessories

Product_Log

Old_Price	New_Price	Price_Diff
	12000	
2000	2500	500

Example 1

```
INSERT INTO Product VALUES (310, 12000, 'Accessories');
```

Example 2

```
UPDATE Product SET Price = Price + 500 WHERE Product_Id = 301;
```

Triggers – Example 2

Example

```
CREATE OR REPLACE TRIGGER del_pdt
AFTER DELETE ON Product
FOR EACH ROW
BEGIN
IF :OLD.Product_Id = 300 THEN
   raise_application_error(-20015, 'You cannot delete this row');
END IF;
END;
/
```

→ Trigger Created

```
DELETE FROM Product WHERE Product_Id = 300;
```

↓

ORA-20015: You cannot delete this row

THANKS