

Advanced Database Transact-SQL



Name

Dicha Zelianivan Arkana

NIM

2241720002

Class

2i

Department

Information Technology

Study Program

D4 Informatics Engineering

1 Practicum - Part 1

1.1 Part 1

1. **SELECT**
 *
FROM Sales.Customers;

SELECT
 custid,
 companyname, contactname, contacttitle,
 address, city, region, postalcode, country,
 phone, fax
FROM Sales.Customers;

Apa perbedaannya dengan hasil pada langkah kedua diatas?

Query pertama menggunakan wildcard * yang berarti akan melakukan seleksi pada semua field. Sedangkan query kedua hanya melakukan seleksi pada field yang ditentukan.

1.2 Part 2

2. **SELECT** country **FROM** Sales.Customers;

Apakah ada data yang terduplikasi? Jika YA mengapa? Capture hasil eksekusi script SQL diatas.



The screenshot shows a SQL query result window with 91 rows. The column selected is 'country'. The results are as follows:

	country
1	Germany
2	Mexico
3	Mexico
4	UK
5	Sweden
6	Germany
7	France
8	Spain
9	France
10	Canada
11	UK

Karena data dari field **country** tidaklah unik untuk setiap record. Sehingga apabila kita melakukan seleksi pada field **country** saja, maka akan terlihat seolah olah data yg didapat merupakan duplikat dikarenakan field lain yang tidak terlihat.

3. `SELECT DISTINCT country FROM Sales.Customers;`

Apakah ada data yang terduplikasi? Jelaskan perbedaan hasil pada langkah tahap 4 dan tahap 3! Apa manfaat dari perintah `DISTINCT`? Capture hasil eksekusi script SQL diatas!



The screenshot shows a SQL query result window with a table titled 'country'. The table contains 11 rows of data, each representing a country. The countries listed are Argentina, Austria, Belgium, Brazil, Canada, Denmark, Finland, France, Germany, Ireland, and Italy. The window also shows a header bar with '21 rows' and navigation icons.

	country
1	Argentina
2	Austria
3	Belgium
4	Brazil
5	Canada
6	Denmark
7	Finland
8	France
9	Germany
10	Ireland
11	Italy

Tidak ada data yang terduplikasi. Perbedaan hasil pada tahap 4 dan 3 adalah pada tahap 4 kita melakukan seleksi pada field `country` menggunakan perintah `DISTINCT`. Perintah `DISTINCT` hanya akan mengembalikan hasil apabila data yang didapat unik.

1.3 Part 5

4. `SELECT`
 `c.contactname, c.contacttitle`
`FROM Sales.Customers AS c;`

`SELECT`
 `c.contactname AS Name,`
 `c.contacttitle AS Title,`
 `c.companyname AS [Company Name]`
`FROM Sales.Customers AS c;`

Apa yang membedakan hasil eksekusi dari query tahap 1 dan tahap 3 diatas? Apa manfaat dari perintah `AS`? Silahkan Jelaskan! Capture hasil eksekusi script SQL diatas!

	contactname	contacttitle
1	Allen, Michael	Sales Representative
2	Hassall, Mark	Owner
3	Peoples, John	Owner
4	Arndt, Torsten	Sales Representative
5	Higginbotham, Tom	Order Administrator
6	Poland, Carole	Sales Representative
7	Ransel, Dushvant	Marketing Manager

	Name	Title	[Company Name]
1	Allen, Michael	Sales Representative	Customer NRZBB
2	Hassall, Mark	Owner	Customer MLTDN
3	Peoples, John	Owner	Customer KBUDE
4	Arndt, Torsten	Sales Representative	Customer HFBZG
5	Higginbotham, Tom	Order Administrator	Customer HGVLZ
6	Poland, Carole	Sales Representative	Customer XHXJV
7	Ransel, Dushvant	Marketing Manager	Customer OYVI A

Perintah AS digunakan untuk memberikan alias pada field yang dipilih. Pada tahap 1 kita melakukan seleksi menggunakan tanpa menggunakan alias sehingga label tiap column akan mengikuti nama field pada tabel. Sedangkan pada tahap 3 kita memberikan alias pada tiap field yang dipilih sehingga label tiap column akan mengikuti alias yang diberikan.

