

Rosa Ariani Sukamto

Email: rosa_if_itb_01@yahoo.com

Blog: <http://udinrosa.wordpress.com>

Website: <http://www.gangsir.com>

Operator Himpunan, DML, DDL, DCL, TCL, Sequence, Index dan Synonym



Operator Himpunan

- Pengambilan data pada Oracle untuk lebih dari satu tabel juga dapat menggunakan Operator Himpunan (selain Operasi Join)
- Operator Himpunan => UNION, UNION ALL, INTERSECT, MINUS

Operator Himpunan => UNION

- UNION = Penggabungan (Penyatuan Beberapa Tabel menjadi sebuah Tabel)
- Jumlah hasil kolom setiap tabel harus sama
- Tipe data setiap kolom yang dihasilkan dari setiap tabel harus sama
- Menampilkan data yang berbeda saja dari tabel pertama dan kedua

```
SQL> select nim, nama, to_char(nilai) from mhs1  
2 union  
3 select nim, nama, to_char(tglahir) from mhs2;
```

Operator Himpunan => UNION

Tabel 1

A	B	C

Tabel 2

A	B	C

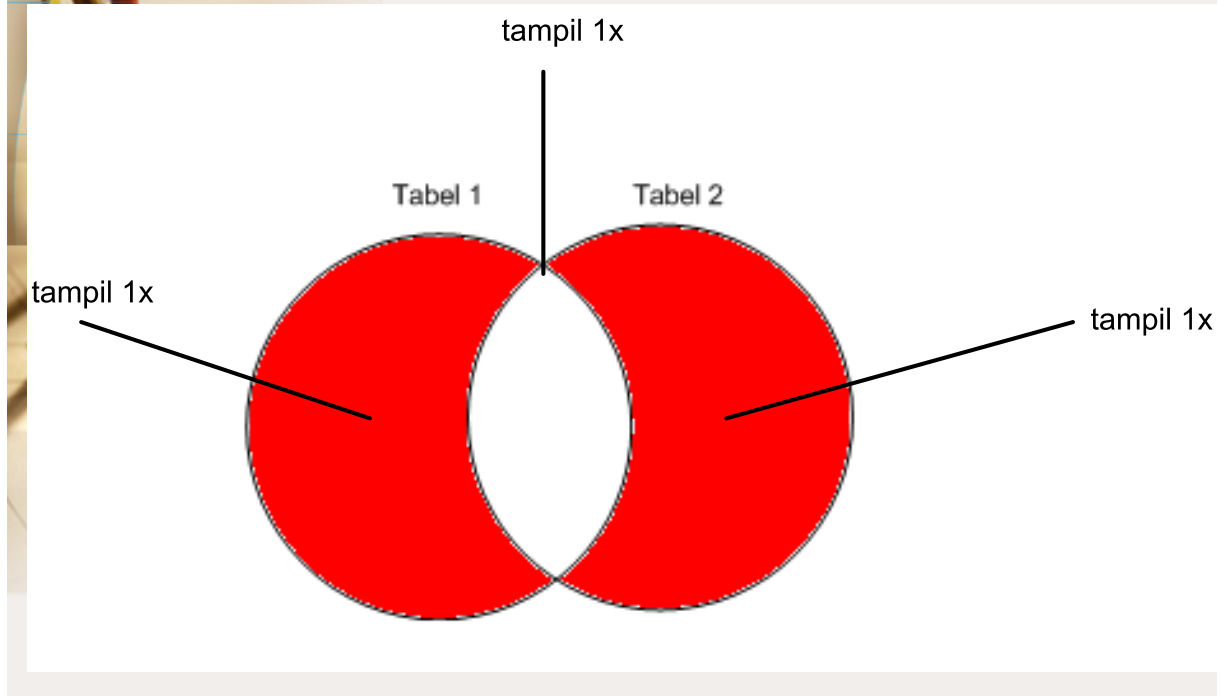
UNION

A	B	C
A	B	C

Tabel 1

Tabel 2

Operator Himpunan => UNION

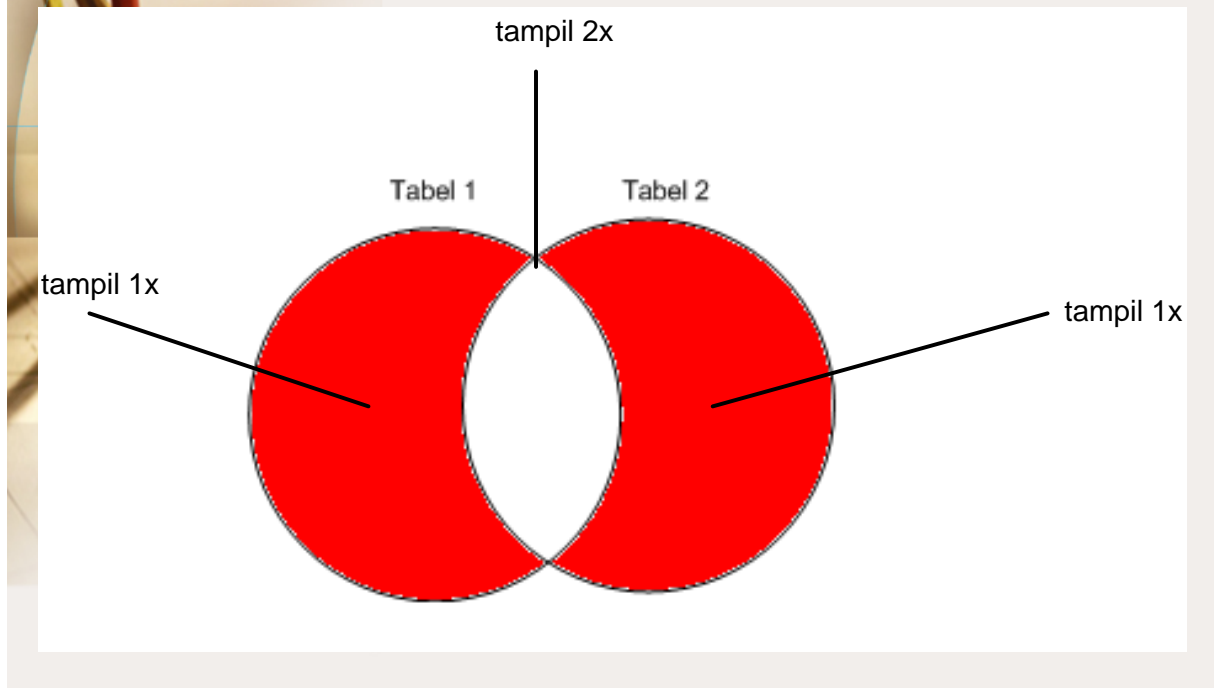


Operator Himpunan => UNION ALL

- Menampilkan semua data walaupun sama tapi dari tabel yang berbeda

```
SELECT location_id FROM locations  
UNION ALL  
SELECT location_id FROM departments;
```

Operator Himpunan => UNION ALL

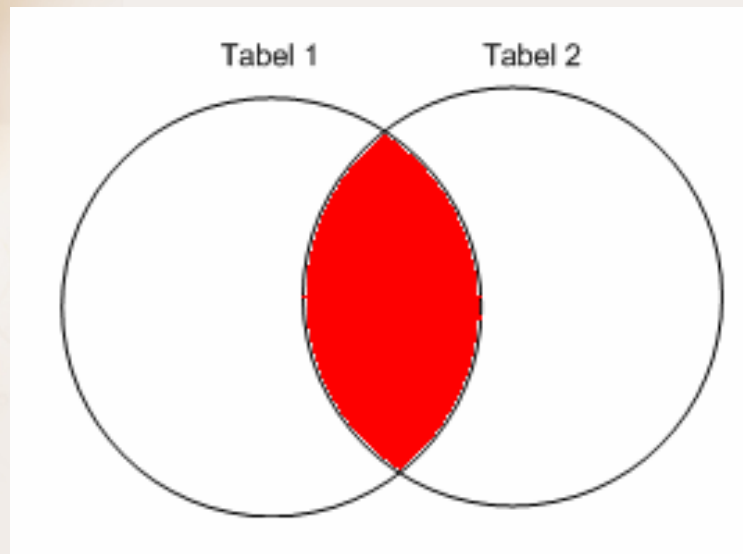


Operator Himpunan => INTERSECT

- Intersect = Irisan
- Hanya menampilkan data yang ada di tabel pertama dan ada di tabel yang kedua

```
select nim, nama from mhs1
intersect
select nim, nama from mhs2;
```

