

ORACLE PL/SQL · PERTEMUAN 06 / 14

JOIN Query

Membaca hubungan tabel tanpa kehilangan baris atau menggandakan total.

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

Versi PDF materi pertemuan 06. Contoh dan tabel dapat dibaca di PDF. Kuis dapat dijalankan di halaman web: [Buka materi daring](#).

Tujuan pembelajaran

1. Memetakan relasi primary key dan foreign key pada transaksi penjualan.
2. Memilih INNER, LEFT OUTER, RIGHT OUTER dan FULL OUTER JOIN sesuai pertanyaan.
3. Menggunakan CROSS JOIN dan SELF JOIN dengan alias yang jelas.
4. Menggabungkan pelanggan, penjualan, detail dan produk serta memeriksa granularitas hasil.

Pokok bahasan / praktikum

- Relasi tabel dan kardinalitas
- INNER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN
- CROSS JOIN
- SELF JOIN
- JOIN beberapa tabel
- Predikat ON dan WHERE
- NULL dan granularitas laporan

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. Pahami SELECT, WHERE, NULL dan agregasi.

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 nama pelanggan tidak perlu disalin ke setiap detail transaksi?
2. Apakah pelanggan yang belum membeli harus muncul pada laporan?
3. Bagaimana menampilkan nama manajer yang tersimpan sebagai ID pegawai?

Uji pemahaman awal

1. JOIN apa yang mempertahankan semua pelanggan di sisi kiri?

- INNER JOIN
- LEFT OUTER JOIN
- CROSS JOIN
- SELF JOIN selalu

2. Mengapa filter status di WHERE dapat menghilangkan pelanggan tanpa transaksi?

- WHERE membuat foreign key baru
- NULL selalu sama dengan LUNAS
- Predikat status tidak TRUE untuk baris NULL hasil outer join
- GROUP BY diperlukan untuk LEFT JOIN

3. Kondisi SELF JOIN manajer yang benar adalah ...

- e.id_manajer = m.id_pegawai
- e.nama = m.nama
- e.id_pegawai = m.id_pegawai saja
- Tanpa ON

PAHAMI KONSEP · BACA TERSTRUKTUR · TERAPKAN

Teori dan pembahasan

Sebelum memilih jenis JOIN, tentukan arti satu baris hasil: pelanggan, transaksi atau item transaksi. Hubungan tabel dan granularitas harus sejalan.

Konsep inti bab ini

- ON menjelaskan pasangan baris; outer join mempertahankan baris tanpa pasangan.
- Filter pada ON dan WHERE dapat memberi hasil outer join yang berbeda.
- Gunakan harga historis detail untuk menghitung nilai transaksi.

Urutan pembahasan

- 01 Relasi menentukan arti satu baris
- 02 INNER JOIN untuk pasangan yang ditemukan
- 03 LEFT dan RIGHT OUTER JOIN
- 04 FULL OUTER JOIN dan ketidaksesuaian dua arah
- 05 CROSS JOIN dan SELF JOIN
- 06 JOIN beberapa tabel dan audit laporan

Gambaran konsep dalam satu tabel

JOIN	Baris yang dipertahankan	Pemakaian
INNER	Hanya pasangan yang cocok	Transaksi dengan relasi ditemukan
LEFT / RIGHT	Semua baris sisi yang dipilih	Pelanggan tanpa transaksi
FULL OUTER	Baris cocok dan tidak cocok dua arah	Audit ketidaksesuaian
CROSS	Semua kombinasi	Daftar penawaran, bukan omzet
SELF	Tabel yang sama dengan dua alias	Pegawai dan manajer

01 Relasi menentukan arti satu baris

Tabel `pelanggan_demo` menyimpan identitas pelanggan, sedangkan `penjualan_demo` menyimpan kepala transaksi.

Foreign key `id_pelanggan` menghubungkan transaksi dengan pelanggan, tetapi nilainya boleh `NULL` untuk pembelian anonim. `detail_jual_demo` menghubungkan transaksi dengan `produk_demo` melalui dua foreign key; pasangan `id_jual` dan `id_produk` merupakan primary key. Karena satu transaksi dapat mempunyai beberapa detail, JOIN bukan sekadar menempelkan nama kolom: ia mengubah banyaknya baris yang sedang dilaporkan.