1.4 Part 4

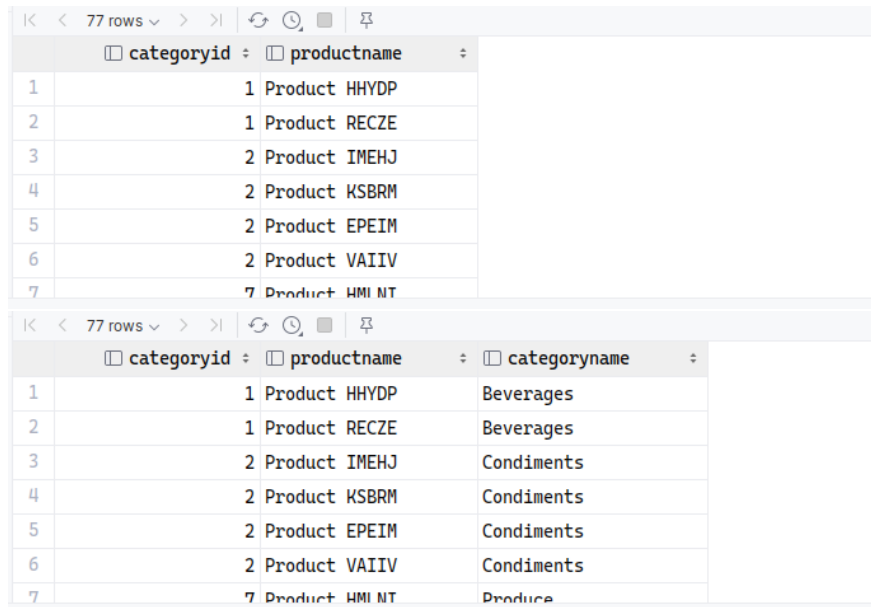
5. SELECT

```
p.categoryid, p.productname
FROM Production.Products AS p;
```

SELECT

```
p.categoryid,
p.productname,
CASE
    WHEN p.categoryid = 1 THEN 'Beverages'
    WHEN p.categoryid = 2 THEN 'Condiments'
    WHEN p.categoryid = 3 THEN 'Confections'
    WHEN p.categoryid = 4 THEN 'Dairy Products'
    WHEN p.categoryid = 5 THEN 'Grains/Cereals'
    WHEN p.categoryid = 6 THEN 'Meat/Poultry'
    WHEN p.categoryid = 7 THEN 'Produce'
    WHEN p.categoryid = 8 THEN 'Seafood'
    ELSE 'Other'
END AS categoryname
FROM Production.Products AS p;
```

Apa yang membedakan hasil eksekusi dari query tahap 1 dan tahap 3 diatas?
Apa manfaat dari perintah CASE? Silahkan Jelaskan! Capture hasil eksekusi script SQL diatas



	categoryid	productname
1	1	Product HHYDP
2	1	Product RECZE
3	2	Product IMEHJ
4	2	Product KSBRM
5	2	Product EPEIM
6	2	Product VAIIV
7	7	Product HMI NT

	categoryid	productname	categoryname
1	1	Product HHYDP	Beverages
2	1	Product RECZE	Beverages
3	2	Product IMEHJ	Condiments
4	2	Product KSBRM	Condiments
5	2	Product EPEIM	Condiments
6	2	Product VAIIV	Condiments
7	7	Product HMI NT	Produce

Pada tahap 1, query hanya melakukan seleksi pada field `categoryid` dan `productname`. Sedangkan pada tahap 3, query melakukan seleksi pada field yang sama dengan tambahan field baru yang diberi nama `categoryname`. Field ini merupakan hasil dari perintah `CASE` yang melakukan pengkondisian untuk mengubah angka menjadi teks berdasarkan value dari field `categoryid`

6. `SELECT`

```
p.categoryid, p.productname,
CASE
    WHEN p.categoryid = 1 THEN 'Beverages'
    WHEN p.categoryid = 2 THEN 'Condiments'
    WHEN p.categoryid = 3 THEN 'Confections'
    WHEN p.categoryid = 4 THEN 'Dairy Products'
    WHEN p.categoryid = 5 THEN 'Grains/Cereals'
    WHEN p.categoryid = 6 THEN 'Meat/Poultry'
    WHEN p.categoryid = 7 THEN 'Produce'
    WHEN p.categoryid = 8 THEN 'Seafood'
    ELSE 'Other'
END AS categoryname,
CASE
    WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
    ELSE 'Non-Campaign Products'
```

```

END AS iscampaigh
FROM Production.Products AS p

```

Silahkan capture hasilnya, data apa yang didapatkan dari perintah query diatas?
Jelaskan!

