

ORACLE PL/SQL · PERTEMUAN 08 / 14

Mini Project SQL Oracle

Merancang schema penjualan sederhana, menguji transaksi, dan mempertanggungjawabkan laporan.

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

Versi PDF materi pertemuan 08. Contoh dan tabel dapat dibaca di PDF. Kuis dapat dijalankan di halaman web: [Buka materi daring](#).

Tujuan pembelajaran

1. Merancang schema logis penjualan dengan relasi dan constraint yang tepat.
2. Memisahkan tahap DDL, data awal dan transaksi uji DML.
3. Menghasilkan laporan JOIN, aggregate dan subquery yang sesuai kebutuhan.
4. Menguji kasus positif/negatif dan mempresentasikan hasil dengan rubric terukur.

Pokok bahasan / praktikum

- Schema logis sederhana
- Tabel dan constraints
- DDL
- DML
- Data transaksional
- JOIN
- Aggregate GROUP BY HAVING
- Subquery
- SAVEPOINT dan ROLLBACK
- Pengujian positif dan negatif
- Presentasi
- Rubrik penilaian

Prasyarat: Selesaikan modul 02/03 atau jalankan setup bersama sekali pada schema latihan sendiri. Gunakan Oracle XE lokal melalui SQL Developer ATAU Oracle FreeSQL setelah login pada schema pribadi yang diizinkan; tidak harus memakai keduanya. Jangan menjalankan ulang CREATE TABLE/seed pada objek yang sudah ada. Selesaikan modul 04–07.

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. Aturan bisnis mana yang harus dijaga constraint, bukan hanya dokumentasi?
2. Bagaimana membuktikan transaksi uji tidak mengotori seed?
3. Apa bukti bahwa laporan benar untuk pelanggan tanpa pembelian?

Uji pemahaman awal

1. Apa arti membuat database pada mini project FreeSQL ini?

- Membuat instance baru memakai CREATE DATABASE
- Mendesain tabel dan relasi dalam schema pribadi yang diizinkan
- Menggunakan akun DBA bersama
- Memasang XE walaupun FreeSQL sudah tersedia

2. Mengapa DDL harus dipisah dari percobaan rollback DML?

- DDL selalu dapat rollback
- DDL menghapus semua foreign key
- DDL Oracle dapat implicit commit dan mengganggu savepoint
- SAVEPOINT hanya berlaku untuk SELECT

3. Dalam outer join pelanggan, hitungan transaksi yang benar memakai ...

- COUNT(*) tanpa memperhatikan pasangan
- COUNT(j.id_jual) dengan agregasi pada granularitas transaksi
- SUM(p.id_pelanggan)
- Jumlah kolom SELECT

PAHAMI KONSEP · BACA TERSTRUKTUR · TERAPKAN

Teori dan pembahasan

Mini project menyatukan rancangan schema, perubahan data dan laporan. Selesaikan tahapannya berurutan agar hasil akhir dapat dijelaskan dan diuji.

Konsep inti bab ini

- Database sederhana di sini berarti schema logis, bukan CREATE DATABASE di FreeSQL.
- Selesaikan DDL sebelum menguji transaksi rollback.
- Pisahkan prediksi hasil, hasil aktual dan kesimpulan pengujian.

Urutan pembahasan

- 01 Cakupan proyek dan dua jalur lingkungan
- 02 Rancangan tabel dan integritas
- 03 Tahap DDL dan data awal
- 04 Tahap transaksi DML yang dapat dipulihkan
- 05 Tahap laporan JOIN, aggregate dan subquery
- 06 Tahap pengujian, presentasi dan rubrik

Gambaran konsep dalam satu tabel

Artefak	Isi utama	Bukti yang ditampilkan
ERD / kamus data	Entitas, relasi dan tipe data	Alasan rancangan
DDL / seed	Tabel, constraints dan data awal	Metadata dan pemeriksaan data
DML / transaksi	Perubahan terkontrol	Keadaan sebelum/sesudah rollback
Query laporan	JOIN, aggregate, subquery	Definisi hasil dan data acuan
Matriks uji	Kasus sukses dan gagal	SQL, output aktual, kesimpulan

01 Cakupan proyek dan dua jalur lingkungan

Mini project membuat rancangan database sederhana dalam arti schema logis: sekumpulan tabel, relasi, aturan, data dan laporan penjualan.