Operator Himpunan => INTERSECT

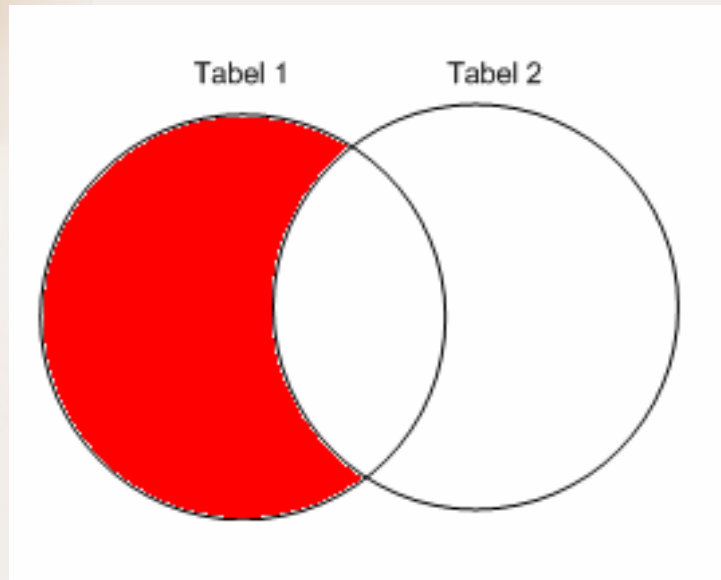


Operator Himpunan => MINUS

- Menampilkan tabel pertama dikurangi tabel yang kedua

```
SQL> select trim(nim), trim(nama) from mhs1
  2  minus
  3  select trim(nim), trim(nama) from mhs2;
TRIM TRIM(NAMA)
-----
01  abcdef
```

Operator Himpunan => MINUS



DML (*Data Manipulation Language*)

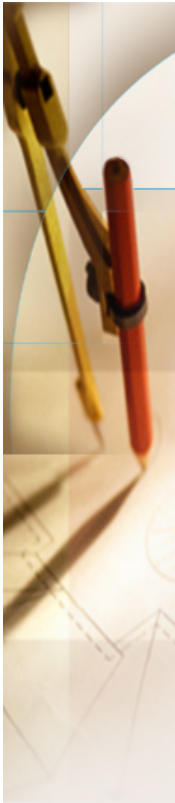
Mengubah isi basis data =>No automatic commit (Explicit)

- INSERT
- UPDATE

```
UPDATE nama_tabel SET  
    nama_kolom1=nilai1,  
    nama_kolom2=nilai2, .... WHERE  
    ....
```

- DELETE

```
DELETE FROM nama_tabel WHERE....
```

DML (*Data Manipulation Language*)

- MERGE
 - gabung tabel untuk sinkronisasi

```
MERGE INTO nama_tabel
USING nama_tabel/view/subquery
ON(kondisi_join)
WHEN MATCHED THEN
    UPDATE SET kolom1=nilai1,
              kolom2=nilai2,.....
WHEN NOT MATCHED THEN
    INSERT(list_kolom)
    VALUES(list_nilai)
```



DML (*Data Manipulation Language*)

```
MERGE INTO copy_emp c
USING employees e
ON(c.employee_id = e.employee_id)
WHEN MATCHED THEN
    UPDATE SET
        c.first_name = e.first_name,
        c.last_name = e.last_name,
        ...
        c.department_id = e.department_id
WHEN NOT MATCHED THEN
    INSERT VALUES(e.employee_id, e.first_name, e.last_name,
                  e.email, e.phone_number, e.hire_date, e.job_id,
                  e.salary, e.commission_pct, e.manager_id,
                  e.department_id);
```



DDL (Data Definition Language)

