



Fundusze Europejskie
Wiedza Edukacja Rozwój



Politechnika Wrocławska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Hurtownie danych

Operatory grupujące SQL. Funkcje agregujące i grupujące.

dr inż. Bernadetta Maleszka



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Politechnika Wrocławska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

SQL - podstawy

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Politechnika Wrocławska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

SQL - podstawy

SELECT <nazwy kolumn>

FROM <tabela>

WHERE <warunki>

GROUP BY <atrybuty>

HAVING <warunki dot. atrybutów zagregowanych>

ORDER BY <atrybuty>

SQL – kolejność wykonywania poleceń

SELECT

FROM

WHERE

GROUP BY


HAVING

ORDER BY

SQL - przykład

```
SELECT SalesPersonID,  
       COUNT(DISTINCT CustomerID) "Liczba klientow",  
       COUNT(*) AS "Liczba zamowien"  
FROM Sales.SalesOrderHeader  
WHERE Year(OrderDate)=2012  
GROUP BY SalesPersonID  
HAVING COUNT(*)>10  
ORDER BY 1,2
```

SalesOrderHeader (Sales)

	SalesOrderID
	RevisionNumber
	OrderDate
	DueDate
	ShipDate
	Status
	OnlineOrderFlag
	SalesOrderNumber
	PurchaseOrderNumber
	AccountNumber
	CustomerID
	SalesPersonID
	TerritoryID
	BillToAddressID
	ShipToAddressID
	ShipMethodID
	CreditCardID
	CreditCardApprovalCode
	CurrencyRateID
	SubTotal
	TaxAmt
	Freight
	TotalDue
	Comment
	rowguid
	ModifiedDate



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

	SalesPersonID	Liczba klientow	Liczba zamowien
1	NULL	2743	2743
2	274	22	22
3	275	73	148
4	276	54	151
5	277	87	166
6	278	28	80
7	279	57	153
8	280	26	45
9	281	25	74
10	282	47	86
11	283	24	63
12	284	16	24
13	289	48	111
14	290	18	42

Funkcje i operatory

- CASE, COALESCE, NULLIF
- CAST
- CTE
- PIVOT
- Zapytania z podsumowaniami – operatory:
 - ROLLUP
 - CUBE
 - GROUPING SETS, GROUPING, GROUPING_ID
- Funkcje okienkowe
 - OVER PARTITION
 - Tworzenie rankingów

Dane transakcyjne vs analityczne

	Order...	OrderDate	OrderA...	CustomerNa...
1	1	2012-03-01	10.0000	Joe
2	2	2012-03-01	11.0000	Sam
3	3	2012-03-02	10.0000	Beth
4	4	2012-03-02	15.0000	Joe
5	5	2012-03-02	17.0000	Sam
6	6	2012-03-03	12.0000	Joe
7	7	2012-03-04	10.0000	Beth
8	8	2012-03-04	18.0000	Sam
9	9	2012-03-04	12.0000	Joe
10	10	2012-03-04	11.0000	Beth
11	11	2012-03-05	14.0000	Sam
12	12	2012-03-06	17.0000	Beth
13	13	2012-03-06	19.0000	Joe
14	14	2012-03-07	13.0000	Beth
15	15	2012-03-07	16.0000	Sam

	Sprzedawca	2011	2012	2013	2014
1	NULL	3863120,2134	6390599,9473	10732127,33	8372829,73
2	274	28926,2465	453524,5233	431088,7238	178584,3625
3	275	875823,8318	3375456,8947	3985374,8995	1057247,3786
4	276	1149715,3253	3834908,674	4111294,9056	1271088,5216
5	277	1311627,2918	4317306,5741	3396776,2674	1040093,4071
6	278	500091,8202	1283569,6294	1389836,8101	435948,9551
7	279	1521289,1881	2674436,3518	2188082,7813	787204,4289
8	280	648485,5862	1208264,3834	963420,5805	504932,044
9	281	967597,2899	2294210,5506	2387256,0616	777941,6519

CASE

CASE

WHEN *<warunek1>* THEN *<wynik1>*

WHEN *<warunek2>* THEN *<wynik2>*

...

