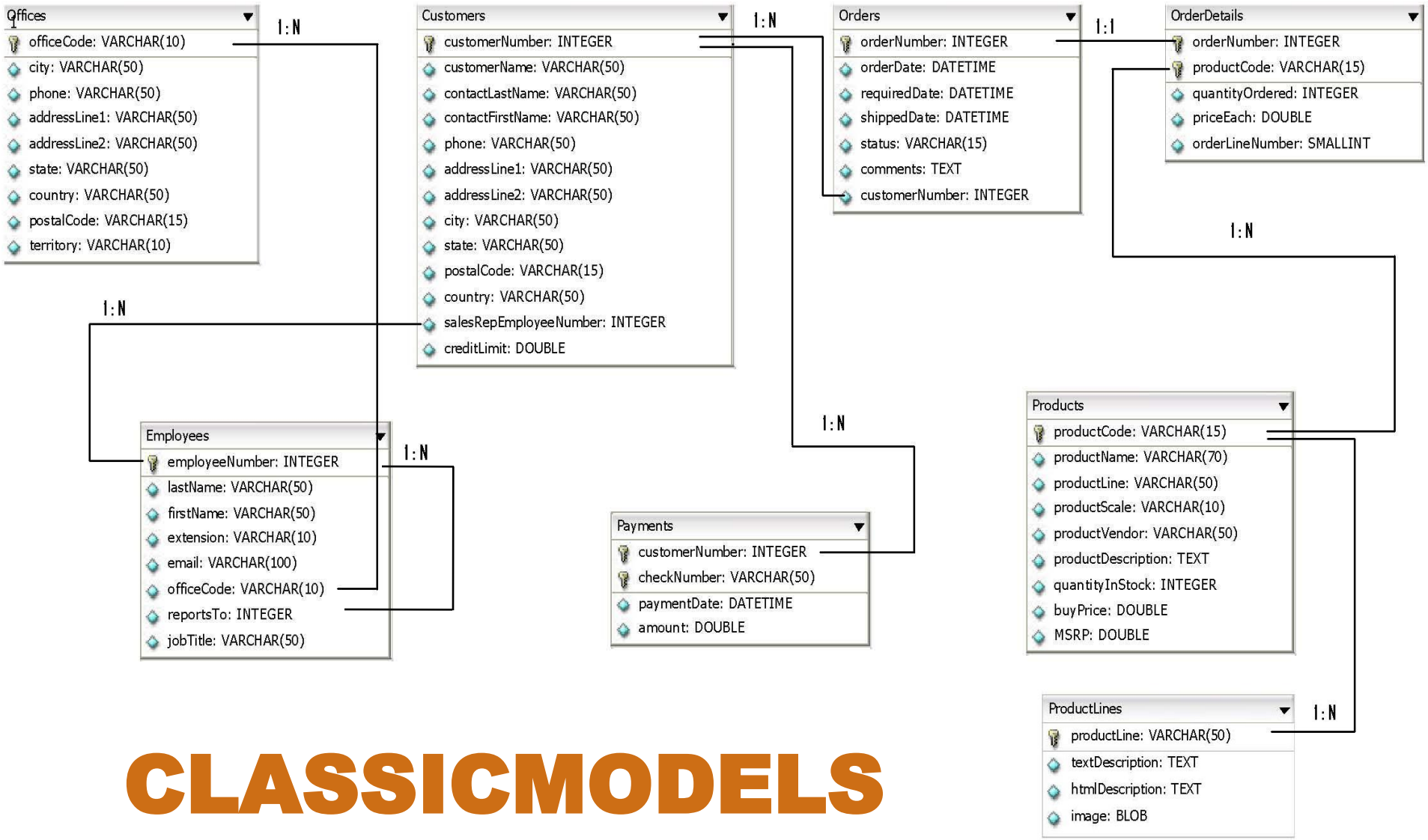
9

SUB QUERY

SKEMA RELASI YANG DIGUNAKAN

CLASSICMODELS



CLASSICMODELS

SUB QUERY

- Sub Query adalah suatu query yang menjadi bagian dari suatu query.
- Sub Query digunakan untuk menangani masalah yang kompleks yang mungkin sulit untuk dilakukan hanya dengan sebuah query.
- Menyediakan cara alternatif untuk melakukan operasi yang membutuhkan join atau union yang rumit.

