

ORACLE PL/SQL · PERTEMUAN 13 / 14

Trigger, Sequence, View & Package

Objek pendukung penjualan: nomor, laporan, audit, dan antarmuka modular.

Instalasi & SQL → Mini Project → PL/SQL → Final Project · Oracle XE lokal atau FreeSQL dengan login

Versi PDF materi pertemuan 13. Contoh dan tabel dapat dibaca di PDF. Kuis dapat dijalankan di halaman web: [Buka materi daring](#).

Tujuan pembelajaran

1. Membuat seq_penjualan_demo mulai 200 dan menjelaskan NEXTVAL/CURRVAL per sesi.
2. Membedakan view sederhana yang dapat diperbarui dengan view agregasi read-only.
3. Membuat trigger BEFORE/AFTER untuk INSERT/UPDATE/DELETE dengan :OLD/:NEW.
4. Merekam audit stok tanpa COMMIT dan membuktikan rollback audit.
5. Menyusun specification/body pkg_latihan_demo dan memeriksa USER_ERRORS.

Pokok bahasan / praktikum

- CREATE SEQUENCE START WITH 200
- NEXTVAL CURRVAL dan sesi
- Simple VIEW dan WITH CHECK OPTION
- Complex VIEW SUM GROUP BY dan update restrictions
- DML trigger BEFORE AFTER INSERT UPDATE DELETE
- Row trigger :OLD :NEW
- audit_stok_demo trg_audit_stok_demo tanpa COMMIT
- Package specification dan body pkg_latihan_demo
- USER_ERRORS dan object status

Prasyarat: Lima tabel dan seed modul 02/03, procedure/function modul 12 serta transaksi modul 11. Buat objek modul ini sebelum uji DML; modul 14 memakai sequence, aggregate view, audit table, dan trigger berikut.

Pilih salah satu lingkungan praktikum

Oracle XE: XE lokal dengan SQL Developer: cocok untuk instalasi, koneksi, user/schema, privilege, tablespace serta latihan SQL/PLSQL.

FreeSQL: FreeSQL dengan login: masuk menggunakan akun sendiri dan gunakan schema latihan yang diizinkan untuk SQL/PLSQL. Tidak perlu memasang XE untuk menyelesaikan latihan non-administratif.

Pengumpulan tugas menerima hasil dari salah satu lingkungan. Pastikan objek prasyarat tersedia dan tetap gunakan schema latihan, bukan schema produksi.

AMATI → PREDIKSI → JELASKAN

Mulai dari pertanyaan

1. Apakah nomor sequence harus kembali setelah transaksi gagal?
2. Mengapa jumlah agregat pada view tidak bisa langsung di-UPDATE?
3. Apakah audit yang dibuat trigger ikut dibatalkan bersama transaksi utama?

Uji pemahaman awal

1. Sesi baru belum memanggil NEXTVAL. Apa akibat membaca CURRVAL?

- Selalu 200
- Membaca nomor sesi lain
- Error karena CURRVAL belum didefinisikan di sesi itu
- Mengembalikan NULL

2. Mengapa v_total_penjualan_demo dibuat read-only?

- View selalu tidak bisa diubah
- Total agregat tidak dipetakan langsung ke satu baris dasar
- LEFT JOIN menghapus semua baris
- Sequence melarang UPDATE

3. Apa akibat ROLLBACK transaksi yang memicu audit biasa?

- Audit tetap permanen
- Trigger melakukan COMMIT otomatis
- Sequence kembali ke awal
- Data utama dan audit sama-sama dibatalkan

PAHAMI KONSEP · BACA TERSTRUKTUR · TERAPKAN

Teori dan pembahasan

Empat objek ini saling melengkapi: sequence untuk nomor, view untuk laporan, trigger untuk respons DML, dan package untuk mengorganisasikan API.

Konsep inti bab ini

- Sequence tidak menjamin nomor tanpa celah; rollback tidak mengembalikan NEXTVAL .
- Audit biasa ikut transaksi utama, bukan menyimpan sendiri.
- Package specification adalah antarmuka; body adalah implementasi.

