

Σχεσιακός Λογισμός Πλειάδων (Tuple Relational Calculus)

- Έχει την ίδια εκφραστική δύναμη με το **Σχεσιακό Λογισμό Πεδίων**
 - Οι προτάσεις του TRC και πάλι δηλώνουν σχέσεις (πιθανά μη-πεπερασμένες).
 - Κάθε πρόταση χρησιμοποιεί ένα σύνολο μεταβλητών, οι οποίες διακρίνονται σε **ελεύθερες (free)** και **δεσμευμένες (bound)**. Η ίδια μεταβλητή μπορεί να εμφανίζεται ως ελεύθερη και δεσμευμένη στην ίδια πρόταση.
- Σε αντίθεση με τον DRC, οι μεταβλητές αντιστοιχούν σε ολόκληρες **πλειάδες** μιας σχέσης και όχι σε μεμονωμένα γνωρίσματα.
 - Το σχεσιακό σχήμα για μια πρόταση είναι ένα σύνολο γνωρισμάτων τα οποία αντιστοιχούν στην ελεύθερη μεταβλητή της πρότασης.

Καλά-Δομημένες Προτάσεις TRC (Well-Formed TRC Formulae)

- Ακολουθούν τον παρακάτω αναδρομικό ορισμό:
- Οι **ατομικές προτάσεις** είναι καλά-δομημένες προτάσεις
 1. Για κάθε σύμβολο κατηγορήματος **p** και μεταβλητή **μ**, η **p(μ)** είναι ατομική πρόταση.
 2. Αν **X, Y** είναι μεμονωμένα γνώρισματα πλειάδων ή σταθερές και $\theta \in \{<, >, =, \leq, \geq, \neq\}$, τότε η **X θ Y** είναι ατομική πρόταση.

Σημείωση: ένα σύμβολο κατηγορήματος **p** δηλώνει μια σχέση, με την ερμηνεία ότι **p(μ)** είναι αληθές αν **μ** ανήκει στη σχέση που αντιστοιχεί στο **p**

Σημείωση: για να δηλώσουμε ένα μεμονωμένο γνώρισμα μεταβλητής πλειάδας **μ**, χρησιμοποιούμε το συμβολισμό **μ[i]**, όπου **i** ο αριθμός θέσης του γνωρίσματος στην πλειάδα.
- Αν οι **F₁, F₂** είναι καλά-δομημένες προτάσεις, τότε και οι **F₁ ∧ F₂**, **F₁ ∨ F₂** και **¬F₁** είναι καλά-δομημένες προτάσεις.
- Αν η μεταβλητή **μ** είναι ελεύθερη στην πρόταση **F**, τότε οι **(∃μ)F** **(∀μ)F** είναι καλά-δομημένες προτάσεις.

Σχεσιακός Λογισμός Πλειάδων (Tuple Relational Calculus)

- Κάθε πρόταση του TRC με μία ή περισσότερες ελεύθερες μεταβλητές ορίζει μια σχέση της οποίας οι πλειάδες αντιστοιχούν στις ελεύθερες μεταβλητές.
- Μια επερώτηση έχει τη μορφή $\{\mu | F(\mu)\}$, όπου η F είναι μια καλά-δομημένη πρόταση του TRC με ελεύθερη μεταβλητή μ .
- Η έκφραση $\{\mu | F(\mu)\}$ αναπαριστά το σύνολο των πλειάδων οι οποίες κάνουν αληθή την F , αν αντικατασταθούν από τη μεταβλητή μ .
- Αν ο βαθμός (arity) μιας μεταβλητής πλειάδας δεν προκύπτει ξεκάθαρα, επισημαίνεται με κατάλληλο εκθέτη στη μεταβλητή: $\{\mu^{(i)} | F(\mu)\}$, για πλειάδα με βαθμό i .
- Ένα υποσύνολο των προτάσεων του TRC, οι **ασφαλείς** προτάσεις, ορίζουν πεπερασμένες σχέσεις.