	categoryid	productname	categoryname	iscampaigh
1	1	Product HHYDP	Beverages	Campaign Products
2	1	Product RECZE	Beverages	Campaign Products
3	2	Product IMEHJ	Condiments	Non-Campaign Products
4	2	Product KSBRM	Condiments	Non-Campaign Products
5	2	Product EPEIM	Condiments	Non-Campaign Products
6	2	Product VAIIV	Condiments	Non-Campaign Products
7	7	Product HMI NT	Produce	Campaign Products

Pada tahap ini, query menambahkan field baru bernama `iscampaigh` yang berisi data apakah produk tersebut merupakan produk yang sedang diiklankan atau tidak. Field ini merupakan hasil dari perintah `CASE` yang melakukan pengkondisian sama seperti pada tahap 5, namun alih alih mendefinisikan setiap kasus, kita menggunakan perintah `IN` untuk mendefinisikan beberapa kasus sekaligus, lalu mengabaikan sisanya menggunakan perintah `ELSE`.

- Capture perintah SQL anda dan berapa jumlah row yang dihasilkan!

```

SELECT
  p.categoryid AS ID_KATEGORI,
  p.productname AS NAMA_PRODUK,
  CASE
    WHEN p.categoryid = 1 THEN 'Beverages'
    WHEN p.categoryid = 2 THEN 'Condiments'
    WHEN p.categoryid = 3 THEN 'Confections'
    WHEN p.categoryid = 4 THEN 'Dairy Products'
    WHEN p.categoryid = 5 THEN 'Grains/Cereals'
    WHEN p.categoryid = 6 THEN 'Meat/Poultry'
    WHEN p.categoryid = 7 THEN 'Produce'
    WHEN p.categoryid = 8 THEN 'Seafood'
    ELSE 'Other'
  END AS NAMA_KATEGORI,
  CASE
    WHEN p.categoryid IN (1, 7, 8) THEN 'Campaign Products'
    ELSE 'Non-Campaign Products'
  END AS STATUS
FROM Production.Products AS p
WHERE p.categoryid = 7;

```

ID_KATEGORI	NAMA_PRODUK	NAMA_KATEGORI	STATUS
7	Product HMLNI	Produce	Campaign Products
7	Product PWCJB	Produce	Campaign Products
7	Product OFBNT	Produce	Campaign Products
7	Product APITJ	Produce	Campaign Products
7	Product BKAZJ	Produce	Campaign Products

Terdapat 5 row yang dihasilkan dari query diatas.

- Tampilkan data employees dari tabel `HR.Employees` yang berasal dari negara 'USA' dan kota 'Seattle', gunakan perintah `ALIAS` untuk merubah nama kolom seperti gambar dibawah ini. Capture perintah SQL anda

```
SELECT
    e.firstname AS FIRST_NAME,
    e.lastname AS LAST_NAME,
    e.city AS CITY,
    e.country AS COUNTRY
FROM HR.Employees AS e
WHERE country = N'USA' AND city = N'Seattle';
```

FIRST_NAME	LAST_NAME	CITY	COUNTRY
Sara	Davis	Seattle	USA
Maria	Cameron	Seattle	USA

1.5 Part 5

- Tulis perintah `SELECT` yang akan mengembalikan nilai pada kolom `custid`, `companyname`, `contactname`, `address`, `city`, `country`, and `phone` pada tabel `Sales.Customers`, kemudian filter hasilnya hanya untuk "Brazil, UK dan USA" (Gunakan predikat `IN` dalam klausa `WHERE`).

```
SELECT
    c.custid,
    c.companyname,
    c.contactname,
    c.address,
    c.city,
    c.country,
    c.phone
FROM Sales.Customers AS c
WHERE country IN (N'Brazil', N'UK', N'USA');
```

	custid	companyname	contactname	address	city	country
1	4	Customer HFBZG	Arndt, Torsten	7890 Hanover Sq.	London	UK
2	11	Customer UBHAU	Jaffe, David	Fauntleroy Circus 4567	London	UK
3	15	Customer JUWKK	Richardson, Shawn	Av. dos Lusadas, 6789	Sao Paulo	Brazil
4	16	Customer GYBBY	Birkby, Dana	Berkeley Gardens 0123 Brewery	London	UK
5	19	Customer RFNQC	Boseman, Randall	5678 King George	London	UK
6	21	Customer KIDPX	Russo, Giuseppe	Rua Ores, 3456	Sao Paulo	Brazil

