

# SQL Data Manipulation Language

- Εμφωλευμένες ερωτήσεις (Nested Queries)
  - Μια εντολή `select` μπορεί να περιέχει μια άλλη εντολή `select` αλλά υπό περιορισμούς
  - Μια εντολή `select` που εμφανίζεται μέσα σε μια άλλη εντολή `select` ονομάζεται `subselect`.
  - Ένα `subselect` μπορεί να εμφανίζεται στην πρόταση `where` με διάφορους τρόπους
  - Λογικά κατηγορήματα χρησιμοποιούνται για να γίνονται έλεγχοι στα `subselects`
- Απαίτηση:
  - μια σχέση στην πρόταση `from` δε μπορεί να είναι το αποτέλεσμα μιας εντολής `select`, πρέπει να είναι μια υπάρχουσα σχέση.

# SQL Data Manipulation Language

- Το κατηγορημα **IN** χρησιμοποιείται για να ελέγξει αν πλειάδες ανήκουν σε μια σχέση. Χρησιμοποιείται
  - με μια πρόταση **select**
  - ή ένα ρητά απαριθμημένο σύνολο

- Παραδείγματα:

«Βρείτε τα ids των πελατών που κάνουν παραγγελίες μέσω πρακτόρων με έδρα τη NY ή το Dallas»

SQL:

```
select cid from Orders
where aid in (
    select aid from agents
    where city='NY' or city='Dallas' );
```

Επιστρέφει τους πράκτορες των οποίων η έδρα είναι η NY ή το Dallas

- Επιστρέφει όλα τα αναγνωριστικά των πελατών από τις πλειάδες εκείνες των οποίων η τιμή aid ανήκει στα αποτελέσματα του εμφωλευμένου **select**.

# SQL Data Manipulation Language

- Το κατηγορημα **IN** χρησιμοποιείται για να ελέγξει αν πλειάδες ανήκουν σε μια σχέση
  - με μια πρόταση **select**
  - ρητά απαριθμημένο σύνολο

- Παραδείγματα:

1. «Βρείτε τους πράκτορες με έδρα το Chicago ή το Dallas».

SQL:

```
select * from Agents where city in ('Chicago', 'Dallas' );
```

2. «Βρείτε τα ονόματα και τις εκπτώσεις των πελατών που κάνουν παραγγελίες μέσω πρακτόρων με έδρα το Chicago ή το Dallas».

SQL:

```
select cname, discount from Customers
```

```
where cid in ( select cid from Orders
```

```
    where aid in (select aid from Agents
```

```
        where city in ('Chicago', 'Dallas' ) ) );
```

- Τα ονόματα των γνωρισμάτων σχετίζονται με τις σχέσεις του πλησιέστερου **select** .

# SQL Data Manipulation Language

- Παραδείγματα:

3. «Βρείτε τα ονόματα των πελατών που παραγγέλνουν το προϊόν p05».

SQL (1) :

```
select distinct cname from Customers, Orders  
where Customers.cid = Orders.cid and Orders.pid = 'p05'
```

SQL (2) :

```
select distinct cname from Customers  
where 'p05' in (select pid from Orders  
                  where cid = Customers.cid) ;
```

# SQL Data Manipulation Language

- **Κατηγορήματα Σύγκρισης με Ποσοδείκτες** (Quantified Comparison Predicates)
  - χρησιμοποιούνται για τη σύγκριση της τιμής μιας έκφρασης με το αποτέλεσμα μιας εντολής **select**
  - γενική μορφή:  $expr \vartheta \{any \mid all\} (subselect)$ ,  $\vartheta \in \{<, \leq, =, \neq, >, \geq\}$
  - συνολικά 12 κατηγορήματα μπορούν να οριστούν
  - Η έκφραση  $expr \vartheta any (subselect)$  είναι αληθής αν για τουλάχιστον ένα στοιχείο  $s$  στο αποτέλεσμα του **subselect** η έκφραση  $expr \vartheta s$  είναι αληθής
  - Η έκφραση  $expr \vartheta all (subselect)$  είναι αληθής αν για όλα τα στοιχεία  $s$  στο αποτέλεσμα του **subselect** η έκφραση  $expr \vartheta s$  είναι αληθής

# SQL Data Manipulation Language

- Παραδείγματα: Agents(aid, aname,city,percent)