Ini bukan instruksi CREATE DATABASE. Pada FreeSQL, login dan gunakan schema pribadi yang diizinkan; mahasiswa tidak membuat instance, pengguna administratif atau tablespace.

POIN PEMBAHASAN

- Alternatifnya adalah Oracle XE lokal melalui SQL Developer pada pengguna latihan di XEPDB1 .

- Kedua jalur cukup untuk tugas SQL ini dan tidak harus dilakukan bersamaan.
- Pastikan sesi tidak menyimpan pekerjaan lain sebelum latihan transaksi.
- Proyek ini merupakan latihan pembelajaran, bukan sistem pembayaran produksi; konkurensi, keamanan aplikasi, pencadangan dan audit lengkap berada di luar cakupannya.

02 Rancangan tabel dan integritas

Gunakan `pelanggan_demo`, `produk_demo`, `penjualan_demo` dan `detail_jual_demo` dari setup bersama; `pegawai_demo` tetap tersedia tetapi tidak wajib menjadi objek proyek penjualan.

Gambarkan hubungan satu pelanggan ke banyak penjualan dan satu penjualan ke banyak detail. Primary key detail merupakan pasangan transaksi dan produk.

POIN PEMBAHASAN

- Jelaskan alasan pelanggan transaksi nullable untuk pembelian anonim, sementara harga dan jumlah detail tidak nullable.
- Constraint `CHECK` menjaga harga positif, stok tidak negatif, dan status dalam daftar yang diperbolehkan.
- `NUMBER` dengan skala uang membatasi penyimpanan desimal, tetapi dapat membulatkan input; aturan bahwa masukan harus maksimal dua desimal perlu divalidasi sebelum penyimpanan jika dibutuhkan aplikasi.
- Jangan mengklaim skala kolom selalu menolak masukan berlebih.

03 Tahap DDL dan data awal

Tahap pertama menginventarisasi objek dan constraint.

Jika schema baru benar-benar belum memiliki tabel, jalankan DDL dan seed dari setup bersama atau modul 02/03 sekali, berurutan dari tabel induk ke anak. Jika objek sudah ada, jangan menjalankan `CREATE TABLE` ulang, jangan `DROP`, dan jangan memasukkan seed duplikat.

POIN PEMBAHASAN

- DDL Oracle dapat melakukan implicit commit, sehingga harus selesai sebelum demonstrasi rollback.
- Tahap berikutnya membaca data awal dan mencatat jumlah baris sebagai pembanding; query ini tidak mengubah data.
- Gunakan literal DATE untuk tanggal agar tidak bergantung pada format sesi.
- Simpan rancangan relasi dan kamus data bersama skrip, bukan hanya tangkapan layar objek.

04 Tahap transaksi DML yang dapat dipulihkan

Contoh bertahap memakai transaksi seed 102 milik Bima dan produk 11 yang belum menjadi detail pada transaksi itu.

Sebelum perubahan, query pemeriksaan harus menunjukkan pasangan 102/11 belum ada. Jika latihan sebelumnya telah menambahkannya, hentikan tahap tersebut dan pilih pasangan valid yang kosong secara eksplisit.

POIN PEMBAHASAN

- `SAVEPOINT` membatasi pekerjaan uji.
- Tambahkan detail, kurangi stok dengan kondisi stok mencukupi, lalu ubah status kepala transaksi.
- Periksa perubahan dan SQL laporan sebelum `ROLLBACK TO savepoint`.
- Ini bukan otomatisasi penjualan yang aman terhadap semua kegagalan: setiap hasil DML harus diperiksa dan mahasiswa wajib rollback bila prasyarat gagal.
- Hindari `COMMIT` tanpa keputusan jelas; pemilik transaksi adalah mahasiswa pada sesi latihan.

