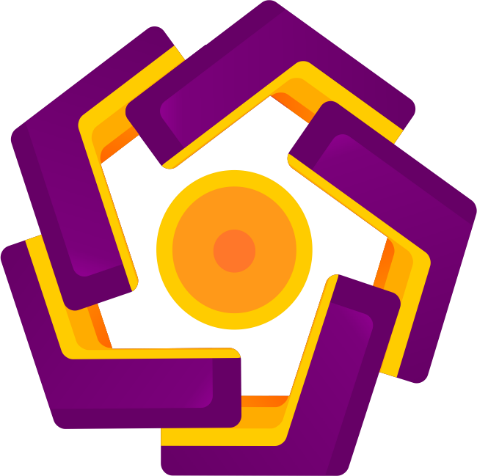
LAPORAN PRAKTIKUM PEMROGRAMAN BASIS DATA

PERTEMUAN PRAKTIKUM KE-[4]

Dosen: [Novi Prisma Yunita, M.Kom]



[Jenderal Nicolas Tampubolon]

[22.11.4856]

PROGRAM STUDI S1 INFORMATIKA, FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM YOGYAKARTA

2023

[DATA RETRIEVAL WITH JOIN]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Buat JOIN statement untuk menampilkan nama item dan kategorinya, urutkan secara ascending berdasarkan nama item. \*Hanya baris yang sesuai kondisi JOIN saja yang ditampilkan

Jawab:

|  |
| --- |
| select name, explanation from items inner join categories using (category\_id) order by name asc |



SELECT name, explanation: Ini adalah bagian dari pernyataan SELECT yang menentukan kolom-kolom mana yang akan ditampilkan dalam hasil query. Dalam hal ini, kita memilih kolom "name" dari tabel "items" dan "explanation" (yang diasumsikan ada di tabel "categories").

FROM items: Ini menentukan tabel mana yang akan digunakan sebagai sumber data untuk query. Dalam hal ini, tabel "items" dipilih sebagai sumber data utama.

INNER JOIN categories: Ini adalah jenis join yang digunakan untuk menggabungkan data dari tabel "items" dengan data dari tabel "categories".

USING (category\_id): Ini menentukan kolom mana yang akan digunakan sebagai kunci gabung antara kedua tabel. Dalam hal ini, kolom "category\_id" dipilih sebagai kunci gabung.

ORDER BY name ASC: Ini menentukan cara pengurutan hasil query. Dalam hal ini, hasil query akan diurutkan secara alfabetis berdasarkan kolom "name" dari tabel "items" secara ascending (ASC).

1. Buat JOIN statement 3 tabel untuk menampilkan nama item, nama kategori, dan nama supplier. \*Hanya baris yang sesuai kondisi JOIN saja yang ditampilkan.

Jawab:

|  |
| --- |
| SELECT i.name as 'Nama Items', c.explanation as 'Category', s.name as 'Nama Supplier' from items.i inner join categories.c using(category\_id) inner join suppliers.s using(supplier\_id) |



SELECT i.name as 'Nama Items', c.explanation as 'Category', s.name as 'Nama Supplier': Ini adalah bagian dari pernyataan SELECT yang menentukan kolom-kolom mana yang akan ditampilkan dalam hasil query. Dalam hal ini, kita memilih kolom "name" dari tabel "items" dengan alias 'Nama Items', kolom "explanation" dari tabel "categories" dengan alias 'Category', dan kolom "name" dari tabel "suppliers" dengan alias 'Nama Supplier'.

FROM items.i: Ini adalah salah satu kesalahan sintaks SQL. Penulisan "items.i" tidak benar. Jika "items" adalah nama tabel, maka cukup tuliskan "items" saja sebagai nama tabel, tanpa perlu menambahkan ".i". Dengan demikian, ini seharusnya ditulis sebagai FROM items i.

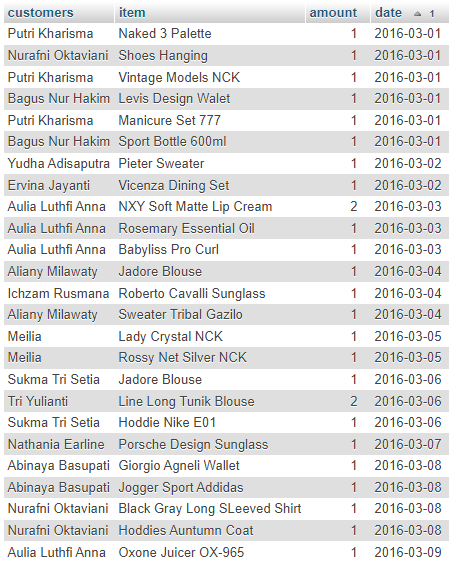
INNER JOIN categories.c USING(category\_id): Ini adalah bagian dari pernyataan JOIN yang menggabungkan tabel "items" dengan tabel "categories" menggunakan kolom "category\_id" yang sama. Keyword USING digunakan untuk menggabungkan tabel berdasarkan kolom yang memiliki nama yang sama di kedua tabel. Namun, penulisan "categories.c" adalah kesalahan sintaks SQL. Seharusnya ditulis sebagai categories c, di mana "c" adalah alias untuk tabel "categories".