1. Βρείτε τα ids των πρακτόρων με το μικρότερο ποσοστό προμήθειας.

```
select aid from agents  
where percent <= all  
    (select percent from agents);
```

2. Βρείτε τα ονόματα των πελατών που έχουν την ίδια έκπτωση με τους πελάτες στις πόλεις Dallas ή Boston. Customers(cid, cname,city,discnt)

```
select cname from customers  
where discnt = any  
    (select discnt from customers  
    where city = 'Dallas' or city = 'Boston');
```

# SQL Data Manipulation Language

Customers(cid, cname,city,discnt)

3. Βρείτε τα ids των πελατών που έχουν έκπτωση μικρότερη από αυτήν οποιουδήποτε πελάτη που ζει στο Dallas.

```
select cid from customers
where discnt < any
    (select discnt from customers
     where city = 'Dallas');
```

- Η έκφραση αυτή είναι λάθος! Ορθή έκφραση:

```
select cid from customers
where discnt < all
    (select discnt from customers
     where city = 'Dallas');
```

# SQL Data Manipulation Language

- Παρατηρήσεις:
  - Το κατηγορημα `expr = any (subselect)` έχει την ίδια σημασιολογία με την έκφραση `expr in (subselect)`
  - Η έκφραση `expr not in (subselect)` δεν είναι ισοδύναμη με το κατηγορημα `expr ≠ any (subselect)`
  - Είναι ισοδύναμη με το κατηγορημα `expr ≠ all (subselect)`
- Το κατηγορημα `exists`
  - Χρησιμοποιείται για να ελεγχθεί αν το αποτέλεσμα ενός subselect είναι κενό.
  - Γενική μορφή: `[not] exists (subselect)`
  - `exists (subselect)` είναι αληθής αν το αποτέλεσμα του subselect είναι μη-κενό σύνολο



# SQL Data Manipulation Language

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discnt)

- Παραδείγματα:

1. Βρείτε τα ονόματα των πελατών που κάνουν παραγγελίες μέσω του πράκτορα a05.

```
select cname from customers, orders  
where customers.cid=orders.cid and aid='a05' ;
```

ή

```
select cname from customers  
where exists (select * from orders  
where customers.cid=orders.cid and aid='a05' ) ;
```

Στο Σ.Λ. Πλειάδων:  $\{t^{(1)} \mid \exists u^{(4)} (\text{customers}(u) \wedge (u[2]=t[1]) \wedge \exists v^{(7)} (\text{orders}(v) \wedge (v[3]=u[1]) \wedge (v[4]='a05')))\}$

# SQL Data Manipulation Language

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discont)

2. Βρείτε τα ids των πελατών που παραγγέλνουν τα προϊόντα p01 και p07.

```
select cid from orders x
where x.pid = 'p01' and exists
(select * from orders where
  cid=x.cid and pid = 'p07');
```

3. Βρείτε τα ονόματα των πελατών που δεν κάνουν καμία παραγγελία μέσω του πράκτορα a05.

```
select cname from customers
where not exists (select * from orders where
orders.cid = customers.cid and aid = 'a05');
```

# SQL Data Manipulation Language

- Ποιο είναι το αποτέλεσμα της εντολής

```
select cname from customers, orders where  
not (orders.cid = customers.cid and aid='a05');
```

σε σχέση με την προηγούμενη ερώτηση;

- επιστρέφει τους πελάτες ο οποίοι δίνουν παραγγελίες με πράκτορες διαφορετικούς από τον a05 αλλά οι οποίοι ενδέχεται να δίνουν παραγγελίες και μέσω του a05.
- Η ερώτηση μπορεί να εκφραστεί και ως:

```
select cname from customers where cid not in  
(select cid from orders where aid='a05'); ή
```

```
select cname from customers where cid ≠ all  
(select cid from orders where aid = 'a05');
```

# SQL Data Manipulation Language

- 4.** Το κατηγορημα **not exists** χρησιμοποιείται για την έκφραση του τελεστή της διαφοράς :
- αν  $R$  και  $S$  είναι συμβατές σχέσεις με σχήμα  $\{A_1, \dots, A_n\}$ , η διαφορά  $R - S$  μπορεί να εκφραστεί ως:

```
select A1, ..., An from R where not exists  
(select * from S where S.A1 = R.A1 and S.A2 =  
R.A2 and ... and S.An = R.An) ;
```

