

PL / SQL

Topik

- Identifikasi obyek PL/SQL
- Trigger dan event trigger
- Identifikasi konfigurasi untuk performance PL/SQL

PL/SQL

Procedural Language/Structured Query Language (PL/SQL) adalah generasi ke-4 (4GL:Fourth–Generation Languages) dari bahasa pemrograman.

PL/SQL menyediakan:

- Tambahan penggunaan prosedur pada SQL
- Kemudahan untuk digunakan secara lintas platform dan produk
- Mendukung pemrograman berorientasi obyek

Melakukan administrasi obyek² PL/SQL

Administrator database diharapkan punya kemampuan untuk:

- Mengidentifikasi permasalahan obyek PL/SQL
- Merekomendasikan penggunaan PL/SQL yang tepat
- Memasukkan objek PL/SQL ke dalam database
- Membantu developer PL/SQL dalam troubleshooting

Obyek-Obyek PL/SQL

Ada beberapa tipe dari obyek-obyek PL/SQL dalam database, yaitu:

- **Package**
Kumpulan dari prosedur dan fungsi yang terelasi secara logic.
- **Package body**
Body secara penuh mendefinisikan kursor dan subprogram.
- **Type body**
Kumpulan dari method-method (prosedur-prosedur dan fungsi-fungsi) yang berhubungan dengan tipe data user-defined.
- **Procedure**
Sebuah blok PL/SQL yang melakukan aksi khusus.
- **Function**
Blok PL/SQL yang mengembalikan nilai tunggal dengan menggunakan perintah PL/SQL RETURN.
- **Trigger**
Blok PL/SQL yang dieksekusi ketika terjadi sebuah kejadian/event tertentu di dalam database.

Functions

Database: orcl.us.oracle.com > Functions Logged in As RIC

Functions

Search

Select an object type and optionally enter a schema name and an object name to filter the data that is displayed in your results set.

Object Type: Schema: Object Name:

To run an exact match search or to run a case sensitive search, double quote the search criteria. The wildcard (%) symbol can still be used in a double quoted search string.

Results

Select Schema	Function Name	Created	Last Modified	Status
No search conducted				

Database: orcl.us.oracle.com > Functions > Create Function Logged in As RIC

Create Function

* Name:

* Schema:

* Source:

```
( salary NUMBER )
RETURN NUMBER
AS
BEGIN
  IF salary<5000 THEN
    RETURN salary*.15;
  ELSE
    RETURN salary*.33;
  END IF;
END;
```

- **Fungsi** digunakan untuk menghitung sebuah nilai.
- Terdapat beberapa fungsi built-in spt: SYSDATE, SUM, AVG, dan TO_DATE.
- Developer juga membuat fungsi utk kebutuhan sendiri ketika menulis aplikasi.
- Fungsi PL/SQL harus mengandung pernyataan RETURN.
- Contoh fungsi dalam SQL:

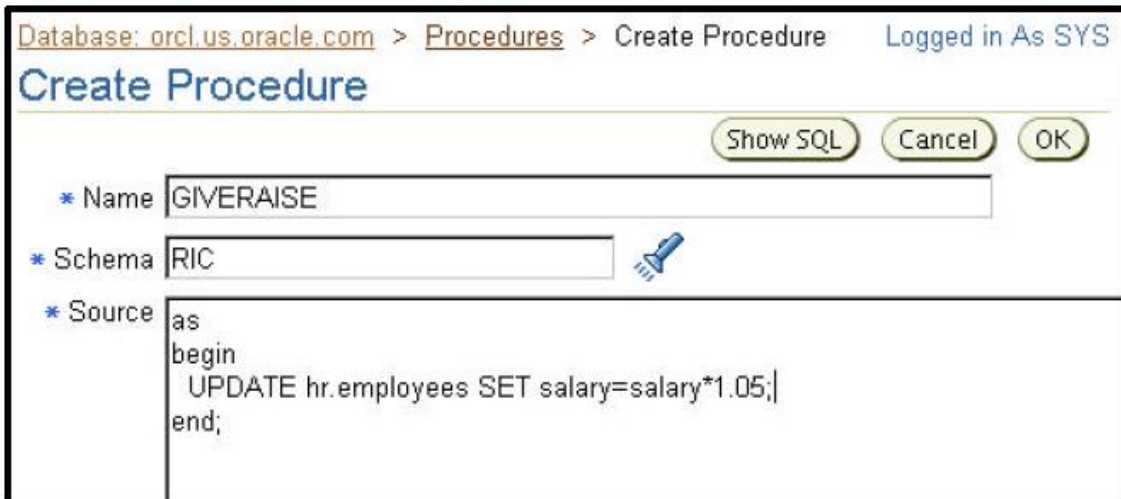
```
CREATE OR REPLACE FUNCTION computetax (salary
NUMBER)
RETURN NUMBER
AS
BEGIN
  IF salary<5000 THEN
    RETURN salary*.15;
  ELSE
    RETURN salary*.33;
  END IF
END;
/
```

Procedures

Prosedur digunakan untuk melakukan sebuah aksi khusus.

Prosedur:

- Memindah nilai ke dalam (IN) dan keluar (OUT) melalui sebuah daftar argumen.
- Dipanggil dengan perintah **CALL**.



Database: orcl.us.oracle.com > Procedures > Create Procedure Logged in As SYS

Create Procedure

Show SQL Cancel OK

* Name: GIVERAISE

* Schema: RIC

* Source:
as
begin
 UPDATE hr.employees SET salary=salary*1.05;
end;

```
SQL> SELECT sum(salary) FROM  
hr.employees;  
SUM(SALARY)
```

```
-----  
691400
```

```
SQL> call giveraise();  
Call completed.
```

```
SQL> SELECT sum(salary) FROM  
hr.employees;  
SUM(SALARY)
```

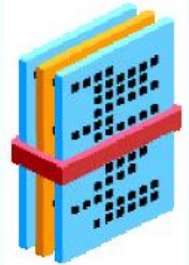
```
-----  
725970
```

Packages

Packages adalah kumpulan dari fungsi dan prosedur.

Tiap package harus terdiri dari **dua obyek**:

- **Package Specification** (package header)
- **Package Body**



Database: orcl.us.oracle.com > Packages > Create Package Logged in As RIC

Create Package

Show SQL Cancel OK

* Name

* Schema 

* Source
as
FUNCTION computetax (salary NUMBER) RETURN NUMBER;
PROCEDURE giveraise;
end;

Cara memanggil:

```
package_name.procedure or  
function name
```

Cara eksekusi:

```
SQL> SELECT  
money.computetax(salary)  
FROM hr.employees  
WHERE employee_id=107;  
  
SQL> EXECUTE money.giveraise;
```


- **Package Specification** : merupakan antarmuka terhadap aplikasi-aplikasi kita; ia mendeklarasikan types, variables, constants, exceptions, cursors, dan subprograms yang tersedia untuk digunakan (package header).

Package Body

Database: orcl.us.oracle.com > Package Bodies > Create Package Body Logged in As RIC

Create Package Body

* Name

* Schema 

* Source

```
as
FUNCTION computetax (salary NUMBER) RETURN NUMBER IS
    BEGIN
        IF salary<5000 THEN
            RETURN salary*.15;
        ELSE
            RETURN salary*.33;
        END IF;
    END computetax;
PROCEDURE giveraise AS
    BEGIN
        EXECUTE IMMEDIATE 'UPDATE hr.employees SET salary=salary*1.05';
    END giveraise;
END money;
```

- **Package Body** adalah Body secara penuh mendefinisikan cursors dan subprograms, dan juga mengimplementasikan spec.