INNER JOIN suppliers.s USING(supplier\_id): Ini adalah bagian lain dari pernyataan JOIN yang menggabungkan tabel "items" dengan tabel "suppliers" menggunakan kolom "supplier\_id" yang sama. Seperti sebelumnya, penulisan "suppliers.s" adalah kesalahan sintaks SQL. Seharusnya ditulis sebagai suppliers s, di mana "s" adalah alias untuk tabel "suppliers".

1. Buat JOIN statement untuk menampilkan nama customer, nama dan jumlah item yang dibeli, serta tanggal pembeliannya, kemudian urutkan berdasarkan tanggal pembelian secara ascending.

Jawab:

|  |
| --- |
| SELECT c.name as customers, i.name as item, a.amount, p.date from customers c join purchases p using (customer\_id) join purchase\_items a using (purchase\_id) join items i using (item\_id) order by p.date asc |



SELECT c.name as customers, i.name as item, a.amount, p.date: Ini adalah bagian dari pernyataan SELECT yang menentukan kolom-kolom mana yang akan ditampilkan dalam hasil query. Dalam hal ini, kita memilih kolom "name" dari tabel "customers" dengan alias 'customers', kolom "name" dari tabel "items" dengan alias 'item', kolom "amount" dari tabel "purchase\_items", dan kolom "date" dari tabel "purchases".

FROM customers c: Ini menentukan tabel mana yang akan digunakan sebagai sumber data untuk query. Dalam hal ini, tabel "customers" dipilih sebagai sumber data utama dengan alias "c".

JOIN purchases p USING (customer\_id): Ini adalah pernyataan JOIN yang menggabungkan tabel "customers" dengan tabel "purchases" menggunakan kolom "customer\_id" yang sama. Keyword USING digunakan untuk menggabungkan tabel berdasarkan kolom yang memiliki nama yang sama di kedua tabel.

JOIN purchase\_items a USING (purchase\_id): Ini adalah pernyataan JOIN lainnya yang menggabungkan tabel "purchases" dengan tabel "purchase\_items" menggunakan kolom "purchase\_id" yang sama.

JOIN items i USING (item\_id): Ini adalah pernyataan JOIN yang menggabungkan tabel "purchase\_items" dengan tabel "items" menggunakan kolom "item\_id" yang sama.

ORDER BY p.date ASC: Ini menentukan cara pengurutan hasil query. Dalam hal ini, hasil query akan diurutkan secara kronologis berdasarkan kolom "date" dari tabel "purchases" secara ascending (ASC).

1. Buat LEFT JOIN statement untuk menampilkan nama customer dan purchase\_id, urutkan berdasarkan nama customernya secara ascending. Perhatikan result-set yang terbentuk. Di bagian mana baris dengan klausa NULL berada, kenapa demikian?

Jawab:

|  |
| --- |
| SELECT c.name, p.purchase\_id  FROM customers c  LEFT JOIN purchases p USING(customer\_id)  ORDER BY c.name ASC; |

****

SELECT c.name, p.purchase\_id: Ini adalah bagian dari pernyataan SELECT yang menentukan kolom-kolom mana yang akan ditampilkan dalam hasil query. Dalam hal ini, kita memilih kolom "name" dari tabel "customers" dan kolom "purchase\_id" dari tabel "purchases".

FROM customers c: Ini menentukan tabel mana yang akan digunakan sebagai sumber data untuk query. Dalam hal ini, tabel "customers" dipilih sebagai sumber data utama dengan alias "c".

LEFT JOIN purchases p USING(customer\_id): Ini adalah pernyataan LEFT JOIN yang menggabungkan tabel "customers" dengan tabel "purchases" menggunakan kolom "customer\_id" yang sama. LEFT JOIN mengambil semua baris dari tabel di sebelah kiri (tabel "customers" dalam hal ini) dan baris yang cocok dari tabel di sebelah kanan (tabel "purchases" dalam hal ini), atau NULL jika tidak ada baris yang cocok di tabel di sebelah kanan.