Urutan pembahasan

- 01 Objek baru tanpa menghapus setup
- 02 Sequence dan nomor yang tidak menjanjikan urutan rapat
- 03 View sederhana dan pembaruan terkontrol
- 04 View laporan agregat penjualan
- 05 BEFORE, AFTER, dan pseudorecord
- 06 Audit stok ikut transaksi utama
- 07 Package sebagai antarmuka dan implementasi

Gambaran konsep dalam satu tabel

Objek	Peran	Batas penting
Sequence	Menghasilkan nomor	<code>CURRVAL</code> perlu <code>NEXTVAL</code> pada sesi yang sama
View	Menyimpan definisi query	Agregasi/read-only membatasi update
Trigger	Respons otomatis DML	Gunakan <code>:OLD/:NEW</code> ; jangan <code>COMMIT</code> di trigger biasa
Package	API dan implementasi terkelompok	Specification dahulu, body kemudian

01 Objek baru tanpa menghapus setup

Gunakan Oracle XE 21c lokal melalui SQL Developer ATAU Oracle FreeSQL setelah login pada schema sendiri yang diizinkan; bukan dua kewajiban.

Selesaikan setup lima tabel dan seed modul 02/03 sekali, tanpa menghapus data. Aktifkan panel DBMS Output pada koneksi lokal atau output editor FreeSQL. SQL memakai titik koma; slash pada baris sendiri adalah delimiter Run Script klien, bukan sintaks server. Jika editor mengirim satu unit PL/SQL, kirim blok tanpa slash. Seluruh latihan menggunakan objek sendiri, tanpa DBA atau langkah administrasi.

POIN PEMBAHASAN

- Modul ini menambah `seq_penjualan_demo`, `v_total_penjualan_demo`, `audit_stok_demo`, `trg_audit_stok_demo`, dan package latihan yang independen.
- Periksa `USER_OBJECTS` terlebih dahulu.
- `CREATE SEQUENCE` dan `CREATE TABLE` dijalankan sekali bila objek belum ada; jangan menghapus objek yang sudah dipakai demi mengulang latihan.
- `CREATE OR REPLACE` aman untuk mengganti definisi tertentu tetapi tetap merupakan DDL dengan konsekuensi commit.

- Selesaikan semua DDL sebelum menyentuh uji stok sementara.

02 Sequence dan nomor yang tidak menjanjikan urutan rapat

Sequence menghasilkan bilangan lintas pemanggilan tanpa pola berbahaya `MAX(id)+1`. `seq_penjualan_demo` dimulai pada 200, meningkat satu, dan tidak berputar.

`NEXTVAL` mengambil nomor baru; `CURRVAL` membaca nomor terakhir yang diambil sesi tersebut. Sesi baru yang belum memakai `NEXTVAL` tidak bisa langsung meminta `CURRVAL` meskipun sesi lain pernah mengambil nomor.

POIN PEMBAHASAN

- `ROLLBACK` tidak mengembalikan nomor sequence.
- Menguji `NEXTVAL` sebelum proyek berarti penjualan pertama proyek mungkin 201 atau lebih, bukan pasti 200.
- Celah juga dapat muncul akibat transaksi gagal, cache, atau sesi lain.
- `NOCACHE` dipakai untuk transparansi praktikum, bukan jaminan tanpa celah atau rekomendasi kinerja produksi.
- Periksa kapasitas ID `NUMBER(8)`; sequence latihan diberi `MAXVALUE` yang sesuai.
- Jangan mengandalkan `CURRVAL` sebagai nomor milik transaksi orang lain.

03 View sederhana dan pembaruan terkontrol

View menyimpan definisi query, bukan salinan permanen hasil.

`v_produk_tersedia_demo` memilih satu tabel tanpa agregasi sehingga `UPDATE` kolom stok dapat diteruskan ke tabel produk.

`WITH CHECK OPTION` menjaga hasil pembaruan tetap memenuhi filter stok > 0; mengubah stok menjadi nol melalui view ini ditolak. Latihan mengubah stok positif lalu rollback, sehingga data awal tetap utuh.