WHEN *<warunekN>* THEN *<wynikN>*

ELSE *wynik*

END;

CASE

```
SELECT CustomerID,
       SUM(CASE WHEN DATEPART(quarter,OrderDate)=1 THEN TotalDue END) AS Q1,
       SUM(CASE WHEN DATEPART(quarter,OrderDate)=2 THEN TotalDue END) AS Q2,
       SUM(CASE WHEN DATEPART(quarter,OrderDate)=3 THEN TotalDue END) AS Q3,
       SUM(CASE WHEN DATEPART(quarter,OrderDate)=4 THEN TotalDue END) AS Q4
FROM Sales.SalesOrderHeader
GROUP BY CustomerID
```

	CustomerID	Q1	Q2	Q3	Q4
1	14324	5659.1783	NULL	NULL	NULL
2	22814	NULL	NULL	5.514	NULL
3	11407	NULL	NULL	NULL	59.659
4	28387	NULL	645.2869	NULL	NULL
5	19897	NULL	659.6408	NULL	NULL
6	15675	2580.1529	NULL	2699.9018	2682.9953
7	24165	666.8565	2699.9018	NULL	NULL
8	27036	NULL	NULL	8.0444	NULL
9	18546	NULL	NULL	NULL	32.5754
10	11453	3729.364	NULL	2633.1377	2673.0613
11	17195	2574.9042	NULL	NULL	1105.4834
12	17026	NULL	NULL	288.836	NULL
13	22768	NULL	NULL	NULL	663.5083

CASE - uwagi

1. Na podstawie analizy klauzuli THEN ustalany jest typ wyniku wyrażenia
2. Typ wartości musi być możliwy do wyznaczenia
3. Wynikiem jest wartość wyrażenia pierwszego spełnionego warunku WHEN
4. Można zagnieżdżać wyrażenie CASE
5. Jeżeli nie została jawnie określona klauzula ELSE wówczas domyślną postacią jest ELSE NULL

CASE - przykład

- Studenci (StudentID, Imię, Nazwisko, Kierunek, Płeć, Wiek)
- Ile kobiet i mężczyzn studiuje poszczególne kierunki?

SELECT Kierunek,

SUM(CASE WHEN Płeć = 'M' THEN 1 ELSE 0 END) AS „Mężczyźni”

SUM(CASE WHEN Płeć = 'K' THEN 1 ELSE 0 END) AS „Kobiety”

FROM Studenci

GROUP BY Kierunek;

COALESCE

COALESCE (<wyrażenie 1.>, <wyrażenie 2.>)

CASE

WHEN <wyrażenie 1.> IS NOT NULL

THEN <wyrażenie 1.>

ELSE <wyrażenie 2.>

END;

NULLIF

NULLIF (<wyrażenie 1.>, <wyrażenie 2.>)

CASE

WHEN <wyrażenie 1.> = <wyrażenie 2.>

THEN NULL

ELSE <wyrażenie 1.>

END;

Konwersja danych: CAST

- `CAST(<operand> AS <typ danych>)`
- `SELECT CAST('20250310' AS Date) "Dzisiejsze zajęcia";`
- `SELECT CAST(SYSDATETIME() AS Date) "Dzisiejsze zajęcia";`



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Politechnika Wrocławska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

CTE

- Common table expression
- nazwana tabela wirtualna zdefiniowana za pomocą wyrażenia zapytania

WITH <nazwa CTE> [(<lista kolumn>)]

AS

(

<definicja zapytania>

)

