

ORACLE PL/SQL · PERTEMUAN 02 / 14

DDL – Data Definition Language

Membangun struktur bersama dan mencoba perubahan hanya pada tabel latihan terpisah.

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

Versi PDF materi pertemuan 02. Contoh dan tabel dapat dibaca di PDF. Kuis dapat dijalankan di halaman web: [Buka materi daring](#).

Tujuan pembelajaran

1. Membuat lima tabel bersama dengan DDL persis sesuai dataset.
2. Memilih tipe Oracle serta menjelaskan semua constraint dasar.
3. Memakai ALTER, RENAME, TRUNCATE, dan DROP pada objek demonstrasi sendiri.
4. Memetakan relasi pelanggan, penjualan, produk, detail, dan pegawai.

Pokok bahasan / praktikum

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- TRUNCATE TABLE
- RENAME
- NUMBER VARCHAR2 CHAR DATE TIMESTAMP CLOB
- PRIMARY KEY FOREIGN KEY UNIQUE NOT NULL CHECK DEFAULT
- Relasi tabel dan self-reference
- Implicit commit DDL

Prasyarat: Modul01 dan schema pribadi. Jalankan di XE lokal ATAU FreeSQL login yang mengizinkan CREATE TABLE; tidak perlu administrasi 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.

AMATI → PREDIKSI → JELASKAN

Mulai dari pertanyaan

1. Mengapa dua baris tidak boleh memakai identitas produk sama?
2. Apa bedanya menghapus isi tabel dengan menghapus tabel?
3. Bolehkah penjualan terjadi tanpa pelanggan yang teridentifikasi?

Uji pemahaman awal

1. Apa arti primary key detail_jual_demo?

- Hanya id_jual unik
- Pasangan id_jual dan id_produk unik dan wajib terisi
- Harga tidak dapat berubah
- Setiap produk harus pernah terjual

2. Apa akibat TRUNCATE yang benar?

- Hanya satu baris dihapus
- Definisi kolom dihapus
- Seluruh isi hilang dan tidak dipulihkan dengan rollback biasa
- Selalu menghapus tabel parent

3. Jika stok diisi NULL eksplisit, apa peran DEFAULT 0 biasa?

- Selalu mengganti NULL
- Tidak mengganti NULL; NOT NULL menolak
- Mengubah NULL menjadi string kosong
- Menghapus baris otomatis

PAHAMI KONSEP · BACA TERSTRUKTUR · TERAPKAN

Teori dan pembahasan

DDL membentuk struktur yang menjadi fondasi seluruh latihan berikutnya. Pelajari tipe data dan aturan integritas sebelum mencoba perubahan atau penghapusan objek.

Konsep inti bab ini

- Primary key memberi identitas; foreign key menjaga hubungan antartabel.
- DEFAULT tidak menggantikan NULL eksplisit pada default biasa.
- Pisahkan DDL dari eksperimen rollback karena DDL dapat melakukan implicit commit.

Urutan pembahasan

01 DDL menentukan bentuk data

02 Tipe data Oracle yang sesuai

03 Kunci dan keunikan

04 Aturan nilai dan default

05 Relasi dan urutan pembuatan

06 Perubahan aman dan implicit commit

Gambaran konsep dalam satu tabel

Operasi	Perubahan utama	Perhatian
CREATE TABLE	Membuat struktur baru	Urutkan tabel induk sebelum anak
ALTER TABLE	Mengubah kolom/constraint	Periksa data lama dan dependensi
RENAME	Mengganti nama objek	Perbarui skrip yang memakai nama lama
TRUNCATE	Mengosongkan seluruh baris	Tidak memiliki sifat rollback DELETE
DROP	Menghapus objek tabel	Jangan hapus tabel bersama sembarangan

01 DDL menentukan bentuk data

DDL mendefinisikan objek: **CREATE** membangun, **ALTER** mengubah struktur, **RENAME** mengganti nama, **TRUNCATE** mengosongkan tabel, dan **DROP** menghapus tabel sebagai objek.

CREATE TABLE biasa menghasilkan tabel kosong, bukan otomatis baris contoh. Karena itu modul02 menyediakan struktur persis yang akan diisi modul03. Pahami definisi setiap kolom sebelum menjalankan skrip, bukan sekadar mengejar pesan berhasil.

