6. MENGGUNAKAN SUBQUERY

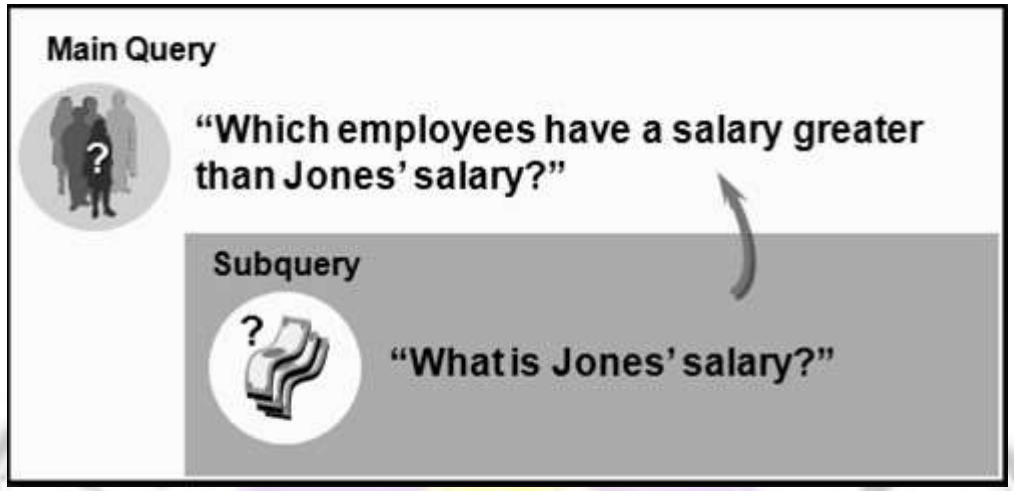
Objektif

Setelah materi ini dipelajari, diharapkan dapat melakukan hal-hal berikut ini:

1. Menggambarkan tipe permasalahan yang dapat dipecahkan oleh *subquery*.
2. Mendefinisikan *subquery*.
3. Membuat *subquery*.
4. Menciptakan *subquery* secara *single-row* dan *multiple-row*.

6.1. Permasalahan Subquery

Pegawai mana yang memiliki penghasilan lebih besar daripada penghasilan Jones ?



Gambar di atas merupakan gambaran sebuah *query* untuk mencari pegawai-pegawai yang memiliki penghasilan lebih besar daripada penghasilan Jones.

Untuk memecahkan masalah di atas, diperlukan dua *query* yaitu :

1. *Query* untuk mencari berapa jumlah penghasilan Jones
2. *Query* untuk mencari penghasilan pegawai-pegawai yang lebih besar dari jumlah penghasilan Jones.

6.2. Subquery

Subquery adalah suatu pernyataan SELECT yang diletakkan di dalam klausa pernyataan SELECT lain. Pernyataan *subquery* sangat bermanfaat ketika *user* memerlukan pemilihan barisbaris data dengan suatu kondisi yang bergantung pada data lain di dalam tabel itu sendiri.

User dapat menempatkan *subquery* di dalam sejumlah klausa-klausa SQL, termasuk :

- Klausa WHERE
- Klausa HAVING
- Klausa FROM

Sintak *Subquery*

```
SELECT    select_list
FROM      table
WHERE     expr operator
                    (SELECT      select_list
                    FROM         table);
```

Lihat Video

Penjelasan sintak :

Operator termasuk suatu kondisi pembanding seperti >, =, atau IN

Catatan : Operator pembanding dibagi ke dalam dua kelas : operator *single-row* (>, =, >=, <, <>, <=) and operator *multiple-row* (IN, ANY, ALL).

Subquery lebih dikenal dengan klausa SELECT bersarang (*nested*), sub-SELECT atau pernyataan SELECT di dalam SELECT. Secara umum, *subquery* akan dieksekusi pertama kali dan hasil keluarannya digunakan untuk melengkapi kondisi-kondisi pada *query* utama (*Main Query*).

6.3. Penggunaan Subquery

```
SQL> SELECT ename
  2  FROM   emp
  3  WHERE  sal >          2975
  4                  (SELECT sal
  5                  FROM   emp
  6                  WHERE  empno=7566);
```

Lihat Video

Pada *statement* di atas, *subquery* (*query* yang di dalam) menetapkan gaji pada baris data pegawai yang memiliki nomor pegawai 7566. *Query* yang di luar (*Main query*) mengambil hasil gaji dari *sub-query* dan digunakan untuk menampilkan semua pegawai yang memiliki gaji lebih besar dari hasil gaji *sub-query* (gaji pada nomor pegawai 7566).

