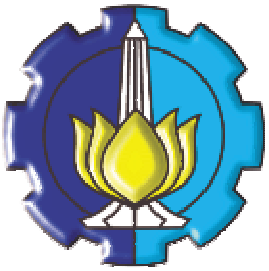
Oracle Academic Initiative

Oracle9i Introduction to SQL



Oleh:

Tessy Badriyah, SKom.MT

Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Surabaya

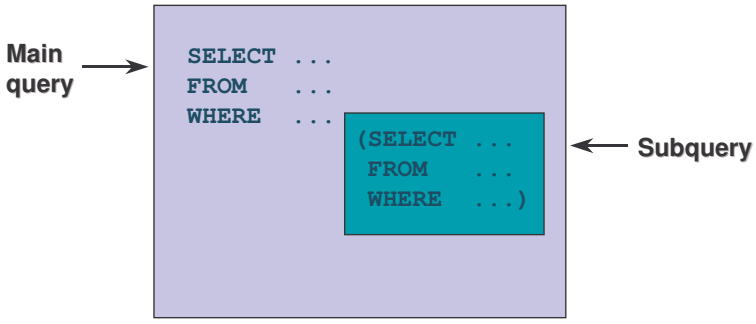
BAB 18 : SubQuery Lanjutan

18.1. Sasaran

- Dapat menulis subquery dengan banyak kolom
- Dapat menggambarkan dan menjelaskan karakteristik dari subqueries pada saat didapatkan nilai NULL
- Dapat menulis subquery dalam klausa FROM
- Dapat menggunakan scalar subqueries dalam SQL
- Dapat menggambarkan tipe dari persoalan yang dapat dipecahkan dengan menggunakan sub query yang berkorelasi.
- Dapat menulis subquery yang berkorelasi.
- Melakukan Update dan Delete baris dengan menggunakan subqueries yang berkorelasi.
- Dapat menggunakan operator EXISTS dan NOT EXISTS
- Dapat menggunakan klausa WITH

18.2. Apa itu SubQuery ?

Apa yang disebut dengan SubQuery ? Subquery adalah statement SELECT yang dilampirkan sebagai klausa dalam SQL Statement yang lain.



18.3. SubQuery

Subquery (inner query) dijalankan sekali sebelum main query. Kemudian hasil dari subquery digunakan oleh main query (outer query).

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

18.4. Penggunaan SubQuery

```
SELECT last_name
FROM   employees
WHERE  salary >
                (SELECT salary
                 FROM   employees
                 WHERE  employee_id = 149) ;
```

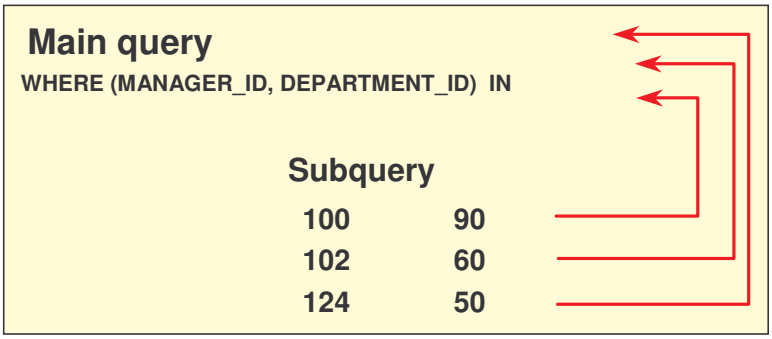
10500

LAST_NAME
King
Kochhar
De Haan
Abel
Hartstein
Higgins

6 rows selected.

18.5. SubQuery Banyak Kolom

Tiap baris dari main query dibandingkan dengan nilai dari subquery multiple-row dan multiple-column.



18.6. Pembandingan Kolom

Pembandingan kolom dalam subquery banyak kolom dapat berupa :

- Pembandingan berpasangan
- Pembandingan tidak berpasangan

18.7. Pairwise Comparison SubQuery

Berikut contoh pembandingan berpasangan untuk menampilkan detail dari data pegawai yang dimanajeri oleh manajer dan department yang sama dengan yang dimiliki oleh nomer pegawai 178 atau 174.

```
SELECT employee_id, manager_id, department_id
FROM   employees
WHERE  (manager_id, department_id) IN
                      (SELECT manager_id, department_id
                       FROM   employees
                       WHERE  employee_id IN (178,174))
AND    employee_id NOT IN (178,174);
```