10. Salin Kode T-SQL pada tahap ke-4 kemudian modifikasi dengan operator perbandingan untuk kolom city pada clause **WHERE** dengan operator **OR**. Setelah itu eksekusi kode tersebut, tunjukkan hasilnya!

```
SELECT
    c.custid,
    c.companyname,
    o.orderid
FROM Sales.Customers AS c
LEFT OUTER JOIN Sales.Orders AS o
    ON c.custid = o.custid OR c.city = 'Paris';
```

	custid	companyname	orderid
1	1	Customer NRZBB	10643
2	1	Customer NRZBB	10692
3	1	Customer NRZBB	10702
4	1	Customer NRZBB	10835
5	1	Customer NRZBB	10952
6	1	Customer NRZBB	11011
7	2	Customer MI TNN	1A308

1.6 Part 6

11. Tuliskan perintah **SELECT** untuk mengambil kolom **custid**, **custname** dari tabel **Sales.Customers** dan kolom **orderid**, **orderdate** dari tabel **Sales.Orders**! Filter hasilnya hanya untuk pesanan pada atau setelah 1 April 2008. Kemudian urutkan hasilnya berdasarkan **orderdate** secara descending (menurun) dan **custid** ascending (menaik)!

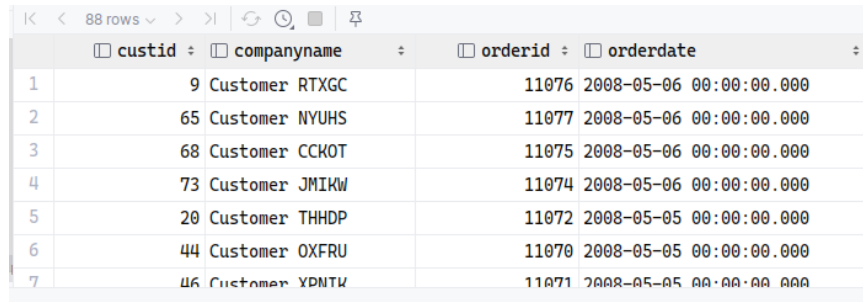
```
SELECT
    c.custid,
    c.companyname,
    o.orderid,
    o.orderdate
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
```



```

ON c.custid = o.custid
WHERE o.orderdate >= '2008-04-01'
ORDER BY o.orderdate DESC, c.custid ASC;

```



	custid	companyname	orderid	orderdate
1	9	Customer RTXGC	11076	2008-05-06 00:00:00.000
2	65	Customer NYUHS	11077	2008-05-06 00:00:00.000
3	68	Customer CCKOT	11075	2008-05-06 00:00:00.000
4	73	Customer JMIKW	11074	2008-05-06 00:00:00.000
5	20	Customer THHDP	11072	2008-05-05 00:00:00.000
6	44	Customer OXFRU	11070	2008-05-05 00:00:00.000
7	46	Customer YDNTU	11071	2008-05-05 00:00:00.000

12. Eksekusi perintah T-SQL pada tahap 3. Apakah terjadi kesalahan? Apa pesan errornya? Menurut Anda, apakah penyebabnya?

```

SELECT
    e.empid, e.lastname, e.firstname, e.title, e.mgrid,
    m.lastname AS mgrlastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE
    mgrlastname = N'Buck';

```

Terjadi kesalahan dengan pesan error:

Invalid column name 'mgrlastname'.

Hal ini disebabkan karena perintah WHERE tidak dapat mengakses field menggunakan alias.

13. Lakukan perubahan perintah T-SQL untuk memperbaiki kesalahan pada uji coba ke-3, kemudian lakukan eksekusi! Bandingkan hasil eksekusi dengan hasil berikut. Jika sama, maka hasil uji coba sudah benar.

```

SELECT
    e.empid, e.lastname, e.firstname, e.title, e.mgrid,
    m.lastname AS mgrlastname, m.firstname AS mgrfirstname
FROM HR.Employees AS e
INNER JOIN HR.Employees AS m ON e.mgrid = m.empid
WHERE
    e.mgrlastname = N'Buck';

```

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	6	Suurs	Paul	Sales Representative	5	Buck	Sven
2	7	King	Russell	Sales Representative	5	Buck	Sven
3	9	Dolgopyatova	Zoya	Sales Representative	5	Buck	Sven