Commit Otomatis => Automatically commits (Implicit)

- CREATE
- ALTER
- DROP
- TRUNCATE
 - menghapus isi data dalam tabel dan membebaskan memori data yang dihapus

```
TRUNCATE TABLE nama_tabel;
```

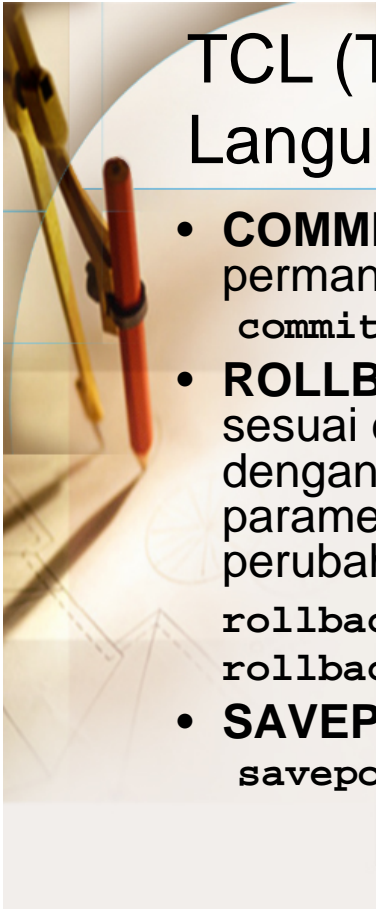
- RENAME
 - mengubah nama

```
RENAME nama_lama TO nama_baru;
```



DCL (Data Control Language)

- GRANT => mengizinkan *user* mengakses basis data
- REVOKE => menarik ijin (*grant*) yang diberikan pada *user*



TCL (Transaction Control Language)

- **COMMIT** => menandai perubahan secara permanen pada data
`commit;`
- **ROLLBACK** => mengembalikan keadaan sesuai dengan titik (keadaan) yang ditandai dengan SAVEPOINT atau jika tanpa parameter akan kembali pada titik perubahan yang terakhir
`rollback;`
`rollback to update_done;`
- **SAVEPOINT** => membuat titik kondisi
`savepoint update_done;`



Sequence

- Sequence adalah objek basis data yang dapat digunakan untuk secara otomatis membangkitkan nilai urut.

```
CREATE SEQUENCE sequence  
[INCREMENT BY n]  
[START WITH n]  
[ {MAXVALUE n | NOMAXVALUE} ]  
[ {MINVALUE n | NOMINVALUE} ]  
[ {CYCLE | NOCYCLE} ]  
[ {CACHE n | NOCACHE} ]
```



Sequence

- **sequence**
 - nama sequence yang akan diciptakan.
- **INCREMENT BY n**
 - menentukan interval nilai sequence. Bila tidak dituliskan maka dianggap sequence memiliki interval 1.
- **START WITH n**
 - menentukan nilai awal sequence. Bila tidak dituliskan maka dianggap sequence diawali dengan nilai 1.



Sequence

- **MAXVALUE n**
 - menentukan nilai maksimum sequence.
- **NOMAXVALUE**
 - menentukan nilai maksimum sequence 1027. Ini nilai default.
- **MINVALUE n**
 - menentukan nilai minimum sequence.
- **NOMINVALUE**
 - menentukan nilai minimum sequence 1.
- **CYCLE | NOCYCLE**
 - menentukan bahwa setelah sequence mencapai nilai maksimumnya maka akan berputar kembali ke nilai awalnya.
- **CACHE n | NOCACHE**
 - menentukan berapa jumlah nilai yang dialokasikan sebelumnya dan diletakkan dalam memori. Defaultnya adalah 20 nilai.