POIN PEMBAHASAN

- Tentukan dahulu apakah satu baris mewakili pelanggan, transaksi, atau item transaksi.
- Gunakan alias singkat dan tulis nama kolom dengan awalan alias agar id yang sama tidak ambigu.
- Sebelum praktik, periksa setup bersama; latihan tidak menghapus atau membuat ulang data.

02 INNER JOIN untuk pasangan yang ditemukan

INNER JOIN hanya menampilkan pasangan yang memenuhi kondisi **ON**.

Hubungkan `p.id_pelanggan` dengan `j.id_pelanggan`, bukan nama pelanggan atau urutan baris. Pada seed awal, Ayu dan Bima mempunyai transaksi, sedangkan Citra belum mempunyai transaksi dan penjualan anonim tidak mempunyai pasangan pelanggan.

POIN PEMBAHASAN

- Karena itu, kedua kelompok terakhir tidak muncul pada inner join tersebut.
- Ini adalah prediksi atas seed, bukan hasil eksekusi Oracle yang sudah diverifikasi.
- Menambahkan `WHERE j.status = 'LUNAS'` menyaring pasangan yang telah terbentuk.
- `SELECT *` kurang baik untuk laporan karena membuat asal kolom kurang jelas.
- Pilih ID dan nama yang dibutuhkan, lalu gunakan `ORDER BY` untuk urutan yang dapat dibandingkan.

03 LEFT dan RIGHT OUTER JOIN

LEFT OUTER JOIN mempertahankan semua baris tabel di sebelah kiri, termasuk baris yang tidak memperoleh pasangan.

Kolom dari sisi kanan menjadi `NULL` pada baris itu. Untuk daftar semua pelanggan, letakkan `pelanggan_demo` di kiri.

POIN PEMBAHASAN

- RIGHT OUTER JOIN mempertahankan sisi kanan; `penjualan_demo` RIGHT OUTER JOIN `pelanggan_demo` dapat menyatakan kebutuhan yang sama.
- Penempatan filter penting: status LUNAS dalam ON tetap mempertahankan pelanggan tanpa penjualan lunas, sedangkan syarat yang sama dalam `WHERE` menolak baris dengan status `NULL`.
- Jangan mengganti seluruh `NULL` dengan nol; ID yang tidak ditemukan bukan ID nol.
- Gunakan `NVL` hanya untuk label atau nilai laporan yang memang mempunyai makna pengganti.

04 FULL OUTER JOIN dan ketidaksesuaian dua arah

FULL OUTER JOIN mempertahankan baris tanpa pasangan dari kedua sisi.

Pada hubungan pelanggan dan penjualan, laporan dapat menunjukkan pelanggan tanpa pembelian sekaligus transaksi anonim. Tampilkan ID pelanggan dari tabel pelanggan dan ID pelanggan yang tercatat pada transaksi agar asal informasi tetap terbaca.

POIN PEMBAHASAN

- Foreign key mencegah referensi ke ID pelanggan yang tidak ada, tetapi tidak menolak `NULL` ketika kolomnya nullable.
- Jadi, baris anonim bukan bukti foreign key rusak.
- Gunakan `IS NULL` untuk menemukan ketidakcocokan; perbandingan `= NULL` tidak menghasilkan `TRUE`.
- Oracle juga memperlakukan string karakter kosong sebagai `NULL`, sehingga label kosong tidak dapat digunakan sebagai identitas pelanggan alternatif.

05 CROSS JOIN dan SELF JOIN

CROSS JOIN membentuk semua pasangan tanpa kondisi kecocokan.

Pasangkan pelanggan dan produk untuk rancangan daftar penawaran, bukan untuk menghitung omzet. Besarnya hasil bertambah sesuai banyaknya kombinasi, sehingga periksa ukuran tabel sebelum menerapkannya pada data besar.

POIN PEMBAHASAN

- SELF JOIN memakai tabel yang sama dengan dua alias berbeda.
- Pada `pegawai_demo`, alias `e` mewakili pegawai dan alias `m` mewakili manajer; `e.id_manajer` cocok dengan `m.id_pegawai`.
- LEFT JOIN mempertahankan pimpinan Sari yang tidak mempunyai manajer.
- Satu self join hanya menampilkan hubungan langsung, bukan seluruh rantai organisasi.
- Hindari mencocokkan nama pegawai, sebab dua orang dapat mempunyai nama yang sama.