SELECT <lista kolumn>

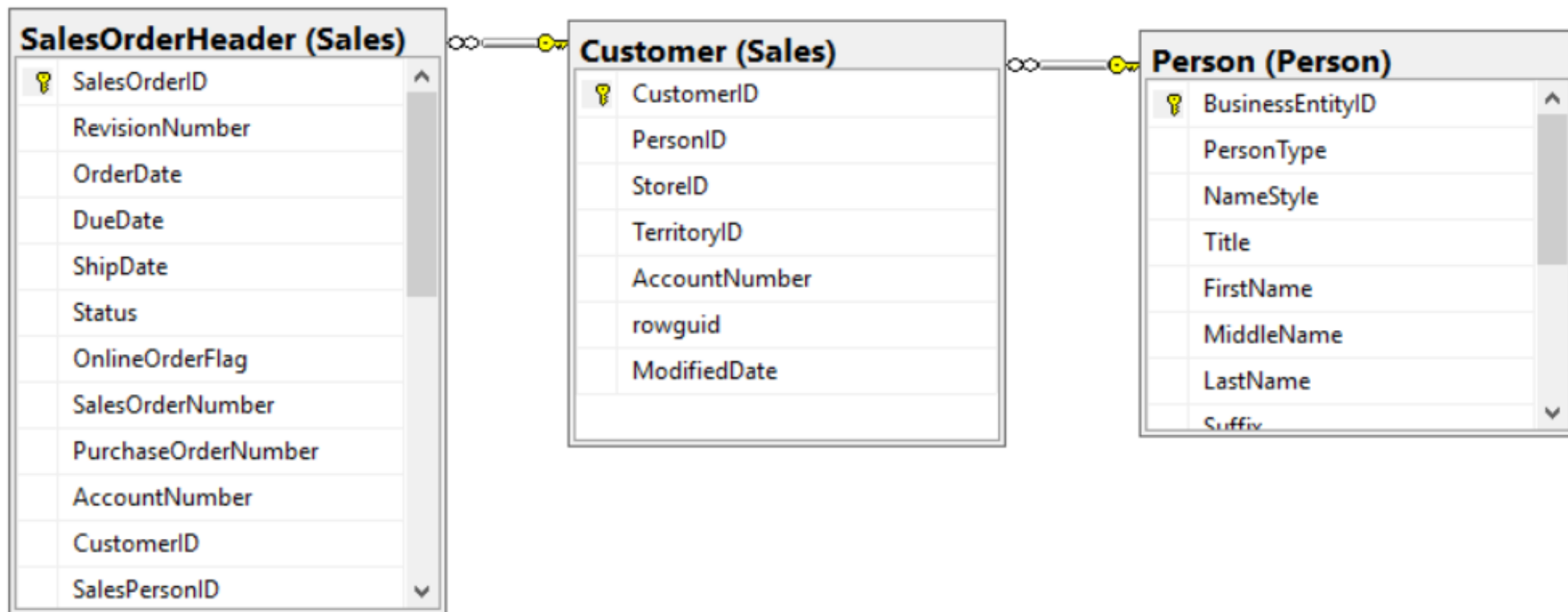
FROM <nazwa CTE>


CTE - przykład


- Utworzyć wykaz klientów, którzy mają więcej niż 25 zamówień w postaci:


klientId	Nazwisko, imię	Liczba zamówień
11091	Perez, Dalton	28
11176	Roberts, Mason	28
11185	Henderson, Ashley	27
11200	Griffin, Jason	27
...

CTE - przykład



SalesOrderHeader (Sales)	
	SalesOrderID
	RevisionNumber
	OrderDate
	DueDate
	ShipDate
	Status
	OnlineOrderFlag
	SalesOrderNumber
	PurchaseOrderNumber
	AccountNumber
	CustomerID
	SalesPersonID

Customer (Sales)	
	CustomerID
	PersonID
	StoreID
	TerritoryID
	AccountNumber
	rowguid
	ModifiedDate

Person (Person)	
	BusinessEntityID
	PersonType
	NameStyle
	Title
	FirstName
	MiddleName
	LastName
	Suffix

wska

oju Politechniki Wrocławskiej”

Unia Europejska
Europejski Fundusz Społeczny



CTE - przykład

```

SELECT S.CustomerID AS "klientID",
       CONCAT(P.LastName, ', ', P.FirstName) AS "Nazwisko, Imię",
       COUNT(*) AS "liczba zamówień"
FROM Sales.SalesOrderHeader s JOIN Sales.Customer c
      ON s.CustomerID = c.CustomerID
      JOIN Person.Person p ON p.BusinessEntityID = c.PersonID
GROUP BY s.CustomerID
HAVING COUNT(*) > 25;

```

SalesOrderHeader (Sales)	
SalesOrderID	
RevisionNumber	
OrderDate	
DueDate	
ShipDate	
Status	
OnlineOrderFlag	
SalesOrderNumber	
PurchaseOrderNumber	
AccountNumber	
CustomerID	
SalesPersonID	

Customer (Sales)	
CustomerID	
PersonID	
StoreID	
TerritoryID	
AccountNumber	
rowguid	
ModifiedDate	