POIN PEMBAHASAN

- Kemampuan update bukan ditentukan nama sederhana atau kompleks saja.

- `GROUP BY`, aggregate, `DISTINCT`, ekspresi tertentu, serta aturan key-preserved table pada join memengaruhi pembaruan.
- Periksa `USER_UPDATABLE_COLUMNS` bila diperlukan.
- Jangan menganggap setiap join view tidak bisa diperbarui atau setiap single-table view pasti menerima semua `INSERT`; kolom wajib yang tidak tersedia tetap dapat menyebabkan kegagalan.

04 View laporan agregat penjualan

`v_total_penjualan_demo` menggabungkan header penjualan dan detail dengan `LEFT JOIN`, kemudian `SUM(jumlah*harga)` dikelompokkan per penjualan.

`LEFT JOIN` mempertahankan header tanpa detail, dan `NVL` mengubah hasil `SUM NULL` menjadi nol. ID pelanggan anonim pada penjualan 103 tetap `NULL`; tidak dikonversi menjadi pelanggan fiktif.

POIN PEMBAHASAN

- View ditetapkan `WITH READ ONLY` karena subtotal agregasi tidak memiliki satu baris dasar yang dapat diubah langsung.
- Pada seed, total 101, 102, dan 103 diprediksi 300000, 150000, serta 60000.
- Mengubah laporan berarti mengubah detail melalui prosedur bisnis, bukan `UPDATE` total hasil `SUM`.
- Harga historis detail dipakai agar perubahan harga katalog tidak menghitung ulang transaksi lama.

05 BEFORE, AFTER, dan pseudorecord

Trigger DML berjalan otomatis saat `INSERT`, `UPDATE`, atau `DELETE`.

`FOR EACH ROW` berarti satu pemanggilan untuk tiap baris terdampak; tanpa itu trigger tingkat statement tidak mempunyai akses baris `:OLD/NEW`. Pada `INSERT`, nilai baru bermakna; pada `DELETE` nilai lama bermakna; `UPDATE` memiliki keduanya. Di badan PL/SQL gunakan titik dua pada `:OLD` dan `:NEW`.

POIN PEMBAHASAN

- BEFORE row trigger dapat memvalidasi atau mengubah :NEW sebelum penulisan; AFTER dipilih untuk mencatat hasil perubahan.
- Contoh BEFORE menolak stok pecahan dan harga tidak positif pada produk baru atau yang diperbarui.
- Batas maksimal dua desimal diterapkan pada input uang fungsi/procedure: nilai kolom `NUMBER(12,2)` mungkin sudah dibulatkan saat trigger melihatnya, sehingga trigger ini bukan pengganti validasi parameter uang mentah.
- DELETE cukup mencetak pesan pada contoh BEFORE, bukan menjalankan query ke tabel yang sedang berubah.

06 Audit stok ikut transaksi utama

`audit_stok_demo` menyimpan produk, jenis operasi, stok lama/baru, waktu, dan pengguna schema.

Trigger AFTER merekam `INSERT` dan `DELETE`, serta `UPDATE` hanya jika nilai stok berubah. Tabel audit sengaja tidak memiliki foreign key ke produk agar riwayat `DELETE` tetap bisa tersimpan. Tidak ada `COMMIT`, `ROLLBACK`, atau autonomous transaction di trigger.

POIN PEMBAHASAN

- Jika pemanggil rollback, perubahan produk dan baris audit sama-sama hilang.
- Jika trigger gagal, statement pemicu ikut gagal; exception tidak boleh ditelan hanya agar audit terlihat sukses.
- Hindari `SELECT` dari `produk_demo` di row trigger ini karena tabel sedang mengalami perubahan dan dapat terkena mutating-table error.
- Gunakan pseudorecord untuk nilai baris.
- Audit praktikum ini bukan arsip tahan manipulasi dan bukan bukti sistem produksi yang aman.