Παραδείγματα Σχεσιακού Λογισμού Πλειάδων

PRODUCTS

pid	pname	city	qty	price
-----	-------	------	-----	-------

CUSTOMERS

cid	cname	city	disct
-----	-------	------	-------

AGENTS

aid	aname	city	percent
-----	-------	------	---------

ORDERS

ordno	month	cid	aid	pid	qty	dollars
-------	-------	-----	-----	-----	-----	---------

- Βρείτε τα ids των πελατών που ζουν στη Νέα Υόρκη.

$$\{\mu^{(1)} \mid (\exists \rho^{(4)}) customers(\rho) \wedge (\rho[3] = 'New York') \wedge (\mu[1] = \rho[1])\}$$

- Βρείτε τα ονόματα και τις τιμές των προϊόντων που παραγγέλνει ο πελάτης c002 μέσω του πράκτορα a01.

$$\left\{ \mu^{(2)} \mid \begin{array}{l} (\exists \rho^{(5)}) (\exists \tau^{(7)}) products(\rho) \wedge orders(\tau) \wedge \\ (\tau[3] = 'c002') \wedge (\tau[4] = 'a01') \wedge \\ (\tau[5] = \rho[1]) \wedge (\mu[1] = \rho[2]) \wedge (\mu[2] = \rho[5]) \end{array} \right\}$$

Παραδείγματα Σχεσιακού Λογισμού Πλειάδων

PRODUCTS

pid	pname	city	qty	price
-----	-------	------	-----	-------

CUSTOMERS

cid	cname	city	discnt
-----	-------	------	--------

AGENTS

aid	aname	city	percent
-----	-------	------	---------

ORDERS

ordno	month	cid	aid	pid	qty	dollars
-------	-------	-----	-----	-----	-----	---------

- Βρείτε ζεύγη πρακτόρων που ζουν στην ίδια πόλη.

$$\left\{ \mu^{(2)} \left| \begin{array}{l} (\exists \rho^{(4)}) (\exists \tau^{(4)}) \text{agents}(\rho) \wedge \text{agents}(\tau) \wedge (\rho[1] \neq \tau[1]) \wedge \\ (\mu[1] = \rho[1]) \wedge (\mu[2] = \rho[2]) \wedge (\rho[3] = \tau[3]) \end{array} \right. \right\}$$

- Βρείτε τα ονόματα όλων των πρακτόρων που κάνουν παραγγελίες για όλα τα προϊόντα που παραγγέλλνει ο πελάτης c002.

$$\left\{ \mu^{(1)} \left| \begin{array}{l} (\exists \rho^{(4)}) \text{agents}(\rho) \wedge (\mu[1] = \rho[2]) \wedge \\ (\forall \tau^{(7)}) (\text{orders}(\tau) \wedge (\tau[3] = 'c002')) \Rightarrow \\ (\exists \sigma^{(7)}) \text{orders}(\sigma) \wedge (\sigma[4] = \rho[1]) \wedge (\sigma[5] = \tau[5]) \end{array} \right. \right\}$$

Ασφαλείς Προτάσεις TRC (Safe TRC Formulae)

- Ασφαλείς προτάσεις ορίζουν πεπερασμένες σχέσεις.
- Ορισμός: Μια πρόταση F του TRC είναι ασφαλής αν ικανοποιούνται οι παρακάτω ιδιότητες:
 1. Δε χρησιμοποιείται ο καθολικός ποσοδείκτης (\forall).
 2. Αν $F = F_1 \vee F_2$ τότε οι F_1 και F_2 έχουν την ίδια ελεύθερη μεταβλητή

Ασφαλείς Προτάσεις TRC (Safe TRC Formulae)

- Ορισμός: Μια πρόταση F του TRC είναι ασφαλής αν ικανοποιούνται οι παρακάτω ιδιότητες:
 3. Για οποιαδήποτε μέγιστη υποπρόταση $F_1 \wedge \dots \wedge F_m$ της F όλες οι ελεύθερες μεταβλητές των F_i πρέπει να είναι **περιορισμένες** με την εξής έννοια:
 - a) Μια μεταβλητή είναι περιορισμένη αν είναι ελεύθερη σε μια υποπρόταση F_i , όπου η F_i δεν είναι αριθμητική σύγκριση και δεν προηγείται \neg .
 - b) Αν F_i είναι $\mu[j] = a$ όπου a είναι σταθερά, τότε η $\mu[j]$ είναι περιορισμένη.
 - c) Αν F_i είναι $\mu[j] = \rho[k]$ και η $\rho[k]$ είναι περιορισμένη, τότε η $\mu[j]$ είναι περιορισμένη.
 - Σημείωση: Η $\mu^{(i)}$ είναι περιορισμένη όταν για κάθε $j = 1 \dots i$ η $\mu[j]$ είναι περιορισμένη.
 4. Αν η F είναι της μορφής $H_1 \wedge \dots \wedge H_j \wedge \neg G \wedge I_1 \wedge \dots \wedge I_j$ και οι συζεύξεις ικανοποιούν τον παραπάνω κανόνα, τότε είναι ασφαλής αν τουλάχιστον ένα από τα H ή I δεν έχει άρνηση.