6.4. Petunjuk Menggunakan Subquery

- *Subquery* diapit di dalam tanda kurung.
- Tempatkan *subquery* di sebelah kanan dari operator pembanding.
- Klausa ORDER BY tidak dapat diletakkan pada *statement subquery*. *User* hanya boleh menggunakan klausa ORDER BY untuk pernyataan SELECT yang berada pada *main query* (*query* utama).
- Gunakan operator *single-row* pada *single-row subquery* dan gunakan operator *multiple-row* pada *multiple-row subquery*.

6.5. Tipe Subquery

- *Single-row subquery* : *query* tersebut mengembalikan hasil satu baris dari pernyataan SELECT yang di dalam (SELECT *subquery*).
- *Multiple-row subquery* : *query* tersebut mengembalikan hasil lebih dari satu baris dari pernyataan SELECT yang di dalam (SELECT *subquery*).
- *Multiple-column subquery* : *query* tersebut mengembalikasn hasil lebih dari satu kolom dari pernyataan SELECT yang di dalam (SELECT *subquery*).

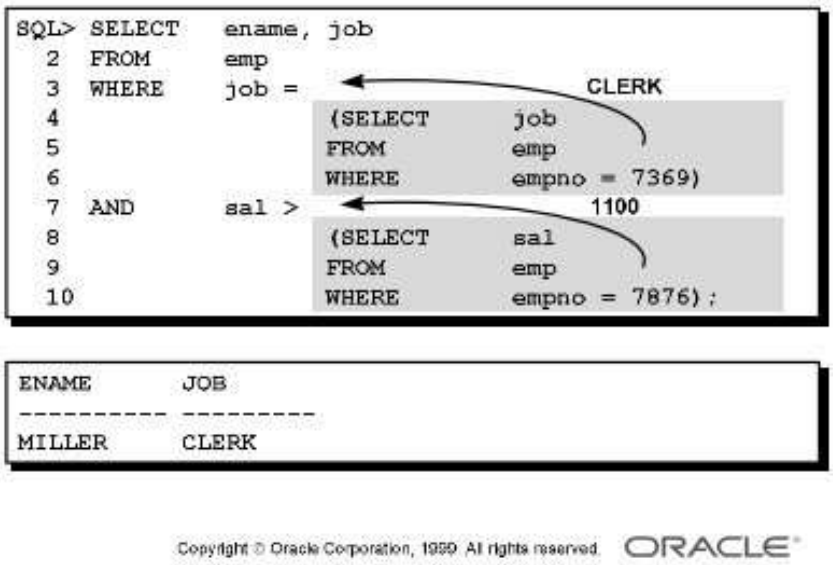
1. Single-Row Subquery

Single-row subquery hanya mengembalikan hasil satu baris dari pernyataan SELECT pada *query* terdalam. *Subquery* ini menggunakan operator-operator pembanding *single-row*. Tabel 1. Daftar Operator *Single-Row*

Operator	Keterangan
=	Sama dengan
>	Lebih besar dari
>=	Lebih besar dari atau sama dengan
<	Lebih kecil dari
<=	Lebih kecil dari atau sama dengan

<>	Tidak sama dengan

Contoh :



Lihat Video

2. Multiple-Row Subquery

Subquery tipe ini mengembalikan nilai lebih dari satu baris yang dinamakan *multiple-row subqueries*. *User* menggunakan operator *multiple-row* untuk menjalankan *multiple-row subquery*. Operator *multiple-row* meminta satu atau lebih nilai.

Operator	Keterangan
IN	**Sama dengan anggota yang ada di dalam daftar.**
ANY	Perbandingan setiap nilai untuk menghasilkan *subquery.*
ALL	Perbandingan semua nilai untuk menghasilkan *subquery*.

Contoh :

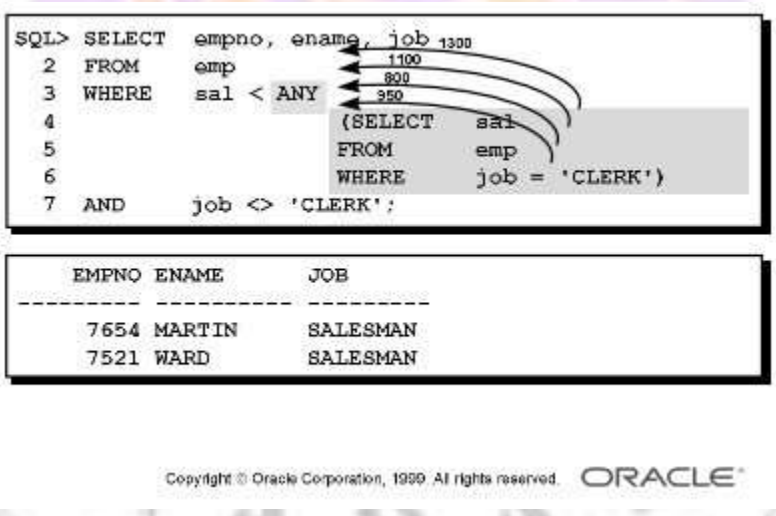
```
SQL> SELECT    ename, sal, deptno
  2 FROM       emp
  3 WHERE      sal IN (SELECT  MIN(sal)
  4                    FROM       emp
                       GROUP BY  deptno);
```

Lihat Video

- Operator ANY (sama seperti operator SOME) membandingkan nilai untuk setiap nilai yang dihasilkan oleh sebuah *subquery*.