CONTOH KASUS 1

- Cari data produk yang harga belinya (buyPrice) melebihi rata-rata harga belinya.
  - Query membutuhkan Sub Query karena untuk dapat mencari produk yang diinginkan, maka rata-rata harga belinya harus dicari terlebih dahulu.

CONTOH KASUS 1

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE buyPrice> (SELECT AVG(buyPrice) FROM Products)
```

ProductCode	ProductName	buyPrice
S10_1949	1952 Alpine Renault 1300	98.58
S10_2016	1996 Moto Guzzi 1100i	68.99
S10_4698	2003 Harley-Davidson Eagle Drag Bike	91.02
S10_4757	1972 Alfa Romeo GTA	85.68
S32_4485	1974 Ducati 350 Mk3 Desmo	56.13
S50_1392	Diamond T620 Semi-Skirted Tanker	68.29
S700_2466	America West Airlines B757-200	68.8
S700_2834	ATA: B757-300	59.33
S700_3167	F/A 18 Hornet 1/72	54.4

54 Rows

Sub query bekerja untuk mencari nilai rata-rata dari buyPrice yang akan menjadi patokan dalam seleksi data produk

CONTOH KASUS 2

- Carilah data produk yang harga belinya sama dengan harga beli termahal.
  - Query membutuhkan Sub Query karena untuk dapat mencari produk yang diinginkan, maka nilai terbesar harga belinya harus dicari terlebih dahulu.
  - Query ini bisa dilakukan dengan menggunakan ORDER BY dan LIMIT, tetapi hanya akan menghasilkan 1 baris saja. Bagaimana jika data yang sesuai dengan kriteria lebih dari 1 baris.

CONTOH KASUS 2

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE buyPrice=(SELECT MAX(buyPrice) FROM Products)
```

ProductCode	ProductName	BuyPrice
S10_4962	1962 LanciaA Delta 16V	103.42

```
SELECT ProductCode,ProductName,buyPrice FROM Products
ORDER BY buyPrice DESC LIMIT 1
```

- Perbedaan dari kedua SQL tersebut adalah :
  - Query pertama mungkin menampilkan data produk lebih dari 1 baris ketika baris yang buyprice-nya sama dengan nilai MAX(buyPrice) lebih dari 1 baris.
  - Qery kedua hanya akan menampilkan 1 baris saja karena ada penggunaan LIMIT. Kekurangan dari SQL ini adalah ketika ada data produk yang sama-sama memiliki nilai sama dengan MAX(buyprice) lebih dari 1 baris.

CONTOH KASUS 3

- Carilah customer yang pernah melakukan pembayaran.
  - Query tersebut membutuhkan sub query karena harus melakukan perbandingan data customer dengan data customer yang ada di tabel pembayaran (payments). Berarti data customer yang ada di tabel payments harus dicari terlebih dahulu.

CONTOH KASUS 3

```
SELECT CustomerNumber,CustomerName FROM Customers
WHERE CustomerNumber IN
        (SELECT DISTINCT CustomerNumber FROM Payments)
```

CustomerNumber	CustomerName
103	Atelier graphique
112	Signal Gift Stores
114	Australian Collectors, Co.
489	Double Decker Gift Stores, Ltd
495	Diecast Collectables
496	Kelly's Gift Shop

98 Rows

- Sub query bekerja untuk mencari data customer yang telah terdaftar di tabel payments (sudah pernah melakukan pembayaran)

CONTOH KASUS 4

- Carilah pegawai yang menjadi anak buah (reportTo ke) pegawai yang bernama “Mary Patterson”
  - Query tersebut harus membutuhkan sub query karena nomor pegawai (EmployeeNumber) dari Mary Patterson harus ditemukan terlebih dahulu, kemudian dibandingkan dengan data customer.