Από τη Σ.Α. στο Σχεσιακό Λογισμό Πλειάδων

- Θεώρημα 1: Κάθε έκφραση της Σχεσιακής Άλγεβρας μπορεί να εκφραστεί ως ασφαλής πρόταση του Σχεσιακού Λογισμού Πλειάδων.
- Απόδειξη: Θα δείξουμε, με επαγωγή στον αριθμό των τελεστών της αλγεβρικής έκφρασης, ότι για κάθε έκφραση E της Σ.Α. η οποία ορίζει μια σχέση βαθμού k , υπάρχει μια ασφαλής πρόταση $F(\mu)$ του TRC η οποία ορίζει την ίδια σχέση.

Βάση επαγωγής: $E = R$, για κάποια σχέση R . Η αντίστοιχη πρόταση του TRC είναι $r(\mu)$.

Υπόθεση επαγωγής: έστω ότι για τις εκφράσεις E_1, E_2 της Σ.Α., οι οποίες ορίζουν σχέσεις βαθμού k , υπάρχουν προτάσεις F_1, F_2 του TRC οι οποίες ορίζουν τις ίδιες σχέσεις.

Από τη Σ.Α. στο Σχεσιακό Λογισμό Πλειάδων

Επαγωγικό βήμα: Θα δείξουμε ότι υπάρχουν προτάσεις του TRC οι οποίες ορίζουν τις ίδιες σχέσεις με τις εκφράσεις $E_1 \cup E_2$, $E_1 - E_2$, $\pi_X(E_1)$, $E_1 \times E_2$ και $\sigma_F(E_1)$.

1. $E = E_1 \cup E_2$ και έστω k ο βαθμός των σχέσεων των E, E_1, E_2 . Από την επαγωγική υπόθεση έχουμε ότι υπάρχουν προτάσεις F_1, F_2 οι οποίες ορίζουν τις ίδιες σχέσεις με τις E_1, E_2 . Υποθέτουμε επίσης ότι οι F_1, F_2 έχουν την ίδια ελεύθερη μεταβλητή μ (αν όχι, μπορούμε να μετονομάσουμε τις μεταβλητές). Τότε, η πρόταση που αντιστοιχεί στην έκφραση E είναι η πρόταση $F_1 \vee F_2$.
2. $E = E_1 - E_2$ και έστω ότι οι προτάσεις $F_1(\mu)$ και $F_2(\mu)$ αντιστοιχούν στις εκφράσεις E_1 και E_2 αντίστοιχα. Τότε, η πρόταση που αντιστοιχεί στην έκφραση E είναι η $F_1 \wedge \neg F_2$.

Από τη Σ.Α. στο Σχεσιακό Λογισμό Πλειάδων

3. $E = \pi_{A_{i_1}, \dots, A_{i_n}}(E_1)$. Έστω $\{A_1, \dots, A_k\}$ τα γνωρίσματα της σχέσης της E_1 . Αν $F_1(\rho)$ είναι η πρόταση που αντιστοιχεί στην έκφραση E_1 , τότε η πρόταση για την έκφραση E είναι $F(\mu^{(n)}) = (\exists \rho^{(k)}) F_1(\rho) \wedge (\mu[1] = \rho[i_1]) \wedge \dots \wedge (\mu[n] = \rho[i_n])$.
4. $E = E_1 \times E_2$ και έστω ότι οι προτάσεις $F_1(\rho^{(n)})$ και $F_2(\sigma^{(m)})$ αντιστοιχούν στις εκφράσεις E_1 και E_2 αντίστοιχα. Τότε, η πρόταση που αντιστοιχεί στην έκφραση E είναι η $F(\mu^{(m+n)}) = (\exists \rho^{(n)}) (\exists \sigma^{(m)}) F_1(\rho) \wedge F_2(\sigma) \wedge (\mu[1] = \rho[1]) \wedge \dots \wedge (\mu[n] = \rho[n]) \wedge (\mu[n+1] = \sigma[1]) \wedge \dots \wedge (\mu[n+m] = \sigma[m])$.
5. $E = \sigma_{A_i \theta A_j}(E_1)$ και έστω ότι η πρόταση $F_1(\rho)$ αντιστοιχεί στην E_1 . Τότε, η πρόταση που αντιστοιχεί στην έκφραση E είναι η $F(\mu) = F_1(\mu) \wedge (\mu[i] \theta \mu[j])$.