  <ANY artinya lebih kecil dari nilai maksimum, >ANY artinya lebih besar dari nilai minimum, -ANY artinya sama dengan IN.

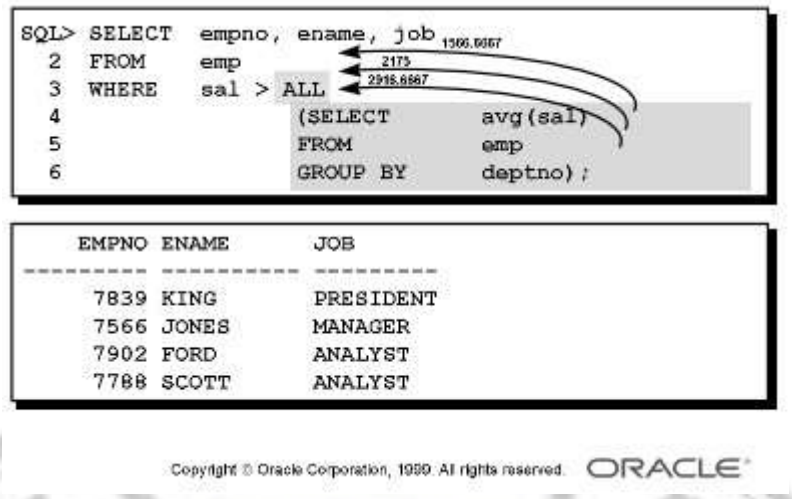
  Contoh :



Lihat Video

- Operator ALL membandingkan nilai dari seluruh nilai yang dihasilkan oleh *subquery*.
  >ALL dimaksudkan lebih dari maksimum dan <ALL dimaksudkan kurang dari minimum.

  Contoh :



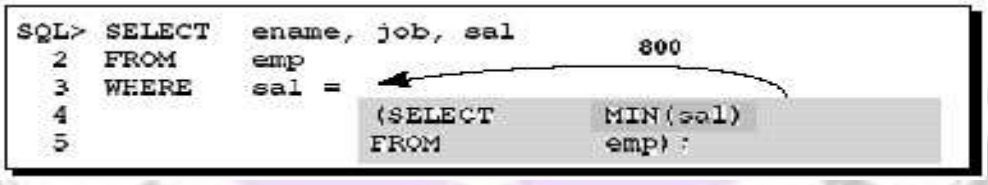
Lihat Video

- Operator NOT dapat digunakan beriringan dengan operator IN, ANY, dan ALL.

6.6. *Group Function* Pada Subquery

User dapat menampilkan data dari *query* utama dengan menggunakan *group function* di dalam *subquery* untuk menghasilkan sebuah baris. *Subquery* berada di dalam tanda kurung dan ditempatkan setelah operator perbandingan.

Contoh :



Lihat Video

6.7. Klausa HAVING Pada Subquery

User dapat meletakkan *subquery* tidak hanya pada klausa WHERE, tetapi bisa juga pada klausa HAVING. Server oracle mengeksekusi *subquery* dan hasilnya dikembalikan ke klausa HAVING pada *query* utama.

Contoh :

```
SQL> SELECT       deptno, MIN(sal)
  2  FROM         emp
  3  GROUP BY     deptno
  4  HAVING       MIN(sal) >
  5                          (SELECT    MIN(sal)
  6                          FROM       emp
  7                          WHERE      deptno = 20);
```

800

Lihat Video

6.8. Kesalahan Pada Subquery

Permasalahan yang umum terjadi pada *subquery* adalah tidak adanya data yang terpilih dari hasil eksekusi *subquery*. Sehingga *subquery* tidak dapat mengembalikan nilai untuk menghasilkan *output* yang dibutuhkan.

Contoh :

```
SQL> SELECT ename, job
  2  FROM   emp
  3  WHERE  job =
  4                (SELECT job
  5                FROM   emp
  6                WHERE  ename='SMYTHE');

no rows selected
```

Subquery returns no values

Kesalahan yang juga umum terjadi pada *subquery* ketika terdapat lebih dari satu baris data dikembalikan untuk suatu *single-row subquery.*

Contoh :

```
SQL> SELECT empno, ename
  2  FROM   emp
  3  WHERE  sal =
  4                (SELECT   MIN(sal)
  5                FROM      emp
  6                GROUP BY  deptno);
```

Single-row operator with multiple-row subquery

```
ERROR:
ORA-01427: single-row subquery returns more than
one row

no rows selected
```

Lihat Video