POIN PEMBAHASAN

- Gunakan schema pribadi pada salah satu jalur.
- Jalankan setup sekali; jika nama tabel sudah ada, periksa `USER_TABLES` dan struktur sebelumnya.
- `ORA-00955` dapat menandakan objek bernama sama.
- Jangan memperbaikinya dengan menghapus semua objek.

- Simpan tabel bersama karena modul berikutnya bergantung padanya; operasi destruktif dibatasi pada `ddl_latihan_demo` yang dibuat khusus untuk percobaan.

02 Tipe data Oracle yang sesuai

NUMBER(p,s) menyimpan angka dengan precision dan scale; harga **NUMBER(12,2)** menyediakan dua digit pecahan.

NUMBER(4) dipakai identitas kecil dalam dataset, bukan pernyataan bahwa semua identitas dunia nyata harus angka. **VARCHAR2** menyimpan teks panjang variabel; **CHAR** panjang tetap cocok untuk kode yang memang tetap, namun padding dapat memengaruhi perbandingan. Ukuran karakter perlu diperhatikan jika sistem menggunakan teks multibita.

POIN PEMBAHASAN

- **DATE** Oracle menyimpan tanggal dan waktu sampai detik, bukan hanya hari kalender.
- **TIMESTAMP** memberi pecahan detik, sementara **CLOB** untuk teks besar yang tidak cocok dipaksakan ke kolom pendek.
- Contoh tabel terpisah memperkenalkan tipe tersebut tanpa mengubah skema bersama.
- Memasukkan teks bukan angka ke kolom **NUMBER** atau teks terlalu panjang dapat gagal; jangan bergantung konversi implisit sebagai strategi desain.

03 Kunci dan keunikan

PRIMARY KEY menjamin identitas unik dan tidak **NULL**; `pelanggan_demo` memakai `id_pelanggan`.

FOREIGN KEY memastikan nilai rujukan non-**NULL** ditemukan pada kunci induk. Pada DDL, kata **REFERENCES** membentuk foreign key meskipun kata **FOREIGN KEY** tidak ditulis eksplisit. Parent pelanggan dan produk dibuat sebelum child penjualan dan detail agar target rujukan sudah tersedia.

POIN PEMBAHASAN

- **UNIQUE** pada email menolak email non-**NULL** yang sama, tetapi kolom itu tetap opsional dan dapat berisi beberapa **NULL**.
- Foreign key `id_pelanggan` pada penjualan juga opsional: penjualan anonim 103 sah karena **NULL** diizinkan.
- Ketidadaan pelanggan pada sebuah penjualan berbeda dari menulis ID pelanggan yang tidak ada.
- Dua aturan ini penting ketika nanti membaca hasil penggabungan tabel.

04 Aturan nilai dan default

NOT NULL mewajibkan informasi, sehingga nama dan harga produk tidak boleh hilang.

CHECK `harga > 0` melarang harga nol atau negatif, dan **CHECK** `stok >= 0` melarang stok negatif. **CHECK** saja tidak menjamin nilai tersedia karena kondisi tak diketahui akibat **NULL** bukan **FALSE**; sebab itu kolom penting menggabungkannya dengan **NOT NULL**. Oracle memperlakukan string kosong sebagai **NULL** pada tipe karakter biasa.

POIN PEMBAHASAN

- **DEFAULT** bukan constraint penolakan, melainkan nilai pengganti saat kolom tidak diberikan atau kata **DEFAULT** digunakan.
- Stok default nol, tanggal default `SYSDATE`, dan status default **BARU**.
- **DEFAULT** biasa tidak mengganti **NULL** eksplisit, jadi **NOT NULL** masih diperlukan.
- Status dibatasi **CHECK** menjadi **BARU**, **LUNAS**, atau **BATAL**; salah ejaan ditolak daripada disimpan lalu membingungkan laporan.

05 Relasi dan urutan pembuatan

Satu pelanggan dapat mempunyai banyak penjualan; pelanggan tanpa penjualan tetap boleh disimpan.