Από τη Σ.Α. στο Σχεσιακό Λογισμό Πλειάδων

- Παράδειγμα:

$$E = \pi_Y(R(X, Y))$$

Η πρόταση του Σχεσιακού Λογισμού Πλειάδων για την E είναι:

$$\{\lambda^{(1)} \mid (\exists \tau^{(2)})(r(\tau) \wedge (\lambda[1] = \tau[2]))\}$$

Από τη Σ.Α. στο Σχεσιακό Λογισμό Πλειάδων

- Παράδειγμα:

$$E = S(X) \times \pi_Y(R(X, Y))$$

Η πρόταση του Σχεσιακού Λογισμού Πλειάδων για την E είναι:

$$\left\{ \lambda^{(2)} \left| (\exists \mu^{(1)}) (\exists \rho^{(1)}) \left(\begin{array}{c} s(\mu) \wedge \\ (\exists \tau^{(2)}) (r(\tau) \wedge (\rho[1] = \tau[2])) \wedge \\ (\lambda[1] = \mu[1]) \wedge (\lambda[2] = \rho[1]) \end{array} \right) \right. \right\}$$

Από τη Σ.Α. στο Σχεσιακό Λογισμό Πλειάδων

- Παράδειγμα:

$$E = R(X, Y) - (S(X) \times \pi_Y(R(X, Y)))$$

Η πρόταση του Σχεσιακού Λογισμού Πλειάδων για την E είναι:

$$\left\{ \lambda^{(2)} \left| r(\lambda) \wedge \neg \left((\exists \mu^{(1)}) (\exists \rho^{(1)}) \left(\begin{array}{c} s(\mu) \wedge \\ (\exists \tau^{(2)}) (r(\tau) \wedge (\rho[1] = \tau[2])) \wedge \\ (\lambda[1] = \mu[1]) \wedge (\lambda[2] = \rho[1]) \end{array} \right) \right) \right. \right. \right\}$$

Από το Σ.Λ. Πλειάδων (TRC) στο Σ.Λ. Πεδίων (DRC)

- Θεώρημα 2: Για κάθε ασφαλή πρόταση του TRC υπάρχει μια ασφαλής πρόταση του DRC που ορίζει την ίδια σχέση.
- Απόδειξη: Η ιδέα είναι να αντικαταστήσουμε κάθε μεταβλητή πλειάδας $\mu^{(k)}$ με μεταβλητές πεδίου X_1, \dots, X_k , χρησιμοποιώντας τη X_j στη θέση της $\mu[j]$ παντού στην πρόταση.
- Θεώρημα 3: Η Σχεσιακή Άλγεβρα, ο ασφαλής Σχεσιακός Λογισμός Πεδίων και ο ασφαλής Σχεσιακός Λογισμός Πλειάδων έχουν την ίδια εκφραστική δύναμη.