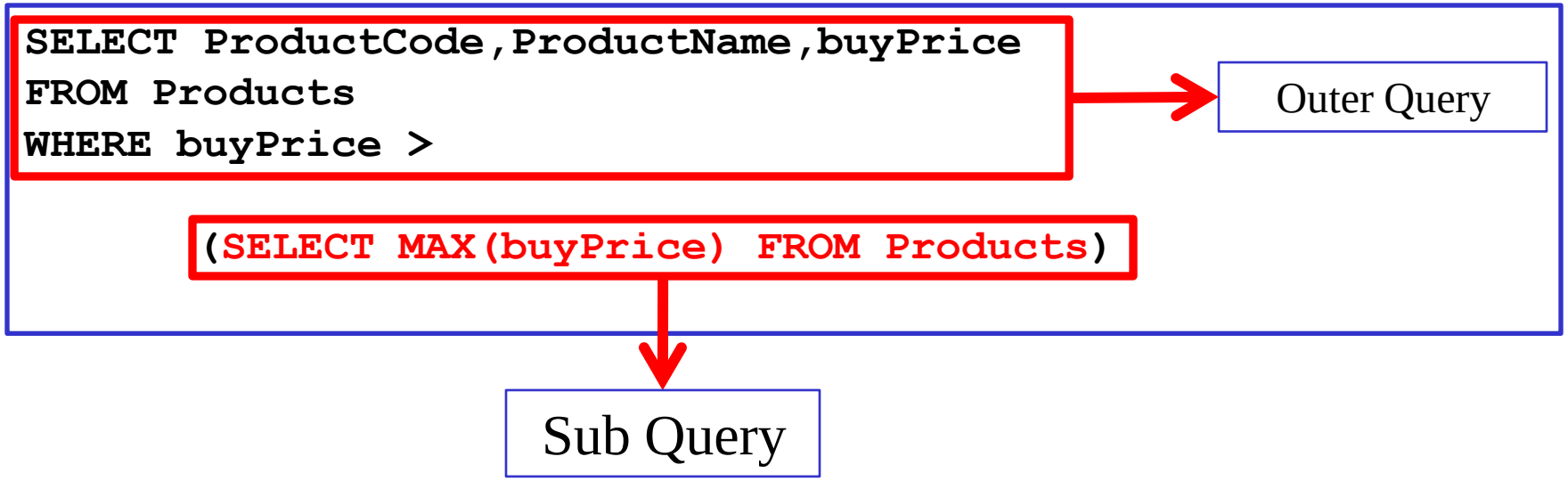
CONTOH KASUS 4

```
SELECT EmployeeNumber,CONCAT_WS(' ',FirstName,LastName)
FROM Employees
WHERE reportsTo=
      (SELECT EmployeeNumber FROM Employees
       WHERE CONCAT_WS(' ',FirstName,LastName) ='Mary Patterson'
      )
```

EmployeeNumber	CONCAT_WS(' ',FirstName,LastName)
1088	William Patterson
1102	Gerard Bondur
1143	Anthony Bow
1621	Mami Nishi

- Sub query bekerja untuk mencari EmployeeNumber pegawai yang bernama Mary Patterson.
- Query utamanya adalah mencari pegawai yang reportTo-nya ke EmployeeNumber milik Mary Patterson.

STRUKTUR SUBQUERY



ATURAN SUB QUERY

- Dalam sebuah query boleh memiliki lebih dari 1 sub query.
- Sebuah sub query boleh memiliki sub query lagi.
- Operator perbandingan yang dapat digunakan adalah =, >, <, >=, <=, <>, !=, <=>, IN, ANY, SOME, ALL, EXISTS, NOT EXISTS

CONTOH DATA

```
SELECT EmployeeNumber, FirstName
FROM Employees
```

EmployeeNumber	FirstName
1002	Diane
1056	Mary
1076	Jeff
1088	William
1102	Gerard
1143	Anthony
1165	Leslie
1166	Leslie
1188	Julie
1216	Steve
1286	Foon Yue
1323	George
1337	Loui
1370	Gerard
1401	Pamela
1501	Larry
1504	Barry
1611	Andy
1612	Peter
1619	Tom
1621	Mami
1625	Yoshimi
1702	Martin

```
SELECT EmployeeNumber
FROM Employees
WHERE OfficeCode=6
```

EmployeeNumber
1088
1611
1612
1619

OPERATOR =, >, <, >=, <=, <>, !=, <=> DALAM SUB QUERY

- Syarat dalam penggunaan operator tersebut adalah sub querynya hanya boleh memiliki 1 baris.
- Jika barisnya memiliki lebih dari 1 baris akan menampilkan pesan “Subquery returns more than 1 row”.

