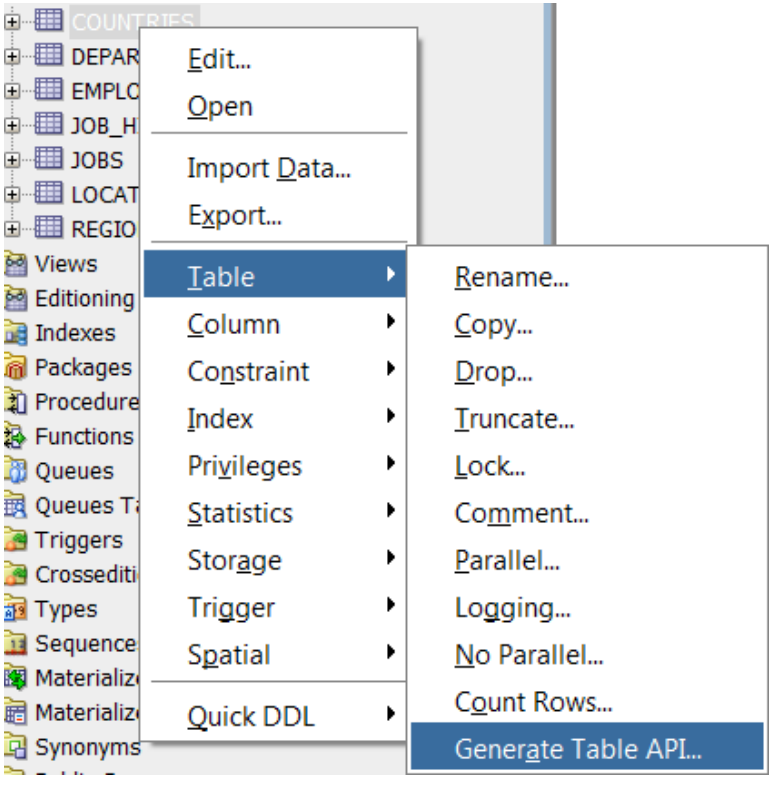
PL/SQL API

Table API (TAPI)

- V případě, že se všechny DML (INSERT, UPDATE, DELETE) operace budou vykonávat prostřednictvím PL/SQL procedur a nikoli přímo, pak hovoříme o Table API (TAPI).
- Při tomto přístupu má každá tabulka balíček, ve kterém jsou procerury (například ins, upd, del), které vykonávají příslušné DML operace.
- Tyto procedury je možné lehce vygenerovat pomocí SQL Developeru:



TAPI

- TAPI příklad:

```
package mytable_tapi is
    procedure ins (p_id integer, p_text varchar2);
    procedure upd (p_id integer, p_text varchar2);
    procedure del (p_id integer);
    function get (p_id integer) returns mytable%rowtype;
end;
```

TAPI vs. XAPI

- Samotné TAPI procedury je možné použít pouze u nejjednodušších aplikací. Součástí TAPI není zapouzdření do transakcí.
- Obvykle použijete Transactional API (XAPI) přístup, kdy Vaše procedury budou zapouzdřovat více operací.
- XAPI procedury není možné vygenerovat, protože je v nich Vaše business logika – která je u každého projektu jiná ;-)
- Existují dva přístupy:
  - Vygenerujete TAPI procedury a v XAPI je budete používat.
  - Nebudete TAPI používat vůbec, vše napíšete v XAPI.
  - Který přístup použít? Ten, který bude ve Vašem projektu méně pracnější ;-)

XAPI

- XAPI příklad:

```
package banking_xapi is
    procedure make_transfer (p_from_account integer,
                             p_to_account integer,
                             p_amount number);
    -- other XAPI procs
end;
```

make_transfer I.

- Procedura `make_transfer` pomocí TAPI:

```
procedure make_transfer (p_from_account integer,
                         p_to_account integer,
                         p_amount number)
is
begin
    transactions_tapi.ins (p_from_account, p_to_account,
                                    p_amount);

    accounts_tapi.upd (p_from_account, -p_amount);

    accounts_tapi.upd (p_to_account, p_amount);
end;
```

make_transfer II.

- Procedura `make_transfer` pouze pomocí XAPI:

```
procedure make_transfer (p_from_account integer, p_to_account integer,
                         p_amount number)
is
begin
    insert into transfer_details (from_account, to_account, amount)
       values (p_from_account, p_to_account, p_amount);

    update accounts set balance = balance-p_amount
    where account_no = p_from_account;

    update accounts set balance = balance+p_amount
    where account_no = p_to_account;
end;
```