Παραδείγματα Σ.Α., Σ.Λ. Πεδίων και Σ.Λ. Πλειάδων

LIVES

person-name	street	city
-------------	--------	------

WORKS

person-name	company-name	salary
-------------	--------------	--------

LOCATED-IN

company-name	city
--------------	------

MANAGES

person-name	manager-name
-------------	--------------

1. Βρείτε τα ονόματα των ατόμων που εργάζονται στη “First Bank”.

$$\pi_{person-name}(\sigma_{company-name="First Bank"}(WORKS))$$

$$\{N | (\exists C)(\exists S) works(N, C, S) \wedge (C = "First Bank")\}$$

$$\{\mu^{(1)} | (\exists \rho^{(3)}) works(\rho) \wedge (\rho[1] = \mu[1]) \wedge (\rho[2] = "First Bank")\}$$

Παραδείγματα Σ.Α., Σ.Λ. Πεδίων και Σ.Λ. Πλειάδων

LIVES

person-name	street	city
-------------	--------	------

WORKS

person-name	company-name	salary
-------------	--------------	--------

LOCATED-IN

company-name	city
--------------	------

MANAGES

person-name	manager-name
-------------	--------------

2. Βρείτε τα ονόματα και τις πόλεις των ατόμων που εργάζονται στη “First Bank”.

$$\pi_{person-name, city}(\sigma_{company-name="First Bank"}(LIVES \bowtie WORKS))$$

$$\{N, C | (\exists T)(\exists S) \text{ lives}(N, T, C) \wedge \text{works}(N, "First Bank", S)\}$$

$$\left\{ \mu^{(2)} \left| \begin{array}{l} (\exists \rho^{(3)})(\exists \tau^{(3)}) \text{ lives}(\rho) \wedge \text{works}(\tau) \wedge \\ (\mu[1] = \rho[1]) \wedge (\mu[2] = \rho[3]) \\ \wedge (\rho[1] = \tau[1]) \wedge (\tau[2] = "First Bank") \end{array} \right. \right\}$$

Παραδείγματα Σ.Α., Σ.Λ. Πεδίων και Σ.Λ. Πλειάδων

LIVES

person-name	street	city
-------------	--------	------

WORKS

person-name	company-name	salary
-------------	--------------	--------

LOCATED-IN

company-name	city
--------------	------

MANAGES

person-name	manager-name
-------------	--------------

3. Βρείτε τα ονόματα των ατόμων που **δεν** εργάζονται στη “First Bank”.

$$\pi_{person-name}(\sigma_{company-name \neq "First Bank"}(WORKS))$$

$$\{N | (\exists C)(\exists S) works(N, C, S) \wedge (C \neq "First Bank")\}$$

$$\{\mu^{(1)} | (\exists \rho^{(3)}) works(\rho) \wedge (\rho[1] = \mu[1]) \wedge (\rho[2] \neq "First Bank")\}$$

Παραδείγματα Σ.Α., Σ.Λ. Πεδίων και Σ.Λ. Πλειάδων

LIVES

person-name	street	city
-------------	--------	------

WORKS

person-name	company-name	salary
-------------	--------------	--------

LOCATED-IN

company-name	city
--------------	------

MANAGES

person-name	manager-name
-------------	--------------

4. Βρείτε τα ονόματα των ατόμων που **δεν** εργάζονται στη “First Bank” (συμπεριλαμβανομένων και των ατόμων που δεν εργάζονται πουθενά)
 $\pi_{person-name}(LIVES) - \pi_{person-name}(\sigma_{company-name="First Bank"}(WORKS))$

$$\left\{ N \mid \left((\exists T)(\exists C) \left(lives(N, T, C) \wedge \neg((\exists C)(\exists S) works(N, C, S)) \right) \wedge (C = "First Bank") \right) \right\}$$

$$\left\{ \mu^{(1)} \mid \left((\exists \rho^{(3)}) lives(\rho) \wedge (\rho[1] = \mu[1]) \wedge \neg(\exists \tau)(works(\tau) \wedge (\tau[1] = \rho[1])) \wedge (\tau[2] = "First Bank") \right) \right\}$$

SQL – Structured Query Language

- Η γλώσσα που χρησιμοποιείται συχνότερα από εμπορικά Συστήματα Διαχείρισης Βάσεων Δεδομένων.
- Παρέχει μια **Γλώσσα Ορισμού Δεδομένων (Data Definition Language – DDL)** και μια **Γλώσσα Χειρισμού Δεδομένων (Data Manipulation Language – DML)**.
- Βασίζεται στο Σχεσιακό Λογισμό και έχει την ίδια εκφραστική δύναμη με τη Σχεσιακή Άλγεβρα και τον ασφαλή Σχεσιακό Λογισμό.
- Επιτρέπει στους χρήστες να θέτουν ad hoc διαδραστικές **επερωτήσεις**
- Εντολές SQL μπορούν να εκτελεστούν ενσωματωμένες σε μια **γλώσσα προγραμματισμού** (C, Pascal, Java, 4GL)
- Εκδόσεις της SQL έχουν γίνει **διεθνή πρότυπα**: το πρότυπο SQL:2011 είναι η πιο πρόσφατη έκδοση (Δεκέμβριος 2011), με την προηγούμενη έκδοση SQL:2008 (Ιούλιος 2008) να είναι η πιο πρόσφατη έκδοση υλοποιημένη σε εμπορικά συστήματα