18.8. NonPairwise Comparison SubQuery

Berikut contoh pembandingan tidak berpasangan untuk menampilkan detail dari data pegawai yang dimanajeri oleh manager yang sama dengan pegawai dengan nomer pegawai 174 atau 141 dan bekerja dalam departement yang sama dengan pegawai yang memiliki nomer pegawai 174 atau 141.

```
SELECT  employee_id, manager_id, department_id
FROM    employees
WHERE   manager_id IN
                    (SELECT  manager_id
                     FROM    employees
                     WHERE   employee_id IN (174,141))
AND     department_id IN
                    (SELECT  department_id
                     FROM    employees
                     WHERE   employee_id IN (174,141))

AND    employee_id NOT IN(174,141);
```

18.9. Penggunaan SubQuery dalam Klausa FROM

```
SELECT  a.last_name, a.salary,
        a.department_id, b.salavg
FROM    employees a, (SELECT   department_id,
                      AVG(salary) salavg
                      FROM     employees
                      GROUP BY department_id) b
WHERE   a.department_id = b.department_id
AND     a.salary > b.salavg;
```

LAST_NAME	SALARY	DEPARTMENT_ID	SALAVG
Hartstein	13000	20	9500
Mourgos	5800	50	3500
Hunold	9000	60	6400
Zlotkey	10500	80	10033.3333
Abel	11000	80	10033.3333
King	24000	90	19333.3333
Higgins	12000	110	10150

7 rows selected.

18.10. Ekspresi Scalar SubQuery

Ekspresi scalar subquery adalah subquery yang mengembalikan hanya satu nilai kolom dari satu baris. Scalar subquery pada Oracle8i hanya terbatas pada :

- SELECT Statement (klausa FROM dan WHERE saja)
- Daftar VALUE dari statement INSERT

Pada Oracle9i, scalar subqueries dapat diguanakan dalam :

- Kondisi dan ekspresi sebagai bagian dari perintah DECODE dan CASE.
- Semua klausa dari SELECT Statement kecuali GROUP BY.

18.11. Contoh Scalar SubQuery

Scalar SubQuery dalam ekspresi CASE :

```
SELECT employee_id, last_name,
       (CASE
        WHEN department_id =
             (SELECT department_id FROM departments
              WHERE location_id = 1800)
        THEN 'Canada' ELSE 'USA' END) location
FROM   employees;
```

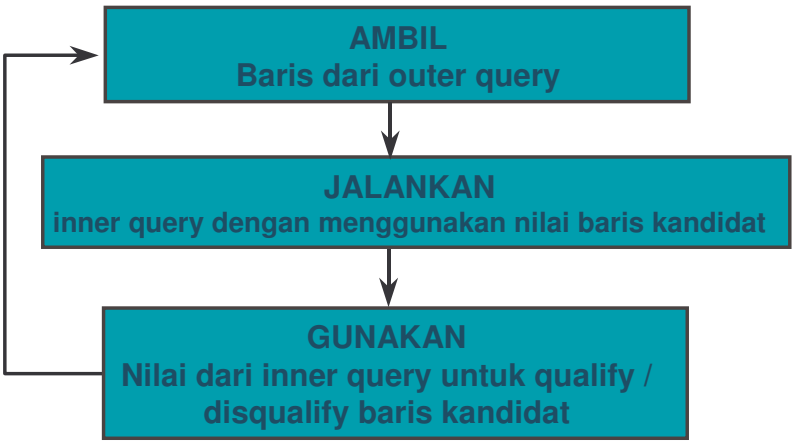
20

Scalar SubQuery dalam klausa ORDER BY :

```
SELECT   employee_id, last_name
FROM     employees e
ORDER BY (SELECT department_name
          FROM departments d
          WHERE e.department_id = d.department_id);
```

18.12. Korelasi SubQuery

Korelasi SubQuery digunakan untuk pemrosesan baris per baris. Tiap-tiap subquery dijalankan sekali untuk setiap baris dari outer query.
Prosesnya sebagai berikut :



Sintak penulisan Korelasi SubQuery :

```
SELECT column1, column2, ...
FROM   table1  outer
WHERE  column1 operator
                    (SELECT  colum1, column2
                     FROM    table2
                     WHERE   expr1 =
                                outer.expr2);
```

Subquery merefer ke kolom dari tabel yang ada pada parent query.

