

Ketepatan Menggunakan Index, View, Sequence, Trigger dan Undo



Workshop Administrasi Basis Data

Yunia Ikawati

Teknik Informatika-PENS



Overview

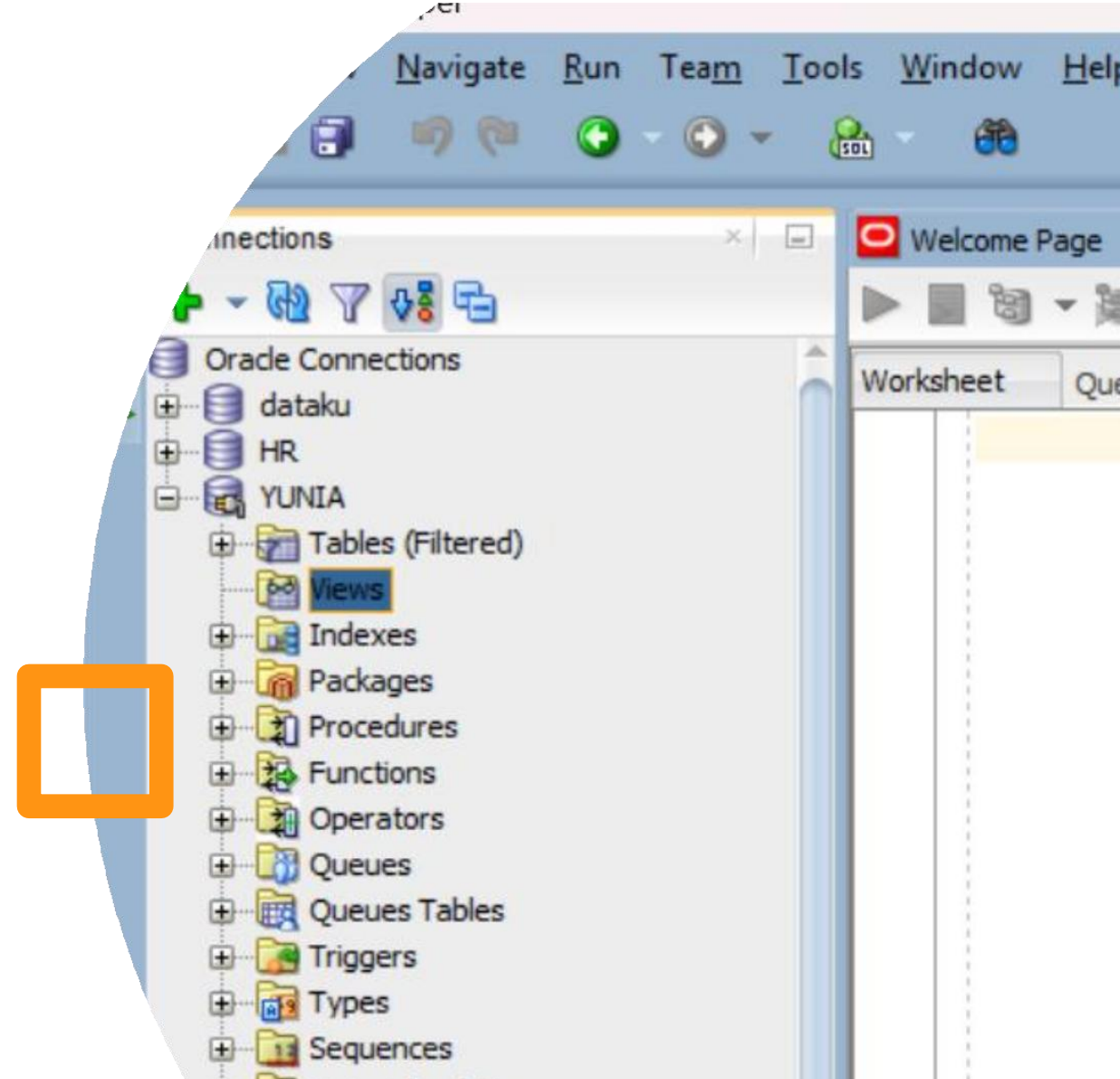
Menggunakan
Index, View,
Sequence

Menggunakan
Trigger

Menggunakan
Undo



Mengelola Objek



Membuat Index

- Indeks digunakan untuk meningkatkan performa query dengan mempercepat pencarian data.
- Gunakan indeks pada kolom yang sering digunakan dalam kondisi WHERE, ORDER BY, atau JOIN.
- Gunakan hanya untuk kolom yang sering diakses melalui filter.
- Hindari membuat terlalu banyak indeks karena dapat memperlambat operasi INSERT/UPDATE/DELETE.

