Pragma is a compiler directive. All pragma will be defined in declared in pl/sql block.

All pragma will be processed during compilation time not at runtime. Once any plsql block defined as autonomous transaction as pragma , this pragma will be processed during compilation and compiled code will be stored in database as pragma. So whenever we execute plsql block which as pragma will not go for compilation again.

**Autonomous transaction**

Autonomous- independent

Plsql block which is defined as autonomous transaction will be executed independent of main transaction/session.

Restriction of autonomous transaction

Commit/rollback/savepoint must be used

**Advantages**

We can ran pl/sql block independent of main block.

For auditing purpose it will be very usefull.

Useful for logging(example if emp is updated/delted then we need to log in log table even though update/delete is failed to tract who performed DML operation)

**Disadvantage**

Commit should be used.

**Example**

**Without Autonomous transaction**

Calling prog procedure

Update ;

Proc

Commit;

Update;

Rollback

trans starts

Here calling program has the procedure which has rollback statement

Since the procedure also runs with the same session as of main program after the procedure completes

The update which is present in the calling program also gets rollbacked.

**With** **Autonomous transaction**

Calling prog procedure

Update ;

Proc

Commit;

Update;

Rollback

trans starts trans

starts

with autonomous transaction

the rollback present in procedure will rollback only statement in procedure and update in the main program will be committed

because autonomous tansaction runs procedure as independent of main program as separate session / transaction

create or replace procedure update\_emp(emo number)

as

begin

update emp set sal=4000 where empno=emo;

rollback;

end;

**main program**

begin

update emp set sal=5000 where empno=7839;

update\_emp(7845);

commit;

end;

/