05 Tahap laporan JOIN, aggregate dan subquery

Laporan detail menampilkan pelanggan, produk, jumlah, harga historis dan nilai baris.

LEFT JOIN menjaga transaksi anonim, sementara aggregate menghitung total setiap transaksi. Laporan pelanggan memakai `COUNT` atas ID penjualan, bukan `COUNT (*)` setelah outer join, agar pelanggan tanpa pembelian tidak tampak mempunyai transaksi.

POIN PEMBAHASAN

- Gunakan `HAVING` untuk memilih kelompok yang memenuhi ambang total; syarat baris biasa ditempatkan di `WHERE`.
- Subquery `NOT EXISTS` menemukan produk yang belum terjual tanpa masalah `NULL` pada kumpulan ID pelanggan.
- Bandingkan laporan saat detail uji masih ada dan sesudah rollback.
- Tulis definisi omzet: contoh menghitung nilai detail semua status, bukan otomatis hanya penjualan lunas.
- Jika kebutuhan berbeda, nyatakan filter status yang digunakan.

06 Tahap pengujian, presentasi dan rubrik

Siapkan matriks uji berisi tujuan, SQL, hasil yang diharapkan, hasil aktual dan kesimpulan.

Kasus positif meliputi relasi valid, transaksi anonim dan produk tanpa detail; kasus negatif meliputi jumlah nol, foreign key tidak ditemukan dan duplicate primary key. Contoh negatif harus dijalankan terpisah karena runner dapat berhenti saat error.

POIN PEMBAHASAN

- Setelahnya rollback ke savepoint dan periksa data awal tetap utuh.
- Presentasi menunjukkan diagram, aturan, skrip berurutan dan satu perbedaan laporan sebelum/sesudah rollback.
- Rubrik berjumlah 100 poin: rancangan dan constraints 25; DDL/DML berurutan serta transaksi 25; JOIN, aggregate dan subquery 25; bukti pengujian 15; presentasi dan dokumentasi 10.
- Bukti yang belum dijalankan ditandai belum diuji, tidak diganti keluaran buatan.

Tahapan mini project

- 1 Rancang schema
- 2 Buat DDL dan seed sekali
- 3 Jalankan DML terkontrol
- 4 Susun query laporan
- 5 Uji dan presentasikan

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

AMATI DATA · IKUTI LANGKAH · PAHAMI 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.

Laporan mini project yang dapat diperiksa

Gunakan laporan pelanggan sebagai salah satu keluaran project, lalu cocokkan dengan data acuan.

Pelanggan	Jumlah penjualan	Total penjualan
Ayu	1	300000
Bima	1	150000
Citra	0	0

Kasus uji	Yang diperiksa	Hasil yang diprediksi
Kunci produk duplikat	PRIMARY KEY	INSERT ditolak

Kasus uji	Yang diperiksa	Hasil yang diprediksi
Jumlah detail = 0	CHECK	INSERT ditolak
Rollback DML	Stok dan transaksi	Kembali ke kondisi sebelum perubahan

Cara membaca contoh

1. Buat schema dan isi seed satu kali, bukan berulang tanpa pemeriksaan.
2. Jelaskan mengapa Citra tetap tampil dan transaksi anonim tidak masuk laporan per pelanggan.
3. Lakukan uji perubahan DML dengan savepoint; setelah rollback, verifikasi lagi total dan stok.

Yang perlu dipahami: Matriks uji mencatat pertanyaan, hasil yang diprediksi dan hasil aktual. Tabel di sini adalah acuan pembahasan, bukan bukti project Anda.

Video pembelajaran

Video utama — fokus bab ini

Referensi rancangan database penjualan menggunakan Oracle SQL Developer/Data Modeler, selaras dengan mini project pelanggan–produk–penjualan.

Oracle SQL Developer "Tata cara pembuatan database penjualan suatu barang"

Pickollo · [Buka video utama di YouTube](#)

Batas cakupan dan panduan menonton: Video ini fokus rancangan, bukan walkthrough lengkap seluruh query project. DDL/DML, JOIN, aggregate, subquery, query testing dan presentasi dikerjakan melalui tahapan project pada halaman. Tidak perlu menginstal aplikasi UI tambahan untuk tugas SQL.