Membuat Index

Perintah membuat index:

```
CREATE INDEX index_nama ON table_name (column);
```

Melihat informasi index:

```
SELECT index_name, table_name, uniqueness  
FROM user_indexes;
```



Contoh:

MAHASISWA

NIM	VARCHAR2(10)
NAMA	VARCHAR2(100)
PRODI	VARCHAR2(50)
ANGKATAN	NUMBER(4)

```
CREATE INDEX idx_mahasiswa_nim  
ON mahasiswa (nim);  
``
```



Membuat View

- View adalah query yang disimpan sehingga memudahkan akses ke data tanpa harus menulis ulang query.
- Pastikan view digunakan untuk menyederhanakan akses data yang sering dipakai.
- Gunakan untuk menyederhanakan akses data yang rumit.
- Gunakan view dengan WITH READ ONLY jika tidak ingin data dalam view bisa diubah.

Membuat View

- Perintah View:

```
CREATE VIEW nama_view AS  
SELECT kolom1, kolom2  
FROM nama_tabel  
WHERE kondisi;
```



Contoh:

MAHASISWA

NIM VARCHAR2(10)
NAMA VARCHAR2(100)
PRODI VARCHAR2(50)
ANGKATAN NUMBER(4)

```
CREATE VIEW vw_mahasiswa_2022 AS  
SELECT nim, nama, prodi, angkatan  
FROM mahasiswa  
WHERE angkatan = 2022;
```

Membuat Sequence

- Sequence adalah objek yang menghasilkan angka unik untuk digunakan sebagai nilai, misalnya pada kolom Primary Key.
- Cocok digunakan untuk memastikan angka unik dalam aplikasi.



Membuat Sequence

- Perintah Sequence:

```
CREATE SEQUENCE nama_sequence  
START WITH nilai_awal  
INCREMENT BY nilai_increment  
[MINVALUE nilai_min]  
[MAXVALUE nilai_max]  
[CYCLE | NOCYCLE]  
[CACHE jumlah_cache | NOCACHE];
```



Contoh:

```
CREATE SEQUENCE seq_mahasiswa  
START WITH 1  
INCREMENT BY 1;
```

Membuat Trigger

- Trigger dalam Oracle adalah blok PL/SQL yang dieksekusi secara otomatis saat terjadi peristiwa tertentu pada tabel atau view, seperti operasi INSERT, UPDATE, atau DELETE.
- Perintah Trigger:

```
CREATE OR REPLACE TRIGGER nama_trigger  
[TIMING] [EVENT] ON nama_tabel  
[FOR EACH ROW]  
BEGIN  
    --Blok kode yang akan dijalankan  
END;  
/
```

1. **CREATE OR REPLACE TRIGGER** : Membuat trigger baru atau menggantikan yang sudah ada dengan nama yang sama.
2. **TIMING** : Menentukan kapan trigger akan dijalankan:
 - **BEFORE** : Sebelum operasi SQL (INSERT, UPDATE, DELETE).
 - **AFTER** : Setelah operasi SQL (INSERT, UPDATE, DELETE).
3. **EVENT** : Jenis operasi yang memicu trigger:
 - **INSERT**, **UPDATE**, **DELETE**.
 - Bisa lebih dari satu operasi, misalnya: **BEFORE INSERT OR UPDATE**.
4. **ON nama_tabel** : Nama tabel yang dihubungkan dengan trigger.
5. **FOR EACH ROW** : Digunakan untuk trigger tingkat baris (row-level trigger).
6. **Blok Kode PL/SQL**: Berisi logika yang akan dieksekusi.



Jenis Trigger

Jenis Trigger	Fungsi Utama
BEFORE	Digunakan sebelum operasi SQL (validasi atau pre-processing).
AFTER	Digunakan setelah operasi SQL (logging atau audit trail).
INSERT	Untuk menangani operasi INSERT.
UPDATE	Untuk menangani operasi UPDATE.
DELETE	Untuk menangani operasi DELETE.
Row-Level Trigger	Untuk setiap baris yang diproses.
Statement-Level Trigger	Untuk seluruh pernyataan SQL.
INSTEAD OF Trigger	Untuk menangani modifikasi pada view yang tidak dapat diubah.



Contoh:

```
CREATE OR REPLACE TRIGGER trg_mahasiswa_bi  
BEFORE INSERT ON mahasiswa  
FOR EACH ROW  
BEGIN  
    :NEW.id_mhs := seq_mahasiswa.NEXTVAL;  
END;  
/
```



Membuat Undo dengan Undo Advisor

- Membatalkan transaksi yang belum dikomit (rollback).
- Mendukung konsistensi pembacaan data dengan menyediakan tampilan data sebelum perubahan dilakukan.
- **Undo Advisor** adalah alat di Oracle yang membantu mengoptimalkan pengelolaan Undo Tablespace dengan memberikan rekomendasi mengenai ukuran Undo Tablespace dan waktu retensi (retention).



LATIHAN

- Silahkan buat study kasus contoh penggunaan masing-masing baik itu Index, View, Sequence, Trigger dan Undo Advisor.
- Menggunakan Oracle SQL Developer.
- Jelaskan penerapan contohnya dan buatlah laporannya.