Contoh Sequence

- Menciptakan sequence DEPARTMENT_ID untuk kolom DEPT_ID pada tabel DEPARTMENT. Sequence diawali dengan nilai 51. Tidak mengizinkan caching dan cycle.

```
CREATE SEQUENCE department_id  
INCREMENT BY 1  
START WITH 51  
MAXVALUE 9999999  
NOCACHE  
NOCYCLE;
```



Menampilkan Sequence

- Menampilkan informasi dari seluruh sequence telah kita ciptakan.

```
SELECT sequence_name,  
min_value, max_value,  
increment_by, last_number  
FROM user_sequences;
```



Menggunakan Sequence

- Menambahkan data baru pada tabel DEPARTMENT

```
INSERT INTO department  
(id, name, region_id)  
VALUES  
(department_id.NEXTVAL, 'Finance', 2);
```

- Menampilkan nilai terakhir dari sequence DEPARTMENT_ID

```
SELECT department_id.CURRVAL  
FROM SYS.dual;
```



Menggunakan Sequence

- NEXTVAL digunakan untuk mengambil nilai sequence berikutnya.
 - Kita harus menyebutkan nama sequencenya `sequence.NEXTVAL`.
 - Ketika kita memanggil NEXTVAL, maka nilai sequence yang baru akan dibangkitkan, dan nilai sequence saat ini akan diletakkan pada CURRVAL.
- CURRVAL digunakan untuk mengetahui nilai sequence yang baru saja dibangkitkan.
 - CURRVAL menunjukkan nilai terakhir yang telah dicapai oleh sequence



Ubah dan Hapus Sequence

```
ALTER SEQUENCE nama_sequence  
  [INCREMENT BY n]  
  [{MAXVALUE n | NOMAXVALUE}]  
  [{MINVALUE n | NOMINVALUE}]  
  [{CYCLE | NOCYCLE}]  
  [{CACHE n | NOCACHE}]
```

```
DROP SEQUENCE nama_sequence
```



Index

- Index adalah sebuah objek sistem basis data yang dapat mempercepat proses pencarian (query) data
 - Unique index => diciptakan secara otomatis ketika kita mendefinisikan kolom yang memiliki constraint PRIMARY KEY atau UNIQUE
 - Non-unique index => tipe index yang tidak mengharuskan nilai-nilai suatu kolom yang memiliki index ini bernilai unik (misal FOREIGN KEY)



Membuat Index

```
CREATE [UNIQUE] INDEX nama_index  
ON table ( column [ , column . . . ] );
```

- dimana:
 - **UNIQUE**: pilihan untuk menciptakan unique index
 - **nama_index**: nama index yang akan diciptakan
 - **table**: nama tabel dari kolom yang akan diindex
 - **column**: nama kolom dari tabel yang akan diindex



Menampilkan Index

untuk menampilkan index:

```
SELECT index_name, table_name, uniqueness  
FROM user_indexes  
WHERE table_name = '<TABLE>'
```

- misal:
 - Menampilkan seluruh index yang telah dibuat untuk tabel EMPLOYEE.

```
SELECT ic.index_name, ic.column_name,  
       ic.column_position col_pos, ix.uniqueness  
FROM user_indexes ix, user_ind_columns ic  
WHERE ic.index_name = ix.index_name  
AND ic.table_name = 'EMPLOYEE';
```


Mengubah dan Menghapus Index

- Index tidak dapat diubah jadi jika ingin mengubah, hapus dulu lalu buat lagi

menghapus Index:

```
DROP INDEX <nama_index>;
```

misal

```
DROP INDEX employee_last_name_idx;
```

Synonym

- Synonym adalah nama alternatif untuk table, view, sequence, procedure, stored function, package, snapshot, atau synonym lain

```
CREATE [PUBLIC] SYNONYM nama_synonym  
FOR object;
```

dimana

- PUBLIC
 - Menciptakan synonym yang dapat diakses oleh seluruh user.
- nama_synonym
 - Nama synonym yang akan diciptakan.
- object
 - Nama object yang akan diwakili oleh synonym tersebut.

Synonym

- Harus memiliki GRANT dari user lain yang akan diakses tabelnya

```
GRANT select ON nama_tabel TO nama_user;
```

- Untuk melihat tabel user lain tidak perlu menuliskan nama_user.nama_tabel

```
CREATE SYNONYM nama_tabel_baru FOR  
nama_user_lain.nama_tabel;
```

Menampilkan:

```
SELECT * FROM nama_tabel_baru;
```

Menghapus:

```
DROP [PUBLIC] SYNONYM nama_synonym;
```