Built-In Packages

Database Oracle 10g mempunyai lebih dari 350 paket PL/SQL built-in yang menyediakan:

- Utiliti untuk administrasi dan pemeliharaan
- Fungsi untuk pengembangan

Gunakan perintah **DESCRIBE** untuk menampilkan subprogram

```
SQL> DESCRIBE dbms_stats
PROCEDURE ALTER_DATABASE_TAB_MONITORING
Argument Name      Type          In/Out        Default?
-----
MONITORING         BOOLEAN      IN            DEFAULT
SYSOBJJS           BOOLEAN      IN            DEFAULT
...
```

Triggers

Database: orcl.us.oracle.com > Triggers Logged in As SYS

Triggers

Search

Select an object type and optionally enter a schema name and an object name to filter the data that is displayed in your results set.

Object Type: Schema: Object Name:

To run an exact match search or to run a case sensitive search, double quote the search criteria. The wildcard (%) symbol can still be used in a double quoted search string.

Results

Edit View Delete Actions Create Like

Select	Schema	Trigger Name	Type	Event	Base Object Type	Base Object Owner	Base Object Name	Status	Enabled?
<input checked="" type="radio"/>	HR	<u>SECURE_EMPLOYEES</u>	BEFORE STATEMENT	INSERT OR UPDATE OR DELETE	TABLE	HR	EMPLOYEES	VALID	NO
<input type="radio"/>	HR	<u>UPDATE_JOB_HISTORY</u>	AFTER EACH ROW	UPDATE	TABLE	HR	EMPLOYEES	VALID	YES

- **Trigger** : blok PL/SQL atau prosedur yang berhubungan dengan table, view, skema atau database yang dijalankan secara implicit pada saat terjadi sebuah **event**.
- **Tipe dari trigger** :
 - **Application trigger** : diaktifkan pada saat terjadi event yang berhubungan dengan sebuah aplikasi
 - **Database trigger** : diaktifkan pada saat terjadi event yang berhubungan dengan data (seperti operasi DML) atau event yang berhubungan dengan sistem (semisal logon atau shutdown) yang terjadi pada sebuah skema atau database.

Option Konfigurasi PL/SQL

Ada beberapa setting dari compiler PL/SQL yang mengontrol performa kinerja PL/SQL. Untuk performa lebih cepat, dapat dilakukan pengaturan parameter:

- **PLSQL_CODE_TYPE=NATIVE**

PL/SQL yang sudah dikompilasi dapat disimpan dengan 2 cara: sebagai interpreted bytecode atau sebagai native machine code. Defaultnya, interpreted bytecode menyediakan fitur debugging yang lebih baik untuk pengembangan. **Native machine** code menyediakan kinerja run-time terbaik, sampai dengan dua kali kecepatan interpreted bytecode.

- **PLSQL_DEBUG=FALSE**

Setting default FALSE mematikan fitur debugging lebih lanjut.

- **PLSQL_OPTIMIZE_MODE=2**

Setting default 1 menyediakan kinerja kompilasi yang optimal, jika diset 2 berarti menyediakan kinerja run-time yang lebih baik tetapi menurunkan kinerja kompilasi.

- **PLSQL_WARNING=DISABLE:ALL**

Dengan menyetting DISABLE : ALL disable pesan peringatan dari kompilasi PL/SQL dan memberikan kinerja terbaik.

Latihan

Buat Laporan Tugas untuk soal berikut:

1. Buat sistem database peminjaman buku, lalu berikan beberapa trigger database pada event:
 - penambahan/perubahan data hanya terjadi pada jam kerja (jam 08.00 - 17.00), tampilkan pesan warning.
 - mencatat semua user yang melakukan penambahan/perubahan data pada table buku.
 - pengecekan jika buku yang dipinjam tidak tersedia/sedang dipinjam, tampilkan pesan.
2. Buat trigger yang mengupdate stok buku berupa pesan "Buku tidak tersedia", jika stok buku habis/sedang dipinjam.