SQL – Data Definition Language

- Εντολή **create table**:

```
create table tablename (attrname datatype [not null]{, attrname datatype [not null]};
```

- Δημιουργεί μια σχέση με όνομα *tablename* και γνωρίσματα *attrname* των καθορισμένων τύπων *datatype*. Το προαιρετικό **not null** υποδηλώνει ότι δεν επιτρέπονται κενές τιμές στο συγκεκριμένο γνώρισμα.

- Παραδείγματα:

```
■ create table customers (cid char(4) not null,  
  cname varchar(13), city varchar(20), discnt real  
■ Create table products (pid char(3) not null,  
  pname varchar(15), city varchar(20),  
  quantity int, price real);
```

SQL – Data Definition Language

- Εντολή **drop table**:
drop table *tablename* ;
- Διαγράφει τη σχέση με όνομα *tablename*.
- Εντολή **alter table**:
alter table *tablename* *action* ;
- Αλλάζει τη σχέση *tablename* εκτελώντας την ενέργεια *action* που μπορεί να είναι για παράδειγμα:
 - **add** *attrname* *datatype* (προσθήκη γνωρίσματος)
 - **drop** *attrname* (διαγραφή γνωρίσματος)

SQL – Data Manipulation Language

- Η εντολή **select** χρησιμοποιείται για να εκφράσουμε επερωτήσεις:
select **[all|distinct]** *expression* { *, expression* }
from *tablename* [*var*] { *, tablename* [*var*] }
[where *search-condition*]
[group-by *attrname* { *, attrname* }]
[having *search-condition*];

- Παραδείγματα:

1. Βρείτε τα ids και τα ονόματα των πρακτόρων που έχουν ως έδρα τη Νέα Υόρκη.

$$\pi_{aid, name}(\sigma_{city="New York"}(AGENTS))$$

```
select aid, aname from agents  
where city="New York";
```

SQL – Data Manipulation Language

- Παραδείγματα:

2. Δώστε το περιεχόμενο της σχέσης customers.

```
select * from customers;
```

3. Βρείτε τα ids των προϊόντων για τα οποία υπάρχει παραγγελία.

```
select pid from orders;
```

- Το αποτέλεσμα θα περιέχει διπλότυπα (αφού είναι πολύ πιθανό να έχουν παραγγελθεί τα ίδια προϊόντα σε διαφορετικές παραγγελίες). Για να επιστραφούν μόνο διακριτές πλειάδες χρησιμοποιούμε τη λέξη-κλειδί **distinct**:

```
select distinct pid from orders;
```

- Η λέξη-κλειδί **all** επιτυγχάνει το αντίθετο αποτέλεσμα: επιστρέφονται όλες οι πλειάδες του αποτελέσματος. Είναι η προεπιλεγμένη λειτουργία για την εντολή select.

SQL – Data Manipulation Language

- Παραδείγματα:

4. Βρείτε όλα τα ζεύγη ονομάτων πελατών-πρακτόρων όπου ο πελάτης κάνει παραγγελία μέσω του πράκτορα.

$$\pi_{cname,aname} \left(\left(\pi_{cid,cname}(CUSTOMERS) \bowtie ORDERS \right) \bowtie AGENTS \right) \text{ ή } \pi_{cname,aname} \left(\begin{array}{c} \sigma_{(CUSTOMERS.cid=ORDERS.cid) \wedge (ORDERS.aid=AGENTS.cid)} \\ (CUSTOMERS \times ORDERS) \times AGENTS \end{array} \right)$$

$$\left\{ \begin{array}{l} cname, aname \end{array} \right\} \left\{ \begin{array}{l} (\exists cid, city, dis, aid, acity, perc, ord, mon, pid, qty, dl) \\ customers(cid, cname, city, dis) \wedge \\ agents(aid, aname, acity, perc) \wedge \\ orders(ord, mon, cid, aid, pid, qty, dl) \end{array} \right\}$$

```
select distinct customers.cname, agents.aname
from customers, orders, agents
where customers.cid = orders.cid and
orders.aid = agents.aid;
```


SQL – Data Manipulation Language

- Παραδείγματα:

4. Βρείτε όλα τα ζεύγη ονομάτων πελατών-πρακτόρων όπου ο πελάτης κάνει παραγγελία μέσω του πράκτορα.