Satu penjualan mempunyai banyak detail, sedangkan satu produk dapat muncul pada banyak detail. Detail menjadi penghubung penjualan dengan produk, dan primary key gabungan (id_jual,id_produk) mencegah produk yang sama diulang sebagai dua detail terpisah pada satu penjualan. Harga detail merekam harga transaksi, bukan selalu harga katalog terkini.

POIN PEMBAHASAN

- `Pegawai_demo` memiliki self-reference: `id_manajer` merujuk `id_pegawai` pada tabel yang sama.
- Manajer puncak dapat memakai `NULL`.
- Constraint ini menjamin manajer yang dirujuk ada, tetapi tidak otomatis mencegah seluruh kemungkinan siklus organisasi.
- Jangan menganggap satu foreign key menyelesaikan semua aturan bisnis.
- Masukkan manajer sebelum bawahannya ketika memakai constraint langsung yang tidak ditunda.

06 Perubahan aman dan implicit commit

`ALTER TABLE` dapat `ADD` kolom, `MODIFY` ukuran, `RENAME COLUMN`, atau `DROP COLUMN`.

Mengurangi panjang kolom berisi data bisa gagal; menambah `NOT NULL` ke tabel terisi memerlukan penanganan data lama. `RENAME` tabel juga memengaruhi skrip yang menggunakan nama sebelumnya. Pada contoh, semua perubahan itu menyentuh tabel demonstrasi saja, sehingga kontrak lima tabel bersama tetap utuh.

POIN PEMBAHASAN

- `TRUNCATE` menghapus seluruh baris tanpa `WHERE` dan bukan pengganti `DELETE` yang dapat diuji dengan rollback.
- `DROP` menghapus definisi tabel; foreign key dapat menghalangi penghapusan parent.
- Oracle melakukan implicit commit terkait DDL, sehingga jangan menyisipkan DDL ke demonstrasi transaksi yang ingin dibatalkan.
- Selesaikan DML sebelumnya dengan keputusan eksplisit.

- Sesudah latihan periksa USER_TAB_COLUMNS dan USER_CONSTRAINTS untuk melihat metadata, bukan menebak struktur dari tampilan data.

Alur merancang tabel

- 1 Tentukan entitas dan kolom
- 2 Pilih tipe data
- 3 Tetapkan primary/foreign key
- 4 Tambahkan aturan nilai
- 5 Periksa metadata hasil DDL

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.

Produk sebelum dan sesudah perubahan struktur

Tabel produk_demo memisahkan identitas, nama, harga dan stok. ALTER menambah kolom, bukan menambah penjualan.

Contoh SQL / PL/SQL

```
-- Opsional: hanya pada schema latihan dan kolom yang belum ada.  
ALTER TABLE produk_demo ADD catatan VARCHAR2(100);
```

Kolom	Contoh nilai	Aturan
id_produk	11	PRIMARY KEY : tidak boleh NULL /duplikat
nama_produk	Kopi	NOT NULL
harga	30000	CHECK harga > 0
stok	20	DEFAULT 0; NOT NULL ; CHECK stok >= 0

Keadaan	Kolom catatan	Baris Kopi
Sebelum ALTER	Belum ada	Tetap ada
Sesudah ALTER	Ada; NULL pada baris lama	Tetap ada

Cara membaca contoh

1. Misalkan Kopi sudah ada: INSERT lagi dengan id_produk 11 melanggar primary key.
2. Harga -1 melanggar CHECK ; stok negatif ditolak. Nama produk tetap harus diisi.
3. ALTER menambah kolom catatan yang awalnya NULL pada baris lama. Jalankan hanya jika kolom belum ada.

Yang perlu dipahami: TRUNCATE mengosongkan baris tetapi struktur tetap; DROP menghapus objek. Keduanya berbeda dari DELETE untuk eksperimen rollback.

Video pembelajaran

Video utama — fokus bab ini

CREATE TABLE dan constraints Oracle; dilengkapi video ALTER, TRUNCATE dan DROP agar cakupan DDL tidak hanya perubahan kolom.

CREATE TABLE WITH CONSTRAINTS - ORACLE - SQL

R.N.A. Creation · [Buka video utama di YouTube](#)

Video tambahan sesuai subtopik

1. [Chapter 2 : Addition, deletion and modification of column, truncating and dropping the table. #DBMS](#)

College Tech with Shanu

Fokus: ALTER TABLE: tambah/ubah/hapus kolom, TRUNCATE TABLE dan DROP TABLE.