ORDER BY c.name ASC: Ini menentukan cara pengurutan hasil query. Dalam hal ini, hasil query akan diurutkan secara alfabetis berdasarkan kolom "name" dari tabel "customers" secara ascending (ASC).

1. Buat RIGHT JOIN statement untuk menampilkan nama item dan nama supplier, urutkan berdasarkan nama item secara ascending. Perhatikan result-set yang terbentuk. Di bagian mana baris dengan klausa NULL berada, kenapa demikian?

Jawab:

|  |
| --- |
| SELECT i.name, s.name  FROM items i  RIGHT JOIN suppliers s USING(supplier\_id)  ORDER BY s.name ASC; |



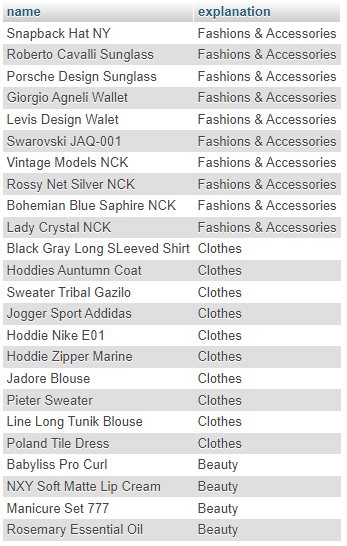
SELECT i.name, s.name: Ini adalah bagian dari pernyataan SELECT yang menentukan kolom-kolom mana yang akan ditampilkan dalam hasil query. Dalam hal ini, kita memilih kolom "name" dari tabel "items" dan kolom "name" dari tabel "suppliers" FROM items i: Ini menentukan tabel mana yang akan digunakan sebagai sumber data untuk query. Dalam hal ini, tabel "items" dipilih sebagai sumber data utama dengan alias "i". RIGHT JOIN suppliers s USING(supplier\_id): Ini adalah pernyataan RIGHT JOIN yang menggabungkan tabel "items" dengan tabel "suppliers" menggunakan kolom "supplier\_id" yang sama. RIGHT JOIN mengambil semua baris dari tabel di sebelah kanan (tabel "suppliers" dalam hal ini) dan baris yang cocok dari tabel di sebelah kiri (tabel "items" dalam hal ini), atau NULL jika tidak ada baris yang cocok di tabel di sebelah kiri.

ORDER BY s.name ASC: Ini menentukan cara pengurutan hasil query. Dalam hal ini, hasil query akan diurutkan secara alfabetis berdasarkan kolom "name" dari tabel "suppliers" secara ascending (ASC).

1. INNER JOIN dan CROSS JOIN bersifat syntactic equivalent (dapat menggantikan satu sama lain). Ubah CROSS JOIN berikut ke dalam bentuk INNER JOIN. Pastikan result-set yang dihasilkan sama

Jawab:

|  |
| --- |
| select name, explanation from items inner join categories using (category\_id) |



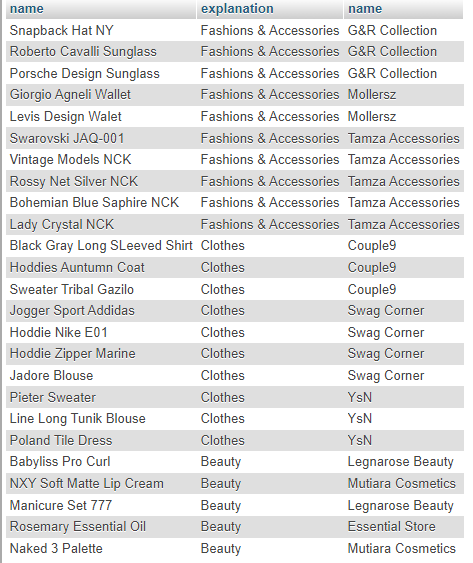
Kode di atas hanya perlu mengganti CROSS JOIN menjadi INNER JOIN dan setelah dirunning hasilnya tetap sama.

1. Ubah JOIN statement berikut sehingga dapat berhasil dieksekusi!

|  |
| --- |
| SELECT name, explanation, name FROM items  JOIN categories USING(category\_id)  JOIN suppliers USING(supplier\_id); |

Jawab:

|  |
| --- |
| select name, explanation from items inner join categories using (category\_id) |



Kode di atas hanya perlu ditambahkan inisial agar kode tidak ambigu, dikarenakan kesamaan nama variable dari masing-masing tabel