Person (Person)	
BusinessEntityID	
PersonType	
NameStyle	
Title	
FirstName	
MiddleName	
LastName	
Suffix	

wska

oju Politechniki Wrocławskiej”

Unia Europejska
Europejski Fundusz Społeczny



CTE – przykład

```

SELECT S.CustomerID AS "klientID",
       CONCAT(P.LastName, ', ', P.FirstName) AS "Nazwisko, Imię",
       COUNT(*) AS "liczba zamówień"
FROM Sales.SalesOrderHeader s JOIN Sales.Customer c
      ON s.CustomerID = c.CustomerID
      JOIN Person.Person p ON p.BusinessEntityID = c.PersonID
GROUP BY s.CustomerID, CONCAT(P.LastName, ', ', P.FirstName)
HAVING COUNT(*) > 25;

```

CTE - przykład

```
with liczbaZam as  
(  
    SELECT S.CustomerID AS "KlientID", COUNT(*) AS "liczba zamówień"  
    FROM Sales.SalesOrderHeader S  
    GROUP BY S.CustomerID  
    HAVING COUNT(*) > 25  
)
```

CTE - przykład

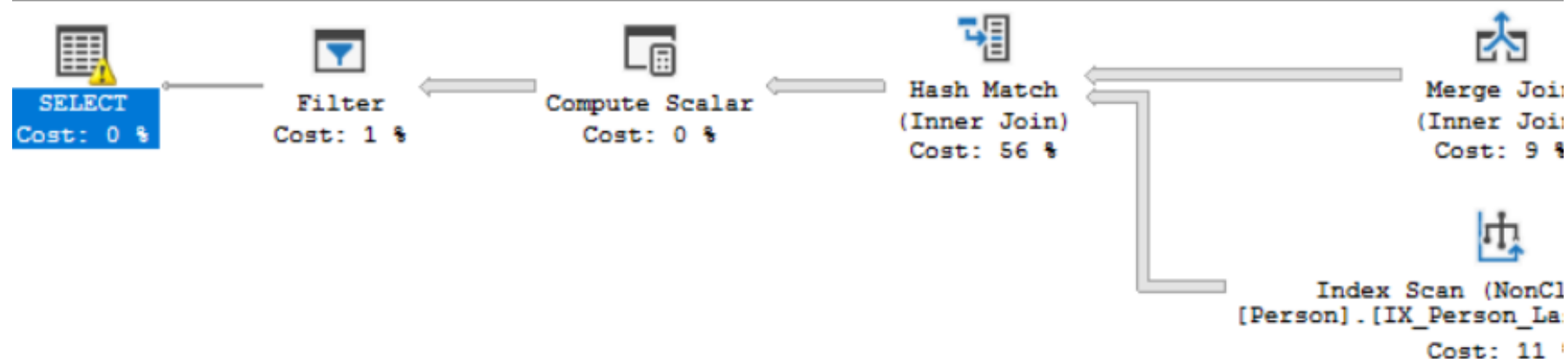
```
SELECT klientID,  
       CONCAT(P.LastName, ', ', P.FirstName) AS "Nazwisko, Imię"  
       , z.[liczba zamówień]  
FROM liczbaZam z JOIN Sales.Customer C  
     ON z.klientID = C.CustomerID  
     JOIN Person.Person P  
     ON P.BusinessEntityID = c.PersonID  
;
```

„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Results Messages Execution plan

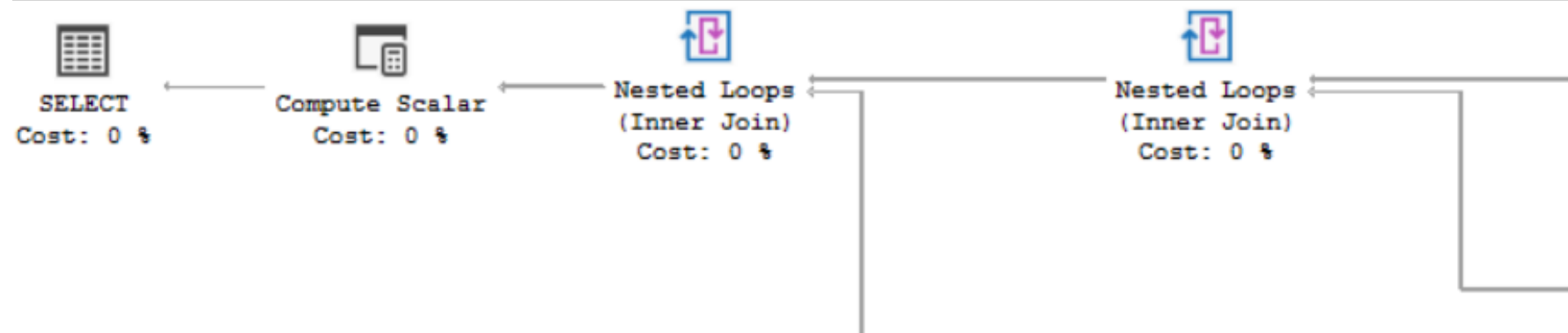
Query 1: Query cost (relative to the batch): 89%