07 Package sebagai antarmuka dan implementasi

Package specification mendeklarasikan anggota publik; body berisi implementasinya. `pkg_latihan_demo` hanya menyediakan function `label_produk` serta procedure `tampil_produk`, sehingga tidak bertabrakan dengan `pkg_penjualan_demo` proyek akhir.

Pemanggil memakai `nama_package.nama_anggota`. Function `label_produk` mengembalikan teks, sedangkan `tampil_produk` membaca cursor FOR dan mencetak laporan tanpa mengubah tabel.

POIN PEMBAHASAN

- Nama dan signature di body harus sesuai specification.
- Buat specification dahulu, lalu body.
- Periksa `USER_ERRORS` untuk PACKAGE, PACKAGE BODY, dan TRIGGER; query status VALID melengkapi pemeriksaan.
- Perubahan specification dapat memengaruhi objek dependen.
- Catat hasil kompilasi sebenarnya pada XE lokal atau FreeSQL sendiri; materi ini menyediakan kode yang perlu diuji, bukan klaim sudah menjalankan kompilasi Oracle.

Alur menyiapkan objek pendukung

- 1 Inventaris objek yang sudah ada
- 2 Buat sequence dan view
- 3 Buat audit/trigger
- 4 Buat package specification dan body
- 5 Periksa `USER_ERRORS` lalu uji rollback

Gunakan alur ini sebagai panduan saat mengikuti praktikum; bukti hasil tetap berasal dari eksekusi Oracle Anda.

AMATI DATA · IKUTI LANGKAH · PAHAM HASIL

Contoh dan tabel pembahasan

Contoh berikut dibaca langsung, tanpa parameter atau tombol simulasi. Nilai adalah **prediksi ilustratif dari data acuan**, bukan hasil eksekusi pada akun Oracle Anda.

Rollback stok/audit berbeda dari sequence

Skenario ilustratif: sequence baru START WITH 200, belum dipakai sesi lain; stok Kopi awal 20 dan belum ada audit dari transaksi contoh.

Tahap	Nomor terakhir yang diambil	Stok Kopi	Audit tambahan transaksi
Awal	Belum mengambil <code>NEXTVAL</code>	20	0
<code>NEXTVAL</code> lalu <code>UPDATE</code> stok	200	15	1
<code>ROLLBACK DML</code>	200	20	0

Cara membaca contoh

1. **NEXTVAL** mengambil sebuah nomor. Pada contoh terisolasi, pengambilan berikutnya menghasilkan 201.
2. **UPDATE** stok 20 menjadi 15 memicu audit biasa dengan OLD=20 dan NEW=15.
3. **ROLLBACK** membatalkan perubahan stok dan audit yang belum committed, tetapi tidak mengembalikan nomor sequence.

Yang perlu dipahami: ID sequence dapat berlubang; jangan menggunakan kerapatan nomor sebagai bukti jumlah transaksi. Nilai sequence aktual bergantung pemakaian sebelumnya dan sesi lain.

Video pembelajaran

Video utama — fokus bab ini

Pembuatan trigger Oracle dengan SQL Developer. Tiga referensi tambahan melengkapi sequence, view dan package specification/body.

How to Create Oracle Trigger Using Oracle SQL Developer

Simple Oracle DB · [Buka video utama di YouTube](#)

Video tambahan sesuai subtopik

1. [SQL tutorial 61: SEQUENCE in Oracle Database By Manish Sharma RebellionRider](#)
Manish Sharma

Fokus: Oracle sequence: pembuatan dan penggunaan nomor untuk primary key.

2. [Indexes and Views in Oracle SQL | Oracle Tutorial | SQL Tutorial | SQL Developer | #oracle #sql](#)

Raj Cloud Technologies Online Job-Based Training

Fokus: Views dalam Oracle SQL.

Topik index pada video bersifat tambahan; fokuskan perhatian pada bagian view.

3. [PL/SQL Package with Example | Package Spec and Body](#)

Explore IT

Fokus: Package specification dan package body Oracle PL/SQL dengan contoh.

Batas cakupan dan panduan menonton: Video utama tidak mewakili seluruh bab13. NEXTVAL/CURRVAL, audit :OLD/:NEW, pembatasan view dan package dipelajari bersama kode Oracle pada halaman.

