

# Podzapytania

- Podzapytania (subqueries)
- Podzapytania do tabel
- Podzapytanie jako wyrażenie
- Podzapytania skorelowane
- operatory IN, NOT IN
- operatory EXISTS, NOT EXISTS

# Podzapytania do tabel

- W miejscu w którym możemy użyć nazwy tabeli, możemy użyć podzapytania

```
SELECT T.orderid, T.customerid  
FROM ( SELECT orderid, customerid  
        FROM orders ) AS T
```

# Podzapytanie jako wyrażenie

- Podzapytanie zwraca pojedynczą wartość
- Podzapytanie może być traktowane jako wyrażenie
  - może pojawić się na liście polecenia select
  - Może się pojawić w warunku

```
SELECT productname, price  
      , ( SELECT AVG(price) FROM products) AS average  
      , price - (SELECT AVG(price) FROM products) AS difference  
FROM products
```

# Podzapytanie w warunku

- Podzapytanie może być użyte w warunku

```
SELECT productname, price
      , ( SELECT AVG(unitprice) FROM products) AS average
      , unitprice - (SELECT AVG(price) FROM products) AS
difference
FROM products
WHERE unitprice > ( SELECT AVG(unitprice) FROM products)
```

# Podzapytania skorelowane

**1** Zewnętrzne zapytanie przekazuje dane do zapytania wewnętrznego

**2** Zapytanie wewnętrzne wykorzystuje te dane od wygenerowania wyniku

```
SELECT productname, unitprice
      , ( SELECT AVG(unitprice)
          FROM products as p_wew
        WHERE p_zew.categoryid = p_wew.categoryid ) AS
          average
FROM products as p_zew
```

**3** Zapytanie wewnętrzne zwraca ten wynik do zapytania zewnętrznego

**4** Proces jest powtarzany dla każdego wiersza zapytania wewnętrznego



**Back to Step 1**

# Podzapytania skorelowane w warunku

- Użycie podzapytania skorelowanego w warunku

```
SELECT productname, unitprice
      , ( SELECT AVG(unitprice) FROM products as p_wew
          WHERE p_zew.category_id = p_wew.categoryid ) AS
      average
FROM products as p_zewn
WHERE price >
      ( SELECT AVG(unitprice) FROM products as p_wew
          WHERE p_zew.category_id = p_wew.categoryid )
```

# Podzapytania skorelowane

- Dla każdego produktu podaj maksymalną liczbę zamówionych jednostek

```
SELECT DISTINCT productid, quantity
FROM [order details] AS ord1
WHERE quantity = ( SELECT MAX(quantity)
                   FROM [order details] AS ord2
                   WHERE ord1.productid =
ord2.productid )
```

# To samo przy użyciu GROUP BY

- Dla każdego produktu podaj maksymalną liczbę zamówionych jednostek

```
select productid, max(quantity)
from [order details]
group by productid
order by productid
```



# Operatory EXISTS, NOT EXISTS

- Zewnętrzne zapytanie testuje wystąpienie (lub nie) zbioru wynikowego określonego przez zapytanie wewnętrzne
  - zapytanie wewnętrzne zwraca TRUE lub FALSE

```
USE northwind
SELECT lastname, employeeid
FROM employees AS e
WHERE EXISTS (SELECT * FROM orders AS o
              WHERE e.employeeid = o.employeeid
              AND o.orderdate = '9/5/97')
GO
```

- Zapytanie zwraca listę wszystkich pracowników którzy obsłużyli zamówienie '9/5/97'

# EXISTS vs JOIN

- podzapytanie

```
SELECT lastname, employeeid
FROM employees AS e
WHERE EXISTS (SELECT * FROM orders AS o
              WHERE e.employeeid = o.employeeid
                 AND o.orderdate = '9/5/97')
```

- join

```
SELECT DISTINCT lastname, e.employeeid
FROM orders AS o
INNER JOIN employees AS e
  ON o.employeeid = e.employeeid
WHERE o.orderdate = '9/5/1997'
```

# Operatory EXISTS, NOT EXISTS

- Zewnętrzne zapytanie testuje wystąpienie (lub nie) zbioru wynikowego określonego przez zapytanie wewnętrzne
  - zapytanie wewnętrzne zwraca TRUE lub FALSE

```
SELECT lastname, employeeid
FROM employees AS e
WHERE NOT EXISTS (SELECT * FROM orders AS o
                  WHERE e.employeeid = o.employeeid
                  AND o.orderdate = '9/5/97')
```

- Zapytanie zwraca listę wszystkich pracowników którzy nie obsłużyli zamówień '9/5/97'

# Operatory IN, NOT IN

- **Zewnętrzne zapytanie testuje wystąpienie elementu w zbiorze (na liście) wygenerowanym przez zapytanie wewnętrzne**
  - zapytanie wewnętrzne zwraca zbiór elementów

```
USE northwind
SELECT lastname, employeeid
FROM employees AS e
WHERE employeeid IN (SELECT employeeid FROM orders AS o
                     WHERE o.orderdate = '9/5/97')
```

- **Zapytanie zwraca listę wszystkich pracowników którzy obsłużyli zamówienie '9/5/97'**

# JOIN vs EXISTS vs IN

- **JOIN**

```
SELECT DISTINCT lastname, e.employeeid
FROM orders AS o
INNER JOIN employees AS e
ON o.employeeid = e.employeeid
WHERE o.orderdate = '9/5/1997'
```

- **EXIST**

```
SELECT lastname, employeeid
FROM employees AS e
WHERE EXISTS (SELECT * FROM orders AS o
              WHERE e.employeeid = o.employeeid
              AND o.orderdate = '9/5/97')
```

- **IN**

```
SELECT lastname, employeeid
FROM employees AS e
WHERE employeeid in (SELECT employeeid FROM orders AS o
                    WHERE o.orderdate = '9/5/97')
```