# SQL Data Manipulation Language

Orders(orderno, month,cid,aid,pid,qty,amt)

Customers(cid, cname,city,discont)

5. Βρείτε τα ονόματα των πόλεων στις οποίες μένουν πελάτες που παραγγέλνουν το προϊόν p01.

```
select distinct city from customers where cid  
in (select cid from orders where pid='p01');
```

```
select distinct city from customers where  
cid = any (select cid from orders where  
pid='p01');
```

```
select distinct city from customers c where  
exists (select * from orders where cid=c.cid  
and pid='p01');
```

# SQL Data Manipulation Language

Orders(orderno, month,cid,aid,pid,qty,amt)

Customers(cid, cname,city,discnt)

```
select distinct city from customers, orders
where orders.cid=customers.cid and
orders.pid = 'p01' ;
```

```
select distinct city from customers where
'p01' in (select pid from orders where cid =
customers. cid) ;
```

- Όλες αυτές οι μορφές είναι ισοδύναμες.
- Η γλώσσα θα είχε την ίδια εκφραστική δύναμη αν το κατηγορήμα **in** και τα κατηγορήματα σύγκρισης με ποσοδείκτες παραλείπονταν.
- Δεν μπορεί όμως να παραληφθεί το κατηγορήμα **exists**.

# SQL Data Manipulation Language

- Τελεστής **union**
  - συνδυάζει **subselects** τα οποία παράγουν συμβατές σχέσεις
  - γενική μορφή: *subselect {union [all] subselect}*
  - περιορισμός: τα subselects δεν μπορούν να περιέχουν τον τελεστή **union**
  - **Παράδειγμα:** Βρείτε τις πόλεις στις οποίες διαμένουν είτε πελάτες είτε πράκτορες

```
select city from customers union  
select city from agents;
```

ή για να επιτρέψουμε επαναλαμβανόμενες πλειάδες

```
select city from customers union all  
select city from agents;
```

# SQL: Διαίρεση

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discont)

- Διαίρεση: η SQL δεν παρέχει τελεστή για διαίρεση
- Παράδειγμα: Βρείτε τα ids των πελατών που **κάνουν παραγγελίες μέσω όλων των πρακτόρων της Νέας Υόρκης**.
  - Ή ισοδύναμα: Βρείτε τα ids των πελατών, έτσι ώστε **να μην υπάρχει πράκτορας στη Νέα Υόρκη** που δεν κάνει παραγγελίες για κάθε έναν από αυτούς τους πελάτες.

```
select cid from customers where not exists
(select * from agents where city="New York"
and not exists
(select * from orders where
orders.cid=customers.cid and
orders.aid=agents.aid) );
```



# SQL: Διαίρεση

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discont)

- **Παράδειγμα:** Βρείτε τα ids των πρακτόρων στη Νέα Υόρκη ή το Dallas που **κάνουν παραγγελίες για όλα τα προϊόντα που κοστίζουν πάνω από 1\$**.
- **Ή ισοδύναμα:** Βρείτε τα ids των πρακτόρων στη Νέα Υόρκη ή το Dallas, έτσι ώστε **να μην υπάρχει προϊόν που να κοστίζει πάνω από 1\$ και να μην το παραγγέλνουν.**

```
select aid from agents where (city="New York"
or city = "Dallas") and not exists
(select pid from products where price > 1.00
and not exists
(select * from orders where
orders.pid=products.pid and
orders.aid=agents.aid) );
```

# SQL: Διαίρεση

- Παράδειγμα: Βρείτε τα ids των προϊόντων που παραγγέλνονται από όλους τους πελάτες στο Dallas.
  - Ή ισοδύναμα: Βρείτε τα ids των προϊόντων που είναι τέτοια ώστε να μην υπάρχει πελάτης στο Dallas που να μην τα παραγγέλνει.  

```
select pid from products where not exists  
(select cid from customers where city="Dallas"  
and not exists  
  
  (select * from orders where  
   orders.pid=products.pid and  
   orders.cid=customers.cid) ) ;
```
- Η έκφραση της διαίρεσης με αυτό τον τρόπο βασίζεται στην ισοδυναμία:

$$\forall z \exists y p(z,y) \equiv \neg \exists z \neg \exists y p(z,y)$$