OPERATOR =, >, <, >=, <=, <>, !=, <=> DALAM SUB QUERY

- Contoh Benar

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE buyPrice > (SELECT AVG(buyPrice) FROM Products)
```

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE buyPrice > ( 54.3951818181818 )
```

- Untuk operator > boleh diganti dengan operator lain untuk mendapatkan hasil yang berbeda.

OPERATOR =, >, <, >=, <=, <>, !=, <=> DALAM SUB QUERY

- Contoh Salah

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE buyPrice >
        (SELECT BuyPrice FROM Products WHERE BuyPrice>100)
```

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE buyPrice > (
                    103.42
                    101.51
                  )
```

- SQL di atas akan menampilkan pesan kesalahan **"#1242 - Subquery returns more than 1 row**" karena mySQL tidak bisa melakukan perbandingan dengan data lebih dari 1 baris. Solusi untuk masalah ini adalah menggunakan ANY, SOME atau ALL.

OPERATOR ANY, SOME, ALL DALAM SUB QUERY

- Operator ANY, SOME dan ALL, harus diawali dengan penggunaan operator perbandingan =, >, <, >=, <=, <>, !=, <=>.
- Operator ANY akan memeriksa apakah suatu nilai dari outer query sesuai dengan SALAH SATU anggota dari hasil sebuah sub query. Kondisi sesuai ditentukan oleh operator yang ditulis sebelumnya.
- Operator SOME adalah alias dari ANY
- Operator ALL akan memeriksa apakah suatu nilai dari outer query sesuai dengan SEMUA anggota dari hasil sebuah sub query.
- Sub query boleh memiliki data lebih dari 1 baris.

OPERATOR ANY, SOME, ALL DALAM SUB QUERY

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE EmployeeNumber = ANY
      (SELECT EmployeeNumber FROM Employees WHERE OfficeCode=6)
```

EmployeeNumber	FirstName
1088	William
1611	Andy
1612	Peter
1619	Tom

OPERATOR ANY, SOME, ALL DALAM SUB QUERY

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE EmployeeNumber = ALL
      (SELECT EmployeeNumber FROM Employees WHERE OfficeCode=6)
```

- Menghasilkan 0 Rows, karena tidak ada EmployeeNumber yang sama dengan semua employeeNumber yang ada di sub query.

OPERATOR ANY, SOME, ALL DALAM SUB QUERY

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE EmployeeNumber > ANY
      (SELECT EmployeeNumber FROM Employees WHERE OfficeCode=6)
```

EmployeeNumber	FirstName
1102	Gerard
1143	Anthony
1165	Leslie
1166	Leslie
1188	Julie
1216	Steve
1286	Foon Yue
1323	George
1337	Loui
1370	Gerard
1401	Pamela
1501	Larry
1504	Barry
1611	Andy
1612	Peter
1619	Tom
1621	Mami
1625	Yoshimi
1702	Martin

OPERATOR ANY, SOME, ALL DALAM SUB QUERY

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE EmployeeNumber > ALL
      (SELECT EmployeeNumber FROM Employees WHERE OfficeCode=6)
```

EmployeeNumber	FirstName
1621	Mami
1625	Yoshimi
1702	Martin

OPERATOR IN DALAM SUB QUERY

- Operator IN akan memeriksa apakah suatu nilai di outer query ada dalam sebuah hasil sub query.
- Operator IN bisa disamakan dengan operator “= ANY”
- Lawan hasil dari operasi IN adalah NOT IN.
- Operator NOT IN bisa disamakan dengan “<> ALL”
- Sub query boleh memiliki data lebih dari 1 baris.