Video merupakan pendamping, bukan pengganti keseluruhan bab. Pemeriksaan relevansi memakai transkrip yang tersedia, deskripsi pembuat dan bab waktu; bukan klaim seluruh durasi telah ditonton. Versi/editor atau bahasa dapat berbeda. Praktikum tetap mengikuti SQL/PLSQL Oracle pada halaman dan pilihan XE atau FreeSQL login. Jika pemutar dibatasi oleh YouTube, gunakan tautan Buka video.

Praktikum Oracle XE atau FreeSQL

Pilih salah satu lingkungan praktikum

Oracle XE: XE lokal dengan SQL Developer: cocok untuk instalasi, koneksi, user/schema, privilege, tablespace serta latihan SQL/PLSQL.

FreeSQL: FreeSQL dengan login: masuk menggunakan akun sendiri dan gunakan schema latihan yang diizinkan untuk SQL/PLSQL. Tidak perlu memasang XE untuk menyelesaikan latihan non-administratif.

Pengumpulan tugas menerima hasil dari salah satu lingkungan. Pastikan objek prasyarat tersedia dan tetap gunakan schema latihan, bukan schema produksi.

Gunakan [FreeSQL dengan login](#) atau SQL Developer yang terhubung ke XE. [Baca panduan dua jalur praktikum](#). Jalankan kode sesuai urutan dan keterangan tiap langkah.

Output contoh adalah prediksi berdasarkan dataset, bukan bukti bahwa kode telah dijalankan pada akun Anda. Periksa koneksi, hak membuat objek, autocommit dan panel DBMS_OUTPUT. Slash pada baris sendiri adalah pemisah mode skrip klien tertentu, bukan statement PL/SQL.

Gunakan tabel pertemuan02 dan data pertemuan03 atau setup index sekali. Periksa keadaan data bila latihan sebelumnya pernah di-commit. Jangan menjalankan ulang CREATE TABLE/INSERT atau menghapus data secara buta.

Langkah 1 — Inventaris sebelum DDL

```
SELECT object_name, object_type, status FROM user_objects
WHERE object_name IN
('SEQ_PENJUALAN_DEMO', 'V_TOTAL_PENJUALAN_DEMO', 'AUDIT_STOK_DEMO', '
TRG_AUDIT_STOK_DEMO', 'PKG_LATIHAN_DEMO')
ORDER BY object_name, object_type;
```

Jika sequence/audit table sudah ada, inspeksi definisinya dan lewati CREATE sekali-jalan; jangan DROP atau reset sequence yang telah dipakai.

Langkah 2 — Sequence mulai 200 dan CURRVAL per sesi

```
-- Sekali saja, hanya ketika belum ada.
CREATE SEQUENCE seq_penjualan_demo
  START WITH 200 INCREMENT BY 1 MAXVALUE 99999999 NOCYCLE NOCACHE;
SELECT seq_penjualan_demo.NEXTVAL AS nomor_baru FROM dual;
SELECT seq_penjualan_demo.CURRVAL AS nomor_sesi FROM dual;
```

Jika baru dibuat, NEXTVAL pertama 200 dan CURRVAL sesi sama 200. Nomor ini telah dikonsumsi meskipun tidak dipakai INSERT. Pada sesi baru, CURRVAL sebelum NEXTVAL menghasilkan error; jangan reset sequence untuk menghapus celah.

Langkah 3 — View sederhana dengan CHECK OPTION dan view agregat read-only

```
CREATE OR REPLACE VIEW v_produk_tersedia_demo AS
  SELECT id_produk, nama_produk, harga, stok FROM produk_demo WHERE
  stok > 0
  WITH CHECK OPTION;
CREATE OR REPLACE VIEW v_total_penjualan_demo AS
  SELECT p.id_jual, p.id_pelanggan, p.tanggal, p.status,
         NVL(SUM(d.jumlah*d.harga), 0) AS total
  FROM penjualan_demo p LEFT JOIN detail_jual_demo d ON
  d.id_jual=p.id_jual
  GROUP BY p.id_jual, p.id_pelanggan, p.tanggal, p.status
  WITH READ ONLY;
SELECT * FROM v_total_penjualan_demo ORDER BY id_jual;
```

```
SELECT column_name, updatable FROM user_updatable_columns
WHERE table_name='V_TOTAL_PENJUALAN_DEMO' ORDER BY column_name;
```

Total view dihitung dari harga detail, header tanpa detail bernilai nol. UPDATE total view tidak diperbolehkan; jangan menganggap SUM bisa dipetakan ke satu baris tabel.