06 JOIN beberapa tabel dan audit laporan

Laporan rincian menggabungkan penjualan, pelanggan, detail, lalu produk.

Pertahankan penjualan anonim dengan LEFT JOIN ke pelanggan, tetapi gunakan INNER JOIN ke detail dan produk ketika pertanyaan hanya meminta item yang benar-benar tercatat. Harga transaksi diambil dari `detail_jual_demo.harga`, bukan harga katalog yang bisa berubah.

POIN PEMBAHASAN

- Perkalian jumlah dengan harga menunjukkan nilai baris detail.
- Untuk total per transaksi, `GROUP BY id_jual` setelah join atau agregasikan detail terlebih dahulu.
- Jangan menggunakan `DISTINCT` sebagai obat umum bagi join yang salah: duplikasi dapat menandakan kondisi ON hilang atau granularitas yang tidak dipahami.
- Bandingkan laporan dengan kepala transaksi dan detailnya, dokumentasikan perubahan data lokal, serta bedakan hasil aktual dari prediksi seed awal.

Alur memeriksa JOIN

- 1 Tentukan granularitas laporan
- 2 Pilih tabel dan alias
- 3 Tulis kondisi ON
- 4 Pilih jenis JOIN dan filter
- 5 Bandingkan ID serta jumlah baris

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.

Mengapa Citra muncul pada LEFT JOIN tetapi tidak INNER JOIN?

Pertanyaan laporan: tampilkan seluruh pelanggan, termasuk yang belum membeli.

Contoh SQL / PL/SQL

```
SELECT p.nama, j.id_jual
FROM pelanggan_demo p
LEFT JOIN penjualan_demo j
  ON j.id_pelanggan=p.id_pelanggan
ORDER BY p.id_pelanggan;
```

INNER JOIN: pelanggan	id_jual
Ayu	101

INNER JOIN: pelanggan	id_jual
Bima	102

LEFT JOIN: pelanggan	id_jual
Ayu	101
Bima	102
Citra	NULL

Cara membaca contoh

1. INNER JOIN hanya menghasilkan Ayu–101 dan Bima–102 karena keduanya mempunyai pasangan.
2. LEFT JOIN mempertahankan Citra, lalu kolom penjualan untuk Citra diisi `NULL`.
3. Transaksi 103 yang tidak punya pelanggan tidak muncul pada LEFT JOIN dari pelanggan; FULL OUTER JOIN dapat menampilkannya.

Yang perlu dipahami: Pilih JOIN berdasarkan kebutuhan laporan. Jangan menganggap `NULL` pada hasil outer join sebagai nilai 0.

Video pembelajaran

Video utama — fokus bab ini

Konsep relasi, INNER/LEFT/RIGHT JOIN; bagian kedua melengkapi FULL OUTER, CROSS dan SELF JOIN.

SQL JOINS Tutorial for beginners | Practice SQL Queries using JOINS - Part 1

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

Video tambahan sesuai subtopik

1. [SQL JOINS Tutorial for beginners | Practice SQL Queries using JOINS - Part 2](#)
techTFQ

Fokus: FULL OUTER JOIN, CROSS JOIN dan SELF JOIN.

FULL OUTER mulai 00:54; CROSS 10:20; SELF JOIN 21:51.

Batas cakupan dan panduan menonton: Tutorial menjelaskan konsep SQL JOIN umum. Gunakan nama tabel, kondisi ON dan studi kasus penjualan Oracle pada halaman; NATURAL JOIN dalam referensi bukan tuntutan tugas.

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 — INNER JOIN pelanggan dan penjualan

```
SELECT j.id_jual, p.id_pelanggan, p.nama, j.status
FROM penjualan_demo j
INNER JOIN pelanggan_demo p ON p.id_pelanggan = j.id_pelanggan
ORDER BY j.id_jual;
```

Prediksi seed: Ayu dan Bima ditemukan; pelanggan tanpa transaksi dan transaksi anonim tidak muncul.