OPERATOR IN DALAM SUB QUERY

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE ProductCode IN
      (SELECT ProductCode FROM Products WHERE BuyPrice>100)
```

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE ProductCode IN ( S10_4962
                       S18_2238 )
```

ProductCode	ProductName	buyPrice
S10_4962	1962 LanciaA Delta 16V	103.42
S18_2238	1998 Chrysler Plymouth Prowler	101.51

OPERATOR IN DALAM SUB QUERY

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE ProductCode NOT IN
      (SELECT ProductCode FROM Products WHERE BuyPrice>100)
```

```
SELECT ProductCode,ProductName,buyPrice FROM Products
WHERE ProductCode NOT IN ( S10_4962
                           S18_2238 )
```

PERBANDINGAN DENGAN LEBIH DARI 1 KOLOM PADA SUB QUERY

- Perbandingan banyak kolom dimungkinkan dengan menuliskan field-field yang akan dicocokkan dalam sebuah tanda kurung ().
- Untuk lebih jelas, lihat contoh di slide berikutnya.

PERBANDINGAN DENGAN LEBIH DARI 1 KOLOM PADA SUB QUERY

```
SELECT OfficeCode,reportsTo,EmployeeNumber,FirstName
FROM Employees
WHERE (OfficeCode,reportsTo) =
      (SELECT OfficeCode,reportsTo
       FROM Employees WHERE employeeNumber=1370)
```

OfficeCode	reportsTo	EmployeeNumber	FirstName
4	1102	1337	Loui
4	1102	1370	Gerard
4	1102	1401	Pamela
4	1102	1702	Martin

- Mencari pegawai yang sekantor dan seatasan dengan pegawai yang bernomor 1370

CORRELATED SUBQUERY

- Sebuah *correlated subquery* adalah suatu subquery yang memiliki sebuah reference ke tabel yang juga menjadi outer query.
- Subquery boleh ditempatkan di daftar kolom atau dalam WHERE

CORRELATED SUBQUERY

```
SELECT e.EmployeeNumber,e.FirstName,
       (SELECT COUNT(*) FROM Employees e2
        WHERE e.EmployeeNumber=e2.reportsTo) BanyakBawahan
FROM Employees e
```

EmployeeNumber	FirstName	BanyakBawahan
1002	Diane	2
1056	Mary	4
1076	Jeff	0
1088	William	3
1102	Gerard	6
1143	Anthony	6
1165	Leslie	0
1166	Leslie	0
1188	Julie	0
1216	Steve	0
1286	Foon Yue	0
1323	George	0
1337	Loui	0
1370	Gerard	0
140[illegible]	Pamela	0
150[illegible]	Larry	0
1504	Barry	0
161[illegible]	Andy	0
1612	Peter	0
1619	Tom	0
162[illegible]	Mami	1
1625	Yoshimi	0
1702	Martin	0

CORRELATED SUBQUERY

```
SELECT e.EmployeeNumber,e.FirstName
FROM Employees e
WHERE (SELECT COUNT(*) FROM Employees e2
        WHERE e.EmployeeNumber=e2.reportsTo)>3
```

EmployeeNumber	FirstName
1056	Mary
1102	Gerard
1143	Anthony

EXISTS DAN NOT EXISTS

- EXISTS digunakan untuk memeriksa apakah subquery memiliki baris atau tidak. Jika minimal ada 1 baris (walaupun hanya berisi NULL), maka akan bernilai TRUE
- NOT EXISTS adalah kebalikan dari EXISTS.

EXISTS DAN NOT EXISTS

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE EXISTS(SELECT * FROM Employees
             WHERE OfficeCode=1)
```

Akan menampilkan semua data employee karena subquerynya bernilai TRUE.

EXISTS DAN NOT EXISTS

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE EXISTS(SELECT * FROM Employees
             WHERE OfficeCode=99)
```

Tidak akan menampilkan semua data employee karena subquerynya bernilai FALSE.

EXISTS DAN NOT EXISTS

```
SELECT EmployeeNumber, FirstName
FROM Employees
WHERE NOT EXISTS(SELECT * FROM Employees
                 WHERE OfficeCode=99)
```

Akan menampilkan semua data employee karena subquerynya bernilai TRUE.

[NOT] EXISTS & CORRELATED SUBQUERY

- Biasanya penggunaan [NOT] EXISTS selalu melibatkan correlated subquery.

```
SELECT e.EmployeeNumber,e.FirstName
FROM Employees e
WHERE EXISTS(SELECT * FROM Employees e2
        WHERE e.EmployeeNumber=e2.reportsTo)
```

EmployeeNumber	FirstName
1002	Diane
1056	Mary
1088	William
1102	Gerard
1143	Anthony
1621	Mami

- Query diatas akan menampilkan semua pegawai yang memiliki bawahan (subquerynya menghasilkan baris)