14. Salin perintah T-SQL pada uji coba 4, dan modifikasi sehingga menghasilkan semua karyawan **ORDER BY** nama depan manajer. Pada awalnya uji coba dengan menggunakan nama asal tabel, kemudian lakukan uji coba menggunakan nama alias tabel! Eksekusi T-SQL tersebut dan bandingkan dengan hasil berikut. Jika Hasilnya sama, maka uji coba sudah benar

	empid	lastname	firstname	title	mgrid	mgrlastname	mgrfirstname
1	3	Lew	Judy	Sales Manager	2	Funk	Don
2	5	Buck	Sven	Sales Manager	2	Funk	Don
3	4	Peled	Yael	Sales Representative	3	Lew	Judy
4	8	Cameron	Maria	Sales Representative	3	Lew	Judy
5	2	Funk	Don	Vice President, Sales	1	Davis	Sara
6	6	Suurs	Paul	Sales Representative	5	Buck	Sven
7	7	Vinn	Russell	Sales Representative	5	Buck	Sven

15. Kenapa kita dapat menggunakan nama kolom sesuai nama asli tabel ataupun menggunakan nama alias tabel?

Karena nama alias tabel merupakan nama baru yang diberikan pada tabel yang digunakan untuk mempermudah penulisan query.

1.7 Part 7

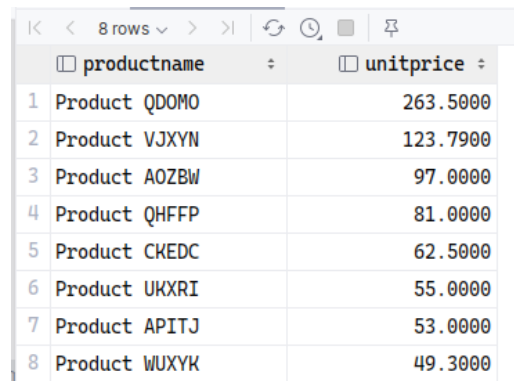
16. Tuliskan perintah **SELECT** untuk menampilkan kolom **productname** and **unitprice** pada tabel **Production.Products** yang diurutkan secara menurun berdasarkan **unitprice**! Tampilkan hasil eksekusinya!

```
SELECT
    p.productname,
    p.unitprice
FROM Production.Products As p
ORDER BY unitprice DESC;
```

	productname	unitprice
1	Product QDOMO	263.5000
2	Product VJXYN	123.7900
3	Product AOZBW	97.0000
4	Product QHFFP	81.0000
5	Product CKEDC	62.5000
6	Product UKXRI	55.0000
7	Product ADTT1	53.0000

-
17. Salin dan modifikasi perintah T-SQL pada uji coba 2 dengan batasan hanya 8 produk yang akan ditampilkan berdasar pemesanan **unitprice**! Eksekusi perintah tersebut, dan bandingkan apakah sudah sesuai dengan hasil berikut.

```
SELECT TOP 8
    p.productname,
    p.unitprice
FROM Production.Products As p
ORDER BY unitprice DESC;
```



	productname	unitprice
1	Product QDOMO	263.5000
2	Product VJXYN	123.7900
3	Product AOZBW	97.0000
4	Product QHFFP	81.0000
5	Product CKEDC	62.5000
6	Product UKXRI	55.0000
7	Product APITJ	53.0000
8	Product WUXYK	49.3000

18. Apakah memungkinkan mengimplementasikan perintah T-SQL uji coba 5 menggunakan klausa **OFFSET-FETCH**?

```
SELECT
    p.productname,
    p.unitprice
FROM Production.Products As p
ORDER BY unitprice DESC
OFFSET 0 ROWS
FETCH NEXT 8 ROWS ONLY;
```

1.8 Part 8

19. Tuliskan perintah **SELECT** untuk menampilkan kolom **custid**, **orderid**, and **orderdate** pada tabel **Sales.Orders**. Urutkan baris berdasarkan **orderdate** dan **orderid**. Ambil 20 baris pertama. Eksekusi perintah tersebut dan bandingkan hasilnya berikut. Jika hasilnya sama, maka uji coba Anda sudah benar.

```
SELECT TOP 20
    custid,
    orderid,
    orderdate
FROM Sales.Orders
ORDER BY orderdate, orderid;
```

	custid	orderid	orderdate
1	85	10248	2006-07-04 00:00:00.000
2	79	10249	2006-07-05 00:00:00.000
3	34	10250	2006-07-08 00:00:00.000
4	84	10251	2006-07-08 00:00:00.000
5	76	10252	2006-07-09 00:00:00.000
6	34	10253	2006-07-10 00:00:00.000
7	14	10254	2006-07-11 00:00:00.000
8	68	10255	2006-07-12 00:00:00.000