Batas cakupan dan panduan menonton: Contoh video utama memakai Oracle10g. Ikuti tipe data, relasi, RENAME dan aturan implicit commit pada kode halaman untuk XE/FreeSQL.

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.

Langkah 1 — DDL lengkap dataset bersama: jalankan sekali

```
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)
);
```

Persis lima tabel pada setup.json. Tidak ada DROP otomatis. Urutan parent sebelum child; tabel kosong sampai modul03. Jangan ulang jika tabel telah dibuat lewat setup indeks.

Langkah 2 — Seluruh keluarga DDL pada tabel terpisah

```
CREATE TABLE ddl_latihan_demo (
  id NUMBER(4) PRIMARY KEY,
  kode CHAR(2), catatan VARCHAR2(30),
  waktu TIMESTAMP, uraian CLOB
);
ALTER TABLE ddl_latihan_demo ADD (aktif CHAR(1) DEFAULT 'Y');
ALTER TABLE ddl_latihan_demo MODIFY (catatan VARCHAR2(60));
ALTER TABLE ddl_latihan_demo RENAME COLUMN catatan TO keterangan;
ALTER TABLE ddl_latihan_demo DROP COLUMN uraian;
INSERT INTO ddl_latihan_demo (id,kode,keterangan) VALUES
(1, 'AA', 'Percobaan');
COMMIT;
RENAME ddl_latihan_demo TO ddl_latihan_baru_demo;
SELECT COUNT(*) AS sebelum FROM ddl_latihan_baru_demo;
TRUNCATE TABLE ddl_latihan_baru_demo;
SELECT COUNT(*) AS sesudah FROM ddl_latihan_baru_demo;
DROP TABLE ddl_latihan_baru_demo;
```

Dijalankan setelah setup dan transaksi lain selesai. Sebelum truncate diprediksi satu baris, sesudahnya nol; DROP menghapus objek demonstrasi. DDL bukan latihan rollback.

Langkah 3 — Inspeksi struktur dan constraint

```
SELECT table_name, column_name, data_type, nullable
FROM user_tab_columns
WHERE table_name IN
('PELANGGAN_DEMO', 'PRODUK_DEMO', 'PENJUALAN_DEMO', 'DETAIL_JUAL_DEMO',
'PEGAWAI_DEMO')
ORDER BY table_name, column_id;
SELECT table_name, constraint_name, constraint_type
FROM user_constraints
WHERE table_name IN
('PELANGGAN_DEMO', 'PRODUK_DEMO', 'PENJUALAN_DEMO', 'DETAIL_JUAL_DEMO',
'PEGAWAI_DEMO')
ORDER BY table_name, constraint_name;
```

P berarti primary key, R referential constraint, U unique, C check termasuk not-null. Nama constraint yang dihasilkan Oracle dapat berbeda antar schema.

Tugas / latihan

1. Pada XE lokal ATAU FreeSQL login, jalankan DDL bersama sekali atau buktikan setup yang sudah ada sesuai; lampirkan metadata lima tabel.
2. Gambar relasi beserta kunci gabungan dan self-reference pegawai; jelaskan penjualan anonim dan pelanggan tanpa penjualan.
3. Jalankan demonstrasi DDL pada tabel terpisah dan jelaskan perbedaan DROP, TRUNCATE, DELETE serta risiko implicit commit.

Pertanyaan analisis

1. Mengapa CHECK harga > 0 perlu dipasangkan dengan NOT NULL?
2. Apakah UNIQUE email mewajibkan setiap pelanggan memiliki email? Jelaskan.

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 Database 19c SQL Language Reference: CREATE-TABLE](#)
- [Oracle Database 19c SQL Language Reference: ALTER-TABLE](#)
- [Oracle Database 19c SQL Language Reference: TRUNCATE-TABLE](#)
- [Oracle Database 19c SQL Language Reference: DROP-TABLE](#)
- [Oracle Database 19c SQL Language Reference: RENAME](#)
- [Oracle Database 19c SQL Language Reference: Data-Types](#)
- [Oracle Database 19c SQL Language Reference: constraint](#)

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