SELECT S.CustomerID AS "klientID", CONCAT(P.LastName, ', ', P.FirstName) AS "Nazwisko"



Query 2: Query cost (relative to the batch): 11%

with liczbaZam as (SELECT S.CustomerID AS "KlientID", COUNT(*) AS "liczba zamówień"



PIVOT

- Grupowanie
- Rozpraszanie
- Agregowanie

niejawne grupowanie wierszy tabeli
na podstawie wszystkich jej kolumn
nie wymienionych jako argument operatora PIVOT

rozpraszanie wartości kolumny tabeli - CASE

agregacja dla każdego wyrażenia CASE

SELECT ...

FROM <źródło danych>

PIVOT (<funkcja agr.>(<element>)

FOR <element podziału> IN (<lista kolumn>)

)AS <alias tab. wynikowej>

PIVOT

```
SELECT SalesPersonID
    , CAST([2011] AS DEC(10,2)) [2011]
    , CAST([2012] AS DEC(10,2)) [2012]
    , CAST([2013] AS DEC(10,2)) [2013]
FROM (SELECT SalesPersonID
    , YEAR(OrderDate) AS Od, SubTotal
    FROM Sales.SalesOrderHeader) AS e
PIVOT (SUM(SubTotal)
    FOR Od IN ([2011],[2012], [2013])) AS X;
```

„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

PIVOT - wynik

SalesPersonID	2011	2012	2013
NULL	3863120.21	6390599.95	10732127.33
274	28926.25	453524.52	431088.72
275	875823.83	3375456.89	3985374.90
276	1149715.33	3834908.67	4111294.91
277	1311627.29	4317306.57	3396776.27
278	500091.82	1283569.63	1389836.81
279	1521289.19	2674436.35	2188082.78
280	648485.59	1208264.38	963420.58
281	967597.29	2294210.55	2387256.06

PIVOT - przykład

- Jaki jest wynik zapytania:

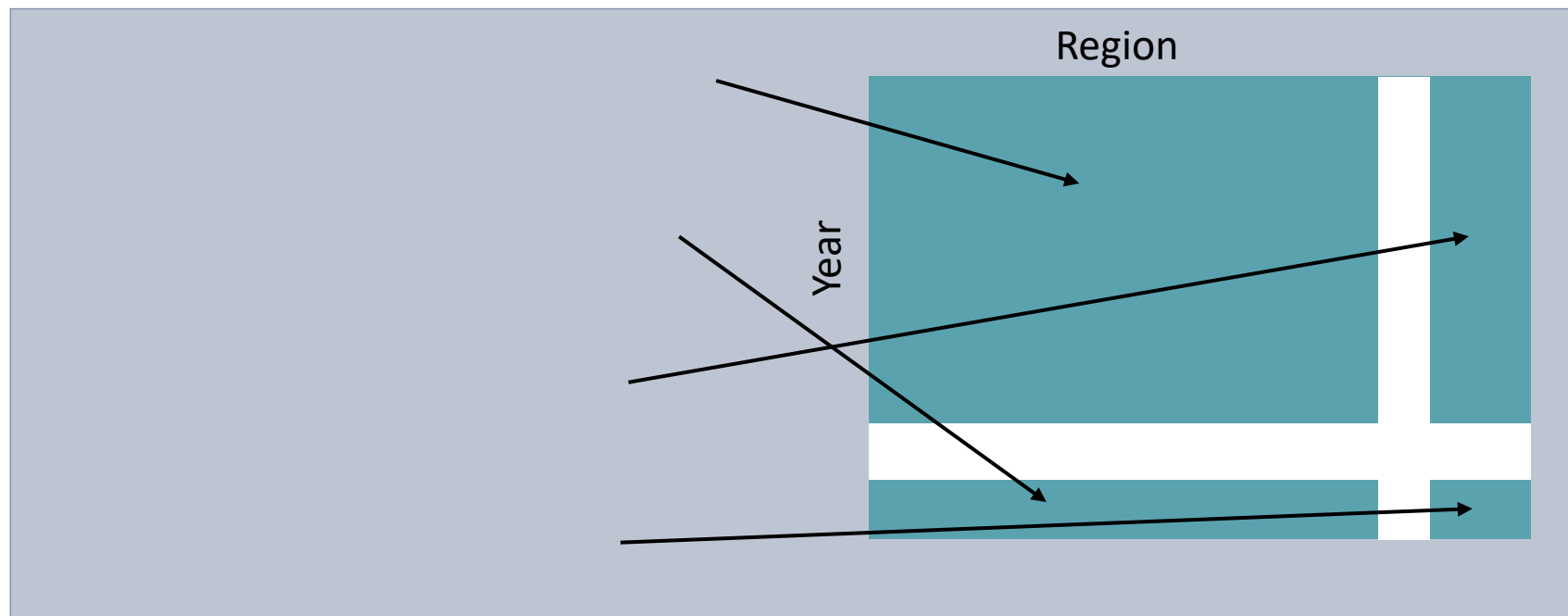
```
SELECT SalesPersonID, [29491], [29605]
FROM
    (SELECT SalesPersonID, CustomerID, SubTotal
     FROM Sales.SalesOrderHeader) S
PIVOT (
    SUM(SubTotal) FOR CustomerID IN ([29491], [29605])
) AS X
WHERE SalesPersonID IS NOT NULL;
```

PIVOT - przykład

SalesPersonID	29491	29605
278	NULL	NULL
275	53863,4443	NULL
284	NULL	NULL
281	NULL	218418,7057
287	NULL	NULL
290	NULL	NULL
282	NULL	NULL
274	33406,7043	29482,0603

Zapytania z podsumowaniami

- Sales (Region, Year, Income)
- UNION
- UNION ALL





Fundusze
Europejskie
Wiedza Edukacja Rozwój



Politechnika Wroclawska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Zapytania z podsumowaniami

- ROLLUP
- CUBE
- GROUPING SETS

ROLLUP

```
SELECT Region, Year, SUM(Income)
FROM Sales
GROUP BY ROLLUP(Region, Year);
```

CUBE

```
SELECT Region, Year, SUM(Income)
FROM Sales
GROUP BY CUBE(Region, Year);
```


GROUPING SETS

```
SELECT Region, Year, SUM(Income)
FROM Sales
GROUP BY GROUPING SETS(Region, Year);
```

Zapytania z podsumowaniami

GROUP BY (a, b, c)