Langkah 4 — Tabel audit dan trigger AFTER lengkap

```
-- CREATE TABLE sekali saja ketika belum ada.
CREATE TABLE audit_stok_demo (
  id_produk NUMBER(4) NOT NULL,
  operasi VARCHAR2(6) NOT NULL CHECK (operasi IN
('INSERT', 'UPDATE', 'DELETE')),
  stok_lama NUMBER(6),
  stok_baru NUMBER(6),
  waktu TIMESTAMP DEFAULT SYSTIMESTAMP NOT NULL,
  pelaku VARCHAR2(128) NOT NULL
);
CREATE OR REPLACE TRIGGER trg_audit_stok_demo
AFTER INSERT OR UPDATE OR DELETE ON produk_demo
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO
audit_stok_demo(id_produk, operasi, stok_lama, stok_baru, pelaku)
VALUES(:NEW.id_produk, 'INSERT', NULL, :NEW.stok, USER);
  ELSIF DELETING THEN
    INSERT INTO
audit_stok_demo(id_produk, operasi, stok_lama, stok_baru, pelaku)
VALUES(:OLD.id_produk, 'DELETE', :OLD.stok, NULL, USER);
  ELSIF UPDATING THEN
    IF :OLD.stok <> :NEW.stok THEN
      INSERT INTO
audit_stok_demo(id_produk, operasi, stok_lama, stok_baru, pelaku)
VALUES(:NEW.id_produk, 'UPDATE', :OLD.stok, :NEW.stok, USER);
    END IF;
  END IF;
END;
/
```

Audit berada dalam transaksi yang sama; tidak ada COMMIT. Harga/stok asli seed tidak diubah ketika objek dibuat.

Langkah 5 — Trigger BEFORE untuk tiga jenis DML

```
CREATE OR REPLACE TRIGGER trg_validasi_produk_demo
BEFORE INSERT OR UPDATE OR DELETE ON produk_demo
FOR EACH ROW
BEGIN
  IF INSERTING OR UPDATING THEN
    IF :NEW.stok IS NULL OR :NEW.stok < 0 OR :NEW.stok <>
TRUNC(:NEW.stok) THEN
      RAISE_APPLICATION_ERROR(-20020, 'Stok harus integer
nonnegatif');
    END IF;
    IF :NEW.harga IS NULL OR :NEW.harga <= 0 THEN
      RAISE_APPLICATION_ERROR(-20021, 'Harga produk harus positif');
    END IF;
  ELSE
    DBMS_OUTPUT.PUT_LINE('DELETE produk ' || :OLD.id_produk);
  END IF;
END;
/
```

BEFORE menilai nilai baris baru untuk INSERT/UPDATE dan membaca OLD pada DELETE. Validasi harga input maksimum dua desimal tetap harus dilakukan sebelum assignment ke kolom.

Langkah 6 — Package latihan mandiri: specification, body, dan panggilan

```
CREATE OR REPLACE PACKAGE pkg_latihan_demo AS
  FUNCTION label_produk(p_nama IN VARCHAR2, p_stok IN NUMBER) RETURN
VARCHAR2;
  PROCEDURE tampil_produk;
END pkg_latihan_demo;
/

CREATE OR REPLACE PACKAGE BODY pkg_latihan_demo AS
  FUNCTION label_produk(p_nama IN VARCHAR2, p_stok IN NUMBER) RETURN
VARCHAR2 IS
```

```

BEGIN
    RETURN NVL(p_nama, 'Tanpa nama') || ' | stok=' ||
TO_CHAR(p_stok);
END label_produk;
PROCEDURE tampil_produk IS
BEGIN
    FOR r IN (SELECT nama_produk, stok FROM produk_demo ORDER BY
id_produk) LOOP
        DBMS_OUTPUT.PUT_LINE(label_produk(r.nama_produk, r.stok));
    END LOOP;
END tampil_produk;
END pkg_latihan_demo;
/
BEGIN
    pkg_latihan_demo.tampil_produk;
END;
/
SELECT pkg_latihan_demo.label_produk('Kopi', 20) AS label FROM
dual;