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 — Tahap 1 — inventarisasi sebelum DDL

```
SELECT table_name FROM user_tables
WHERE table_name IN
('PELANGGAN_DEMO', 'PRODUK_DEMO', 'PENJUALAN_DEMO', 'DETAIL_JUAL_DEMO',
'PEGAWAI_DEMO')
ORDER BY table_name;
```

```

SELECT table_name, constraint_name, constraint_type, status
FROM user_constraints
WHERE table_name IN
('PELANGGAN_DEMO', 'PRODUK_DEMO', 'PENJUALAN_DEMO', 'DETAIL_JUAL_DEMO')
ORDER BY table_name, constraint_name;

```

Jika tabel lengkap sudah ada, lewati tahap CREATE/seed berikutnya. Objek yang hanya sebagian ada harus ditinjau, bukan langsung dihapus.

Langkah 2 — Tahap 2 — DDL hanya pada schema kosong

```

CREATE TABLE pelanggan_demo (
  id_pelanggan NUMBER(4) PRIMARY KEY,
  nama VARCHAR2(60) NOT NULL,
  kota VARCHAR2(30),
  email VARCHAR2(80) UNIQUE
);
CREATE TABLE produk_demo (
  id_produk NUMBER(4) PRIMARY KEY,
  nama_produk VARCHAR2(60) NOT NULL,
  harga NUMBER(12,2) NOT NULL CHECK (harga > 0),
  stok NUMBER(6) DEFAULT 0 NOT NULL CHECK (stok >= 0)
);
CREATE TABLE penjualan_demo (
  id_jual NUMBER(8) PRIMARY KEY,
  id_pelanggan NUMBER(4) REFERENCES pelanggan_demo(id_pelanggan),
  tanggal DATE DEFAULT SYSDATE NOT NULL,
  status VARCHAR2(12) DEFAULT 'BARU' NOT NULL,
  CONSTRAINT ck_jual_status CHECK (status IN
('BARU', 'LUNAS', 'BATAL'))
);
CREATE TABLE detail_jual_demo (
  id_jual NUMBER(8) REFERENCES penjualan_demo(id_jual),
  id_produk NUMBER(4) REFERENCES produk_demo(id_produk),
  jumlah NUMBER(6) NOT NULL CHECK (jumlah > 0),
  harga NUMBER(12,2) NOT NULL CHECK (harga > 0),
  CONSTRAINT pk_detail_demo PRIMARY KEY (id_jual, id_produk)
);

```

```
CREATE TABLE pegawai_demo (
  id_pegawai NUMBER(4) PRIMARY KEY,
  nama VARCHAR2(60) NOT NULL,
  id_manajer NUMBER(4) REFERENCES pegawai_demo(id_pegawai)
);
```

Salinan DDL bersama. Jalankan sekali hanya ketika kelima tabel belum ada; schema existing menggunakan objek modul 02. DDL dilakukan sebelum savepoint dan dapat commit implisit.

Langkah 3 — Tahap 3 — seed hanya jika belum dimuat

```
INSERT ALL
  INTO pelanggan_demo VALUES (1, 'Ayu', 'Subang', 'ayu@example.test')
  INTO pelanggan_demo VALUES
(2, 'Bima', 'Bandung', 'bima@example.test')
  INTO pelanggan_demo VALUES (3, 'Citra', NULL, NULL)
SELECT 1 FROM dual;
INSERT ALL
  INTO produk_demo VALUES (11, 'Kopi', 30000, 20)
  INTO produk_demo VALUES (12, 'Teh', 15000, 30)
  INTO produk_demo VALUES (13, 'Gula', 12000, 40)
  INTO produk_demo VALUES (14, 'Sabun', 10000, 0)
SELECT 1 FROM dual;
INSERT ALL
  INTO penjualan_demo VALUES (101, 1, DATE '2026-10-01', 'LUNAS')
  INTO penjualan_demo VALUES (102, 2, DATE '2026-10-02', 'BARU')
  INTO penjualan_demo VALUES (103, NULL, DATE '2026-10-03', 'BARU')
SELECT 1 FROM dual;
INSERT ALL
  INTO detail_jual_demo VALUES (101, 11, 10, 30000)
  INTO detail_jual_demo VALUES (102, 12, 10, 15000)
  INTO detail_jual_demo VALUES (103, 13, 5, 12000)
SELECT 1 FROM dual;
INSERT INTO pegawai_demo VALUES (1, 'Sari', NULL);
INSERT INTO pegawai_demo VALUES (2, 'Rudi', 1);
INSERT INTO pegawai_demo VALUES (3, 'Tono', 1);
COMMIT;
```

