PL/SQL

Rozszerzenie języka SQL - Procedural Language/SQL.
Język blokowo-strukturalny

```
BEGIN
 insert into tabela1 (pole1, pole2) values (1,'abc');
 insert into tabela1 (pole1, pole2) values (2,'xyz');

???
 commit;
END;
.
```

PL/SQL - struktura bloku

```
declare
  deklaracja zmiennych
  ...
begin
  główna część bloku
  ...
exception when (...) then
  sekcja obsługi wyjątków
  ...
end;
```

PL/SQL - zagnieżdżanie bloków

```
declare
  x number(4);
begin
   declare
     y number(4);
   begin
     ...
   exception when typ wyjątku then
    ...
   end;
exception when typ wyjątku  then
  ...
end;
```

PL/SQL - podstawienia i komentarze

```
declare
  x number(4);
begin
   x := 1;

    -- jednolinijkowy komentarz

   x := x+1;

   /* komenarz
       wielolinijkowy */
end;
```

PL/SQL - instrukcja warunkowa

```
declare
  x number(4);
begin
   x := ?;
   ....
   if x = 1 then
     ....
   elsif x = 2 then
     ....
   else
     ....
   end if;
end;
```

<, >, =, <=, >=, <>,

and, or, not

operatory SQL:

is null, like, between, in

PL/SQL - pętle

```
loop
  ....
  ....
exit when warunek ;
end loop;
```

```
Declare
  i number(3);
begin
  i:=1;
  loop
    insert into tabela (p1, p2)
          values (i, i*i);
    exit when i=10;
    i:= i+1;
  end loop;
end;
```

PL/SQL - pętle

```
while warunek loop
  ....
  ....
end loop;
```

```
Declare
  i number(3);
begin
  i:=1;
  while i<=10 loop
    insert into tabela (p1, p2)
          values (i, i*i);
    i:= i+1;
  end loop;
end;
```

PL/SQL - pętle

```
for i in 1 .. 10 loop
  ....
  ....
end loop;
```

```
Declare
  i number(3);
begin
 for i in 1..10 loop
    insert into tabela (p1, p2)
          values (i, i*i);
end loop;
end;
```

PL/SQL - select

```
begin
  select * from emp;
end;
```

PLS-00428: w tej instrukcji SELECT spodziewano się klauzuli INTO

```
Declare
  x number(20);
begin
 select empno into x from
   emp where ename = 'SCOTT';
end;
```

PL/SQL - select

```
Declare
  x number(20);
begin
 select empno into x from emp;
end;
```

ORA-01422: dokładne pobranie zwraca większą liczbę wierszy niż zamówiono.

PL/SQL - select

```
declare
  xjob varchar2(20);
  xsal number(20);
begin
 select job,sal into xjob,xsal
   from emp
     where ename = 'KING';

  /* ...  */

  insert into test (p1,p2) values (xjob,xsal);
  commit;
end;
```

PL/SQL - typy zmiennych

```
declare
  xjob emp.job%TYPE;
  xsal emp.sal%TYPE;
begin
 select job,sal into xjob,xsal
   from emp
     where ename = 'KING';

  /* ...  */

  insert into test (p1,p2) values (xjob,xsal);
  commit;
end;
```

PL/SQL - zmienna rekordowa

```
declare
  type pracownik is record
  (
     nazwisko emp.ename%TYPE,
     wiek number
  );

xprac1 pracownik;
...
BEGIN
 ...
END;
```

PL/SQL - zmienna rekordowa

```
declare
  xprac2 emp%ROWTYPE;
begin
 xprac2.empno := 6666;
 xprac2.ename := 'JANICKI';
 xprac2.sal := 3750;
 insert into emp
  (empno,ename,sal)
 values
  (xprac2.empno, xprac2.ename, xprac2.sal);
end;
```

PL/SQL - kursory

CURSOR, OPEN, FETCH, CLOSE

```
declare
 cursor XYZ is
 select deptno,dname from dept;
 xdeptno dept.deptno%type;
 xdname dept.dname%type;
begin
 open xyz;
 fetch xyz into xdeptno,xdname;
 fetch xyz into xdeptno,xdname;
 fetch xyz into xdeptno,xdname;
 close xyz;

end;
```

PL/SQL - atrybuty kursora

%FOUND, %NOTFOUND, %ROWCOUNT, %ISOPEN

```
declare
 cursor XYZ is
 select deptno,dname from dept;
 xdeptno dept.deptno%type;
 xdname dept.dname%type;
begin
  if XYZ%ISOPEN then
     fetch XYZ into xdeptno, xdname;
  else
    open XYZ;
  end if;

end;
```

PL/SQL - atrybuty kursora

```
begin
  loop
   fetch XYZ into xdeptno, xdname;
   exit when XYZ%ROWCOUNT>10;
  end loop;

end;
```

PL/SQL - atrybuty kursora

```
begin
  loop
   fetch XYZ into xdeptno, xdname;
   exit when XYZ%NOTFOUND;

   /* ... */

  end loop;
end;
```

PL/SQL - kursory c.d.

```
declare
  cursor XYZ is
    select deptno,dname from dept;
 rec dept%rowtype;
begin
  ...
  fetch XYZ into rec.deptno, rec.dname;
```

```
declare
  cursor XYZ is
    select deptno,dname from dept;
 rec XYZ%rowtype;
begin
  ...
  fetch XYZ into rec;
```

PL/SQL - kursory c.d.

```
declare
  cursor XYZ is
    select deptno,dname from dept;
  rec XYZ%rowtype;
begin
  for rec in XYZ loop
    fetch XYZ into rec.deptno, rec.dname;
    rec.dname := rec.dname||'xxx';
  end loop;
end;
```