```

Specification hanya kontrak publik, body memuat seluruh implementasi. Nama package tidak bertabrakan dengan proyek akhir.

Langkah 7 — Uji view dan audit INSERT UPDATE DELETE dengan rollback

```

-- Transaksi mandiri: jangan ada pekerjaan penting yang tertunda.
DECLARE
    v_sebelum NUMBER;
    v_sesudah NUMBER;
    v_ada NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_sebelum FROM audit_stok_demo;
    SELECT COUNT(*) INTO v_ada FROM produk_demo WHERE id_produk=99;
    IF v_ada > 0 THEN RAISE_APPLICATION_ERROR(-20022, 'ID produk uji
99 sudah dipakai'); END IF;
    UPDATE v_produk_tersedia_demo SET stok=stok+1 WHERE id_produk=11;
    INSERT INTO produk_demo(id_produk, nama_produk, harga, stok)
VALUES(99, 'Produk uji', 1000, 2);
    UPDATE produk_demo SET stok=3 WHERE id_produk=99;

```

```

DELETE FROM produk_demo WHERE id_produk=99;
SELECT COUNT(*) INTO v_sesudah FROM audit_stok_demo;
IF v_sesudah-v_sebelum <> 4 THEN
RAISE_APPLICATION_ERROR(-20023,'Jumlah audit tidak sesuai'); END
IF;
  DBMS_OUTPUT.PUT_LINE('Audit tambahan sementara=' || (v_sesudah-
v_sebelum));
  ROLLBACK;
  SELECT COUNT(*) INTO v_sesudah FROM audit_stok_demo;
  IF v_sesudah <> v_sebelum THEN
RAISE_APPLICATION_ERROR(-20024,'Audit tidak kembali sesudah
rollback'); END IF;
  DBMS_OUTPUT.PUT_LINE('Rollback data dan audit dibuktikan');
EXCEPTION WHEN OTHERS THEN ROLLBACK; RAISE;
END;
/
SELECT id_produk,stok FROM produk_demo WHERE id_produk IN (11,99)
ORDER BY id_produk;

```

Prediksi empat audit tambahan sebelum rollback, jumlah audit awal dipulihkan sesudahnya. Stok Kopi kembali nilai awal dan produk 99 tidak tersisa. Hindari sesi lain menulis selama perbandingan count.

Langkah 8 — Buktikan pembatasan update view tanpa merusak seed

```

DECLARE
  v_kode NUMBER;
BEGIN
  SAVEPOINT batas_view_13;
  BEGIN
    UPDATE v_produk_tersedia_demo SET stok=0 WHERE id_produk=11;
    RAISE_APPLICATION_ERROR(-20996,'CHECK OPTION tidak menolak
perubahan');
  EXCEPTION WHEN OTHERS THEN
    v_kode := SQLCODE;
    ROLLBACK TO batas_view_13;
    IF v_kode <> -1402 THEN RAISE; END IF;
    DBMS_OUTPUT.PUT_LINE('CHECK OPTION menolak stok nol');
  END;

```

```

BEGIN
  EXECUTE IMMEDIATE 'UPDATE v_total_penjualan_demo SET total=1
WHERE id_jual=101';
  RAISE_APPLICATION_ERROR(-20997, 'View read-only justru dapat
diubah');
  EXCEPTION WHEN OTHERS THEN
    v_kode := SQLCODE;
    ROLLBACK TO batas_view_13;
    IF v_kode NOT IN (-42399, -1732) THEN RAISE; END IF;
    DBMS_OUTPUT.PUT_LINE('View agregat/read-only menolak UPDATE');
  END;
  ROLLBACK;
  EXCEPTION WHEN OTHERS THEN ROLLBACK; RAISE;
  END;
  /

```

Gunakan transaksi mandiri. CHECK OPTION biasanya ORA-01402; larangan read-only/operasi view dapat ORA-42399 atau ORA-01732. Error lain tidak ditelan. Nilai seed tetap utuh.