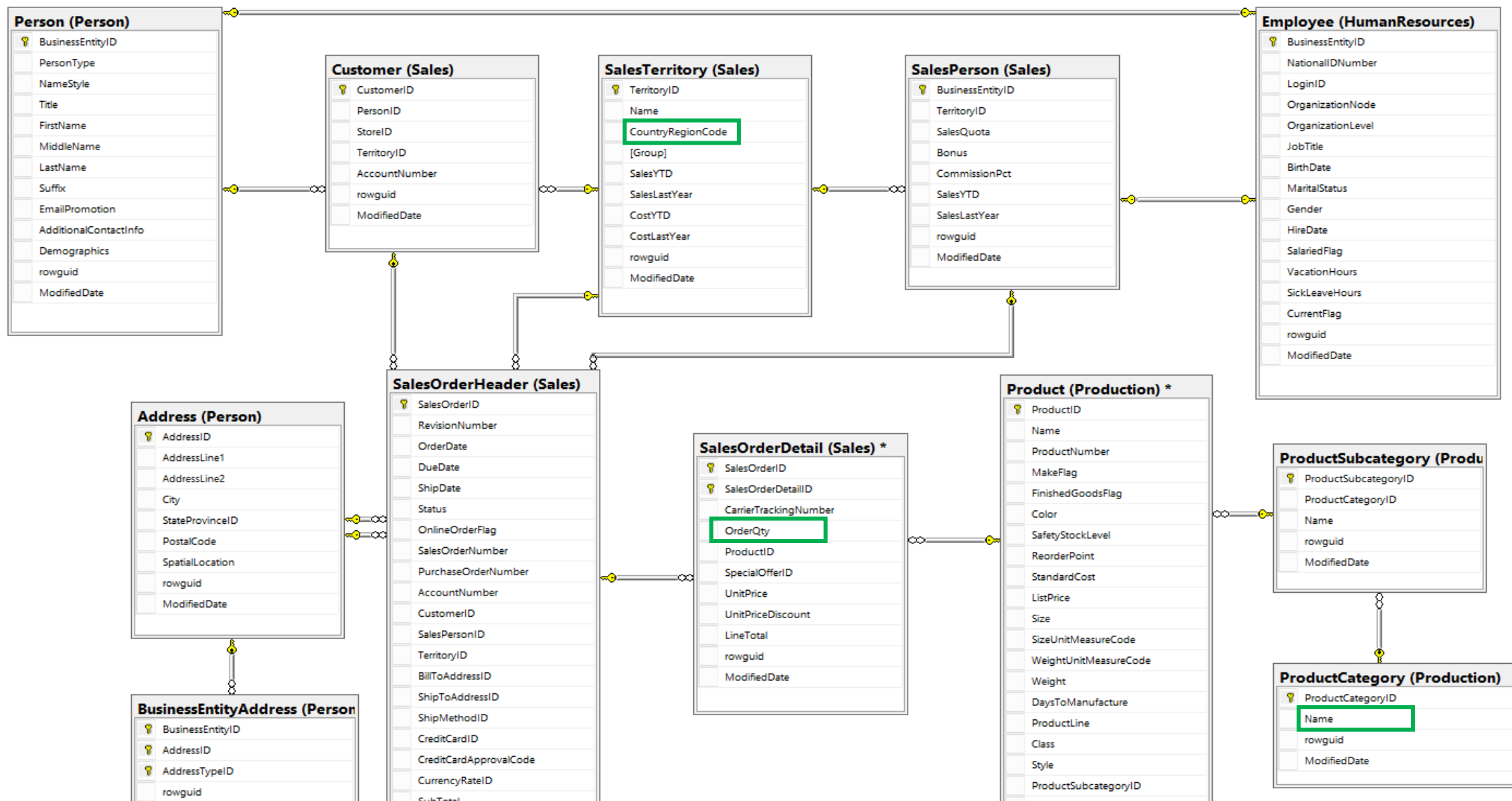
- ROLLUP
 - (), (a), (a, b), (a, b, c)
- CUBE
 - (), (a), (b), (c), (a, b), (a, c), (b, c), (a, b, c)
- GROUP BY GROUPING SETS ((a), (c))
 - (a), (c)

Zapytania z podsumowaniami - przykład

```
SELECT salesPersonID, customerID, SUM(subTotal)
FROM Sales.SalesOrderHeader
GROUP BY CUBE(salesPersonID, customerID)
```

pracID	klientID	Suma
NULL	NULL	127337180.11
NULL	1	102351.80
NULL	2	29623.50
NULL	3	433942.38
NULL	29483	2049.10
268	NULL	1369624.65
268	7	3569.43
268	38	2785.51

Utwórz zestawienie przedstawiające **liczbę sprzedanych produktów** (SalesOrderDetail.OrderQty) **w poszczególnych kategoriach** (ProductCategory.Name) i **regionach** (SalesTerritory.CountryRegionCode) wraz z **podsumowaniami dla każdego regionu oraz kategorii**





Fundusze
Europejskie
Wiedza Edukacja Rozwój



Politechnika Wrocławska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Zadanie

Przygotuj wykaz nazw 20 najlepiej sprzedających się produktów wraz z ich kategoriami.

Przeanalizuj uzyskane wyniki.

Porównaj zużycie pamięci dla zapytań napisanych bez oraz z użyciem CTE.



Fundusze Europejskie
Wiedza Edukacja Rozwój



Politechnika Wrocławska

Unia Europejska
Europejski Fundusz Społeczny



„ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej”

Hurtownie danych

Dziękuję za uwagę

dr inż. Bernadetta Maleszka