20. Tuliskan perintah **SELECT** untuk menampilkan hasil yang sama dengan soal no. 19, lewati 20 baris awal, dan lanjutkan dengan 20 baris selanjutnya menggunakan klausa **OFFSETFETCH**! Eksekusi perintah tersebut dan bandingkan dengan hasil berikut. Jika hasilnya sama, maka uji coba Anda sudah benar.

```

SELECT
    custid,
    orderid,
    orderdate
FROM Sales.Orders
ORDER BY orderdate, orderid
OFFSET 20 ROWS
FETCH NEXT 20 ROWS ONLY;

```

	custid	orderid	orderdate
1	33	10268	2006-07-30 00:00:00.000
2	89	10269	2006-07-31 00:00:00.000
3	87	10270	2006-08-01 00:00:00.000
4	75	10271	2006-08-01 00:00:00.000
5	65	10272	2006-08-02 00:00:00.000
6	63	10273	2006-08-05 00:00:00.000
7	85	10274	2006-08-06 00:00:00.000
8	49	10275	2006-08-07 00:00:00.000

1.9 Part 9

21. Tulis sebuah SQL yang menampilkan hasil pada praktikum-1 langkah-1 & 2 secara sekaligus (gabungan) dengan menggunakan **UNION**!

```

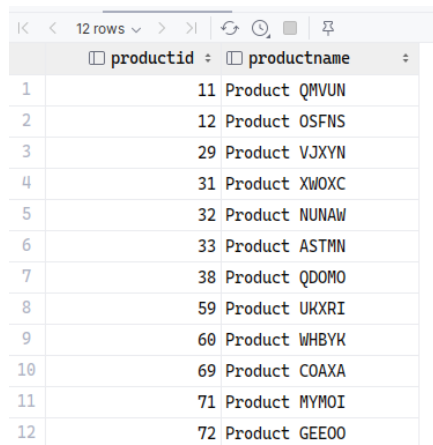
SELECT
    productid,
    productname
FROM
    Production.Products
WHERE categoryid = 4

```

```

UNION
SELECT
    P.productid,
    P.productname
FROM Production.Products P
INNER JOIN Sales.OrderDetails OD
    ON P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING SUM(OD.qty * OD.unitprice) > 50000;

```



The screenshot shows a database query result with 12 rows. The columns are 'productid' and 'productname'. The data is as follows:

productid	productname
11	Product QMVUN
12	Product OSFNS
29	Product VJXYN
31	Product XWOXC
32	Product NUNAW
33	Product ASTMN
38	Product QDOMO
59	Product UKXRI
60	Product WHBYK
69	Product COAXA
71	Product MYMOI
72	Product GEE00

22. Serupa dengan langkah sebelumnya, kali ini tulislah sebuah SQL yang menampilkan hasil pada praktikum-1 langkah-1 & 2 secara sekaligus (gabungan) dengan menggunakan UNION ALL!

```

SELECT
    productid,
    productname
FROM
    Production.Products
WHERE categoryid = 4
UNION ALL
SELECT
    P.productid,
    P.productname
FROM Production.Products P
INNER JOIN Sales.OrderDetails OD
    ON P.productid = OD.productid
GROUP BY
    P.productid, P.productname
HAVING SUM(OD.qty * OD.unitprice) > 50000;

```

productid		productname	
1	11	Product	QMVUN
2	12	Product	OSFNS
3	31	Product	XWOXC
4	32	Product	NUNAW
5	33	Product	ASTMN
6	59	Product	UKXRI
7	60	Product	WHBYK
8	69	Product	COAXA
9	71	Product	MYMOI
10	72	Product	GEEEO
11	29	Product	VJXYN
12	38	Product	QDOMO
13	59	Product	UKXRI
14	60	Product	WHBYK

23. Apa bedanya UNION & UNION ALL?

- **UNION:** Hanya mengembalikan record yang memiliki nilai unik
- **UNION ALL:** Mengembalikan keseluruhan record tanpa mempedulikan hasil yang duplikat

24. Tuliskan SQL untuk menampilkan 10 pelanggan yang membeli paling awal serta 10 pelanggan yang membeli paling akhir.

```
SELECT TOP 10
    c.custid,
    c.companyname,
    o.orderdate
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
    ON c.custid = o.custid
ORDER BY o.orderdate DESC;
```

```
SELECT TOP 10
    c.custid,
    c.companyname,
    o.orderdate
FROM Sales.Customers AS c
INNER JOIN Sales.Orders AS o
    ON c.custid = o.custid
ORDER BY o.orderdate ASC;
```