Langkah 9 — Inspeksi USER_ERRORS dan status

```

SELECT name, type, line, position, text
FROM user_errors
WHERE name IN
('TRG_AUDIT_STOK_DEMO', 'TRG_VALIDASI_PRODUK_DEMO', 'PKG_LATIHAN_DEM
O')
ORDER BY name, type, sequence;
SELECT object_name, object_type, status FROM user_objects WHERE
object_name IN
('TRG_AUDIT_STOK_DEMO', 'TRG_VALIDASI_PRODUK_DEMO', 'PKG_LATIHAN_DEM
O', 'V_TOTAL_PENJUALAN_DEMO') ORDER BY object_name, object_type;

```

Kumpulkan hasil aktual. USER_ERRORS kosong saja belum membuktikan objek ada, sehingga cek USER_OBJECTS juga.

Tugas / latihan

1. Pada XE lokal ATAU FreeSQL schema sendiri, buat objek baru sekali dan lampirkan daftar objek/status serta USER_ERRORS aktual.
2. Bandingkan NEXTVAL/CURRVAL dalam sesi yang sama dan jelaskan mengapa rollback tidak memulihkan nomor. Tidak wajib membuka dua sesi bila editor tidak mendukung.
3. Uji view sederhana dan dokumentasikan penolakan UPDATE pada view agregat dalam transaksi tersendiri; rollback perubahan.
4. Jalankan audit INSERT/UPDATE/DELETE menggunakan ID produk kosong; buktikan audit ikut rollback tanpa menghapus seed.
5. Presentasikan pemisahan specification/body pkg_latihan_demo dan tandai empat objek yang digunakan kembali pada modul 14.

Pertanyaan analisis

1. Mengapa audit DELETE tidak diberi foreign key ke produk, dan apa batas integritas pilihan ini?
2. Mengapa CURRENT session CURRVAL dan celah nomor tidak boleh dianggap bukti transaksi berhasil atau gagal?

Pengumpulan

Tugas boleh dikerjakan di **Oracle XE lokal atau FreeSQL dengan login**; cukup pilih salah satu. Kumpulkan satu laporan melalui LMS/kanal yang ditetapkan dosen. Cantumkan Nama, NPM, pertemuan dan lingkungan yang dipakai. Sertakan kode SQL/PLSQL sebagai teks, output aktual atau pesan error, serta jawaban analisis.

Jika FreeSQL menyediakan tautan berbagi yang dapat diakses dosen, tautan boleh ditambahkan sebagai bukti pendamping, bukan pengganti hasil eksekusi yang terdokumentasi. Jangan membagikan password, token atau koneksi pribadi.

Hasil yang belum dieksekusi diberi label “prediksi”. Screenshot ilustrasi browser bukan bukti praktikum Oracle.

Rubrik penilaian

- Ketepatan konsep dan rancangan: **20%**
- Kebenaran query/program: **30%**
- Pengujian normal, batas dan gagal: **20%**

- Analisis serta penjelasan hasil: **20%**
 - Identitas dan kerapian laporan: **10%**
-

Referensi resmi

- [Oracle 19c — CREATE SEQUENCE](#)
- [Oracle 19c — CREATE VIEW](#)
- [Oracle 19c — PL/SQL Triggers](#)
- [Oracle 19c — PL/SQL Packages](#)
- [Oracle 19c — USER_ERRORS](#)

Panduan XE21c digunakan untuk konteks instalasi lokal; dokumentasi SQL/PLSQL untuk konsep bahasa. Versi FreeSQL/editor dapat berbeda, sehingga periksa fitur dan pesan aktual.
