

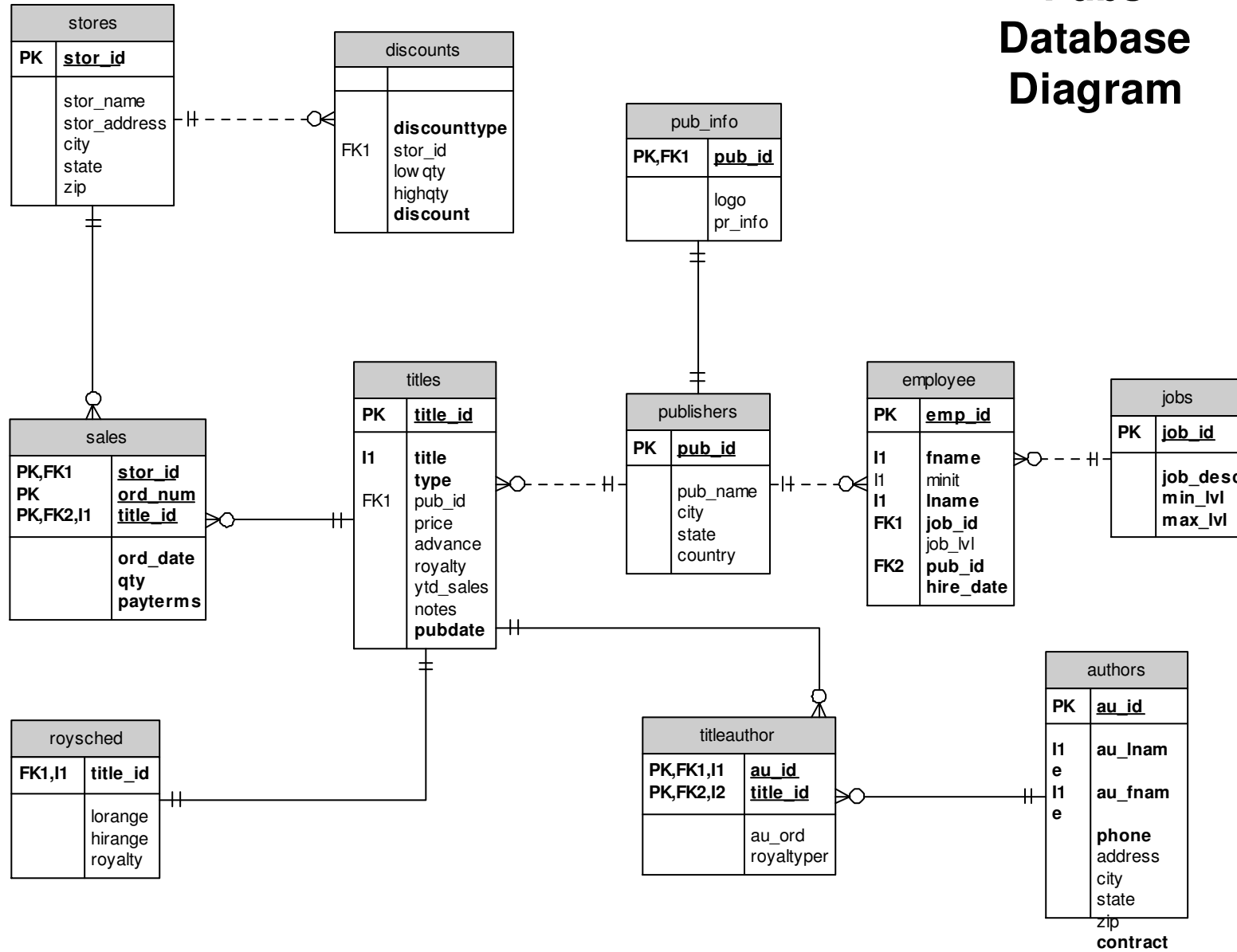


# SQL

## Podzapytania

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

# Pubs Database Diagram



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

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

# Podzapytanie jako wyrażenie



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

```
USE pubs
SELECT title, price
      , ( SELECT AVG(price) FROM titles) AS average
      , price - (SELECT AVG(price) FROM titles) AS
difference
FROM titles
WHERE type='popular_comp'
GO
```

- UWAGA: Podzapytanie jest wykonywane tylko raz

# Podzapytania skorelowane



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

```
USE northwind
SELECT orderid, customerid
FROM orders AS or1
WHERE 20 < (SELECT quantity
            FROM [order details] AS od
            WHERE or1.orderid = od.orderid
            AND od.productid = 23)

GO
```

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

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

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

**Back to Step 1**

- Dla każdego produktu podaj maksymalną liczbę zamówionych jednostek

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

# 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
```



- Podaj typy książek publikowane przez więcej niż jednego wydawcę

```
USE pubs
SELECT DISTINCT t1.type
FROM titles AS t1
WHERE t1.type IN
    (SELECT t2.type
     FROM titles AS t2
     WHERE t1.pub_id <> t2.pub_id)
```

# Podzapytania vs złączenia



- ```
USE pubs
SELECT DISTINCT t1.type
FROM titles AS t1
WHERE t1.type IN
    (SELECT t2.type
     FROM titles AS t2
     WHERE t1.pub_id <> t2.pub_id)
```

- złączenie

```
USE pubs
SELECT DISTINCT t1.type
FROM titles AS t1
INNER JOIN titles AS t2
    ON t1.type = t2.type
WHERE t1.pub_id <> t2.pub_id
```

# Przykład



- Podaj wszystkie tytuły których cena jest większa niż średnia cena książek tego samego typu

# Podzapytania vs HAVING



## ■ podzapytanie

```
USE pubs
SELECT t1.type, t1.title, t1.price
  FROM titles AS t1
 WHERE t1.price > ( SELECT AVG(t2.price) FROM titles AS t2
                   WHERE t1.type = t2.type )
GO
```

## ■ to samo przy użyciu grupowania i operatora HAVING

```
USE pubs
SELECT t1.type, t1.title, t1.price
  FROM titles AS t1
 INNER JOIN titles AS t2 ON t1.type = t2.type
 GROUP BY t1.type, t1.title, t1.price
  HAVING t1.price > AVG(t2.price)
GO
```

# Operator 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  
złożyli zamówienie '9/5/97'

# EXISTS vs JOIN



## ■ podzapytanie

```
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
```

## ■ join

```
USE northwind
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'

GO
```