18.13. Penggunaan Korelasi SubQuery

Contoh penggunaan Korelasi SubQuery untuk mencari pegawai yang penghasilannya melebihi rata-rata penghasilan pada departemen tempat mereka bekerja.

```
SELECT last_name, salary, department_id
FROM   employees outer
WHERE  salary >
          (SELECT AVG(salary)
          FROM   employees
          WHERE  department_id =
                 outer.department_id) ;
```

Setiap saat baris dari outer query diproses, maka inner query dievaluasi.

Contoh yang lain dari korelasi subquery adalah untuk menampilkan pegawai yang pernah berganti job sedikitnya dua kali.

```
SELECT e.employee_id, last_name,e.job_id
FROM   employees e
WHERE  2 <= (SELECT COUNT(*)
             FROM   job_history
             WHERE  employee_id = e.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID
101	Kochhar	AD_VP
176	Taylor	SA_REP
200	Whalen	AD_ASST

18.14. Penggunaan Operator EXIST

Operator EXISTS menguji keberadaan dari baris dalam himpunan hasil dari subquery.
Jika ditemukan, maka :

- pencarian tidak dilanjutkan dalam inner query dan kondisi ditandai TRUE.

Jika tidak ditemukan, maka :

- Kondisi ditandai FALSE dan kondisi pencarian dilanjutkan dalam inner query.

Berikut penggunaan operator EXISTS untuk mencari pegawai yang memiliki sedikitnya satu orang bawahan.

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees outer
WHERE  EXISTS ( SELECT 'X'
                FROM   employees
                WHERE  manager_id =
                       outer.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
124	Mourgos	ST_MAN	50
149	Zlotkey	SA_MAN	80
201	Hartstein	MK_MAN	20
205	Higgins	AC_MGR	110

8 rows selected.

18.15. Penggunaan Operator NOT EXIST

Berikut contoh penggunaan operator NOT EXIST untuk menampilkan semua departemen yang tidak mempunyai pegawai.

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                  FROM   employees
                  WHERE  department_id
                         = d.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME
190	Contracting

18.16. Korelasi UPDATE

Korelasi Subquery juga dapat digunakan untuk meng-update baris pada satu table berdasarkan pada baris dari table yang lain.

```
UPDATE table1 alias1
SET    column = (SELECT expression
                 FROM   table2 alias2
                 WHERE  alias1.column =
                        alias2.column);
```

Contoh penggunaan Korelasi UPDATE :

18.17. Korelasi DELETE

Korelasi Subquery juga dapat digunakan untuk menghapus baris pada satu table berdasarkan pada baris dari table yang lain.

```
DELETE FROM table1 alias1
WHERE  column operator
             (SELECT expression
              FROM   table2 alias2
              WHERE  alias1.column = alias2.column);
```

Contoh penggunaan Korelasi DELETE :

```
DELETE FROM employees E
WHERE employee_id =
           (SELECT employee_id
            FROM   emp_history
            WHERE  employee_id = E.employee_id);
```

18.18. Klausa WITH

```
ALTER TABLE employees
ADD(department_name VARCHAR2(14));
```

```
UPDATE employees e
SET    department_name =
              (SELECT department_name
               FROM   departments d
               WHERE  e.department_id = d.department_id);
```

Dengan menggunakan klausa WITH, kita dapat menggunakan blok query yang sama dalam statement SELECT pada saat terjadi lebih dari sekali dalam complex query. Klausa WITH mendapatkan hasil dari blok query dan menyimpannya dalam tablespace temporer kepunyaan user. Klausa WITH dapat meningkatkan performansi.

18.19. Contoh Klausa WITH

Contoh : Gunakan klausa WITH untuk menampilkan nama department dan total gaji untuk tiap departemen yang memiliki total gaji lebih besar dari gaji rata-rata pada sembarang department.

```
WITH
dept_costs  AS (
   SELECT  d.department_name, SUM(e.salary) AS dept_total
   FROM    employees e, departments d
   WHERE   e.department_id = d.department_id
   GROUP BY d.department_name),
avg_cost    AS (
   SELECT SUM(dept_total)/COUNT(*) AS dept_avg
   FROM   dept_costs
SELECT *
FROM   dept_costs
WHERE  dept_total >
        (SELECT dept_avg
         FROM avg_cost)
ORDER BY department_name;
```

18.20. Latihan