Data persis setup bersama; jalankan sekali setelah DDL pada schema kosong. Jangan menduplikasi seed modul 03. COMMIT di sini menutup tahap data awal, bukan transaksi uji.

Langkah 4 — Tahap 4 — periksa prasyarat transaksi uji

```
SELECT id_jual, id_pelanggan, status FROM penjualan_demo WHERE
id_jual = 102;
SELECT id_produk, harga, stok FROM produk_demo WHERE id_produk =
11;
SELECT COUNT(*) AS pasangan_sudah_ada FROM detail_jual_demo
WHERE id_jual = 102 AND id_produk = 11;
```

Lanjutkan hanya jika kepala 102 dan produk 11 ada, stok mencukupi, dan pasangan_sudah_ada nol. Pada seed asli syarat tersebut terpenuhi; tetap verifikasi pada sesi sendiri.

Langkah 5 — Tahap 5 — DML dan laporan sebelum pemulihan

```
SAVEPOINT sebelum_mini;
INSERT INTO detail_jual_demo (id_jual, id_produk, jumlah, harga)
SELECT 102, id_produk, 2, harga FROM produk_demo
WHERE id_produk = 11 AND stok >= 2;
UPDATE produk_demo SET stok = stok - 2 WHERE id_produk = 11 AND
stok >= 2;
UPDATE penjualan_demo SET status = 'LUNAS' WHERE id_jual = 102;

SELECT j.id_jual, NVL(p.nama, 'Anonim') AS pelanggan,
       r.nama_produk, d.jumlah, d.harga, d.jumlah*d.harga AS nilai
FROM penjualan_demo j
LEFT JOIN pelanggan_demo p ON p.id_pelanggan = j.id_pelanggan
JOIN detail_jual_demo d ON d.id_jual = j.id_jual
JOIN produk_demo r ON r.id_produk = d.id_produk
WHERE j.id_jual = 102
ORDER BY d.id_produk;
```

Latihan manual, bukan prosedur atomik otomatis: pastikan INSERT dan kedua UPDATE masing-masing memengaruhi satu baris. Jika tidak, ROLLBACK TO sebelum_mini lalu hentikan. Tidak menjalankan DDL selama tahap ini.

Langkah 6 — Tahap 6 — aggregate, HAVING dan subquery

```

WITH total_jual AS (
  SELECT id_jual, SUM(jumlah*harga) AS total
  FROM detail_jual_demo GROUP BY id_jual
)
SELECT p.id_pelanggan, p.nama, COUNT(j.id_jual) AS
banyak_transaksi,
      NVL(SUM(t.total),0) AS nilai_semua_status
FROM pelanggan_demo p
LEFT JOIN penjualan_demo j ON j.id_pelanggan = p.id_pelanggan
LEFT JOIN total_jual t ON t.id_jual = j.id_jual
GROUP BY p.id_pelanggan, p.nama
ORDER BY p.id_pelanggan;

SELECT id_jual, SUM(jumlah*harga) AS total
FROM detail_jual_demo GROUP BY id_jual
HAVING SUM(jumlah*harga) >= 100000
ORDER BY id_jual;

SELECT r.id_produk, r.nama_produk FROM produk_demo r
WHERE NOT EXISTS (SELECT 1 FROM detail_jual_demo d WHERE
d.id_produk = r.id_produk)
ORDER BY r.id_produk;

```

CTE mencegah jumlah transaksi mengganda per detail. Definisi nilai mencakup semua status; mahasiswa dapat menambahkan filter LUNAS sesuai kebutuhan yang didokumentasikan.