Langkah 2 — LEFT OUTER dan RIGHT OUTER yang ekuivalen

```
SELECT p.id_pelanggan, p.nama, j.id_jual
FROM pelanggan_demo p
LEFT OUTER JOIN penjualan_demo j ON j.id_pelanggan =
p.id_pelanggan
ORDER BY p.id_pelanggan, j.id_jual;

SELECT p.id_pelanggan, p.nama, j.id_jual
FROM penjualan_demo j
RIGHT OUTER JOIN pelanggan_demo p ON j.id_pelanggan =
p.id_pelanggan
ORDER BY p.id_pelanggan, j.id_jual;
```

Keduanya mempertahankan Citra pada seed. Kolom id_jual miliknya NULL, bukan nol.

Langkah 3 — FULL OUTER JOIN dan filter ON versus WHERE

```
SELECT p.id_pelanggan, p.nama, j.id_jual, j.id_pelanggan AS
pelanggan_transaksi
FROM pelanggan_demo p
FULL OUTER JOIN penjualan_demo j ON j.id_pelanggan =
p.id_pelanggan
ORDER BY p.id_pelanggan NULLS LAST, j.id_jual;
```

```

SELECT p.nama, j.id_jual
FROM pelanggan_demo p
LEFT JOIN penjualan_demo j
  ON j.id_pelanggan = p.id_pelanggan AND j.status = 'LUNAS'
ORDER BY p.id_pelanggan;

SELECT p.nama, j.id_jual
FROM pelanggan_demo p
LEFT JOIN penjualan_demo j ON j.id_pelanggan = p.id_pelanggan
WHERE j.status = 'LUNAS'
ORDER BY p.id_pelanggan;

```

FULL mempertahankan Citra dan penjualan anonim. Dua SELECT berikutnya sengaja berbeda: filter ON mempertahankan semua pelanggan, filter WHERE hanya pasangan lunas.

Langkah 4 — CROSS JOIN dan SELF JOIN manajer

```

SELECT p.nama, r.nama_produk
FROM pelanggan_demo p CROSS JOIN produk_demo r
ORDER BY p.id_pelanggan, r.id_produk;

SELECT e.id_pegawai, e.nama AS pegawai, m.nama AS manajer
FROM pegawai_demo e
LEFT JOIN pegawai_demo m ON m.id_pegawai = e.id_manajer
ORDER BY e.id_pegawai;

```

CROSS menghasilkan kombinasi penawaran. SELF JOIN mempertahankan Sari dengan manajer NULL dan memasangkan Rudi/Tono dengan Sari.

Langkah 5 — JOIN empat tabel dan total transaksi

```

SELECT j.id_jual, NVL(p.nama, 'Anonim') AS pelanggan,
       r.nama_produk, d.jumlah, d.harga,
       d.jumlah * d.harga AS nilai_detail
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
ORDER BY j.id_jual, r.id_produk;

```

```

SELECT j.id_jual, NVL(p.nama, 'Anonim') AS pelanggan,
       SUM(d.jumlah * d.harga) AS total
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
GROUP BY j.id_jual, p.nama
ORDER BY j.id_jual;

```

Query pertama bergranularitas detail; kedua transaksi. Transaksi tanpa detail tidak muncul karena JOIN detail bersifat inner.

Tugas / latihan

1. Pada XE lokal ATAU FreeSQL schema pribadi, buat laporan semua produk termasuk yang belum terjual dengan LEFT JOIN; tampilkan ID agar NULL mudah diperiksa.
2. Buat laporan empat tabel dan total per transaksi; jelaskan granularitas dan sumber harga transaksi.
3. Serahkan SQL dan hasil aktual INNER/LEFT/FULL serta SELF JOIN manajer. Bandingkan filter ON dan WHERE tanpa mengubah seed.

Pertanyaan analisis

1. Mengapa COUNT(*) dan COUNT(j.id_jual) dapat berbeda setelah LEFT JOIN? Hubungkan jawaban dengan baris tanpa pasangan.
2. Jika satu transaksi mempunyai beberapa detail, kapan penjumlahan setelah JOIN menjadi benar dan kapan join tambahan menggandakan total?

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 SQL Reference — Joins](#)
- [Oracle 19c SQL Reference — 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.
