## Functions:

CREATE OR REPLACE FUNCTION **#Functionname#**(**#param#** in **#paramtype#**)

exception <<<< if exception is thrown in begin block start this

when NO\_DATA\_FOUND then : Exception example

when TOO\_MANY\_ROWS then : Exception example

when others then……. : Default Exception example

end

### Calling function:

select functionName(parameter) from DUAL; <<<< Outputs returned value(s)

eg. select RegNo, **#function#,** marks from marks2

## Cursors:

IS

Rec QA%RowType; <<< records var is like struct in C

CURSOR x is select \* from QA; <<< used to point to data

BEGIN

Open x;

LOOP

FETCH x into rec

EXIT when X%NOTFOUND <<<way to exit loop

DBMS\_OUTPUT.put\_line(‘Name : ’||Rec.Name);

DBMS\_OUTPUT.put\_line(‘Group : ’||Rec.GRP);

## Database Triggers

**USER\_TRIGGERS** < data dictionary shows list of triggers

**Triggers** are triggered if any DML commands happen (Insert, update, delete)

**Timing** is needed (when should the trigger trigger when the event occurs? *BEFORE* or *AFTER*?

**How many times**? Eg. *FOR EACH ROW*

|  |  |  |
| --- | --- | --- |
| **Per row** | **:NEW** <<value | **:OLD** <<value |
| **DELETE** | **X** | **/** |
| **INSERT** | **/** | **X** |
| **UPDATE** | **/** | **/** |

**Table shows where values will be null / not-null**

**Create or replace Trigger *trigger1***

*Before* delete on QA

FOR EACH ROW

BEGIN

INSERT into ***BACKUPTABLE***

*Values(****:old.****Regno,* ***:old****.Name,* ***:old****.GRP,* ***:old****.Area,* ***:old****.Marks,* ***user, sysdate****);* ***<< user and sysdate used to save name of the current user and the date the trigger is called***

END;

**Boolean values automatically generated when in trigger**

**Deleting <**

**Updating <**

**Inserting <**

**Create or replace trigger ABC**

Before **delete** or **update** or **insert** on ***school***

**For each row**

BEGIN

IF **UPDATING** then

…

End if;

IF **DELETING** then

…

End if;

IF **INSERTING** then

…

End if;