Langkah 7 — Tahap 7 — rollback perubahan uji dan verifikasi

```

ROLLBACK TO sebelum_mini;
SELECT id_jual, id_pelanggan, status FROM penjualan_demo WHERE
id_jual = 102;
SELECT id_produk, harga, stok FROM produk_demo WHERE id_produk =
11;
SELECT COUNT(*) AS pasangan_setelah_rollback FROM detail_jual_demo
WHERE id_jual = 102 AND id_produk = 11;
COMMIT;

```

Bandingkan status, stok dan jumlah pasangan dengan baseline tahap 4. COMMIT hanya menutup sesi transaksi latihan yang sejak awal dipastikan tidak memuat pekerjaan lain.

Langkah 8 — Tahap 8a — uji negatif CHECK terpisah

```
SAVEPOINT uji_check_mini;
INSERT INTO detail_jual_demo (id_jual,id_produk,jumlah,harga)
VALUES (102,11,0,30000);
-- Jalankan manual setelah error yang diharapkan:
ROLLBACK TO uji_check_mini;
```

Dijalankan setelah rollback tahap 7, dengan pasangan 102/11 belum ada. Jumlah nol diharapkan ditolak CHECK. Jika runner berhenti saat error, jalankan rollback terpisah; jangan mengklaim error sudah diuji.

Langkah 9 — Tahap 8b — uji negatif FK dan primary key terpisah

```
-- Jalankan setiap percobaan secara terpisah dan catat error
aktual.
SAVEPOINT uji_fk_mini;
INSERT INTO penjualan_demo (id_jual,id_pelanggan,tanggal,status)
SELECT 200, 9999, DATE '2026-10-04', 'BARU' FROM dual
WHERE NOT EXISTS (SELECT 1 FROM penjualan_demo WHERE id_jual =
200)
AND NOT EXISTS (SELECT 1 FROM pelanggan_demo WHERE id_pelanggan
= 9999);
-- Jika INSERT nol baris, prasyarat tidak terpenuhi: jangan anggap
FK telah diuji.
ROLLBACK TO uji_fk_mini;

SAVEPOINT uji_pk_mini;
INSERT INTO pelanggan_demo (id_pelanggan,nama) VALUES
(1,'Duplikat');
-- Jalankan manual setelah error yang diharapkan:
ROLLBACK TO uji_pk_mini;
```

Pada seed, 200 dan 9999 belum ada sehingga FK ditolak. ID 200 hanya percobaan, bukan data tetap/sequence. Jika sudah dipakai, pilih ID uji tidak terpakai dan catat. PK mengasumsikan pelanggan 1 ada. Lanjutkan rollback manual jika runner berhenti.

Tugas / latihan

1. Kerjakan melalui XE lokal ATAU FreeSQL schema pribadi: serahkan diagram relasi, kamus data dan skrip DDL/seed yang aman untuk schema baru. Jelaskan bahwa schema existing melewati DDL/seed.
2. Jalankan tahapan DML, laporan JOIN/agregasi/subquery, rollback dan verifikasi baseline; lampirkan bukti aktual perubahan baris.
3. Buat matriks uji positif/negatif dan presentasi singkat. Rubrik: rancangan/constraints 25, DDL/DML/transaksi 25, laporan 25, pengujian 15, dokumentasi/presentasi 10; total 100.

Pertanyaan analisis

1. Apa konsekuensi menghitung harga dari produk saat laporan historis dibuat, dibandingkan harga pada detail?
2. Apa kelemahan transaksi manual contoh jika salah satu UPDATE memengaruhi nol baris, dan bagaimana rollback serta pemeriksaan hasil membantu membatasi dampaknya?

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.

Presentasi project: jelaskan skema, jalankan query/proses, tampilkan kasus sukses dan gagal, lalu terangkan integritas serta transaksi.

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 TABLE](#)
- [Oracle 19c — COMMIT](#)
- [Oracle 19c — Joins](#)
- [Oracle 19c — SELECT](#)

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