



The Developer can choose within the body of the PLSQL to skip the processing if certain conditions are not met, however one way to distinguish different triggers which fire for the same triggering event and timing point is to specify the optional 'When' clause, this will only fire the trigger when the condition is met.

In theory an event/timing point combination should only have one trigger dedicated to it, but there may be a need to use the When condition to limit to firing only on Sundays for example or for certain Database Roles.

The following gives several examples of the use of When conditions - Note the use of brackets...



## DML Triggers - DML Triggers When Conditions Examples

---

```
create or replace trigger prod_bi_row_trg  
before insert on products for each row  
when (user = 'TRYAN')
```

```
create or replace trigger prod_bi_row_trg2  
before insert on products for each row  
when (to_char(sysdate, 'DY') = 'FRI')
```

```
create or replace trigger prod_bi_row_trg3  
before insert on products for each row  
when (to_char(sysdate, 'DY') = 'FRI' and user = 'BTHOMAS')
```

```
create or replace trigger prod_bi_row_trg2  
before insert on products for each row  
when (new.pro_name = 'WIDGETS')
```



Note that all the previous examples of When conditions are declared as 'for each row', it is not possible to create a 'table' level trigger (that is, with no 'for each row') and combine it with a 'When' clause. If that is attempted then the following error will occur ...

```
create or replace trigger five  
before insert on temp  
when (to_char(sysdate, 'DY') = 'THU')  
declare begin null; end;
```

*ERROR at line 3:*

*ORA-04077: WHEN clause cannot be used with table level triggers*