Σημείωση: η σύζευξη όπως βλέπουμε υλοποιείται παίρνοντας το καρτεσιανό γινόμενο των σχέσεων στην πρόταση **from**, ακολουθούμενο από μια επιλογή σύμφωνα με τις συνθήκες στην πρόταση **where**.

```
select distinct customers.cname, agents.aname  
from customers, orders, agents  
where customers.cid = orders.cid and  
orders.aid = agents.aid;
```

SQL – Data Manipulation Language

- Παραδείγματα:

5. Υποθέστε ότι το κέρδος από ένα προϊόν υπολογίζεται πολλαπλασιάζοντας την ποσότητα επί την τιμή και στη συνέχεια αφαιρώντας το 60% για κόστος χονδρικής, την έκπτωση (σε %) του πελάτη και την προμήθεια (σε %) του πράκτορα. Βρείτε το κέρδος για κάθε παραγγελία.

```
select ordno, x.cid, x.aid, x.pid,  
        .40*(x.qty*p.price)-  
        .01*(c.discnt+a.percent)*(x.qty*p.price)  
from orders x, customers c, agents a, products p  
where c.cid=x.cid and a.aid=x.aid and p.pid=x.pid;
```

Σημείωση: τα x, c, a, p είναι **ψευδώνυμα (aliases)** για τις σχέσεις και ισχύουν μόνο στο πλαίσιο της εντολής που δηλώνονται.

SQL – Data Manipulation Language

- Στην προηγούμενη εντολή η SQL εκτελεί **μετατροπή τύπου** στην πρόσθεση ακεραίου με αριθμό κινητής υποδιαστολής, με το αποτέλεσμα να είναι αριθμός κινητής υποδιαστολής.
- Επιπλέον δε δίνεται όνομα για το γνώρισμα που σχετίζεται με την αριθμητική έκφραση. Τα περισσότερα συστήματα δίνουν ένα όνομα (π.χ. **COL5**) σε τέτοιες περιπτώσεις ή επιτρέπουν στο χρήστη να δηλώσει όνομα:

```
select ordno, x.cid, x.aid, x.pid,  
      .40*(x.qty*p.price) -  
      .01*(c.discnt+a.percent)*(x.qty*p.price) as profit  
from orders x, customers c, agents a, products p  
where c.cid=x.cid and a.aid=x.aid and p.pid=x.pid;
```

SQL – Data Manipulation Language

- Παραδείγματα:

6. Βρείτε όλα τα ζεύγη πελατών που μένουν στην ίδια πόλη.

$C1 := CUSTOMERS, C2 := CUSTOMERS$

$\pi_{C1.cid, C2.cid}(\sigma_{C1.cid < C2.cid}(C1 \bowtie_{C1.city = C2.city} C2))$

```
select c1.cid, c2.cid  
from customers c1, customers c2  
where c1.city=c2.city and c1.cid<c2.cid;
```

Σημείωση: χωρίς τη χρήση ψευδωνύμων η παραπάνω επερώτηση δε μπορεί να απαντηθεί στην SQL.

SQL – Data Manipulation Language

- Παραδείγματα:

7. Βρείτε τα ids των προϊόντων που έχουν παραγγελθεί από τουλάχιστον δυο πελάτες.

```
select distinct x1.pid  
from orders x1, orders x2  
where x1.pid=x2.pid and x1.cid<x2.cid;
```

Σημείωση: τα ψευδώνυμα x1 , x2 μπορούν να θεωρηθούν ως **μεταβλητές περιοχής**, με κάθε μία να εκτείνεται στο σύνολο των πλειάδων της σχέσης ORDERS.

SQL – Data Manipulation Language

- Παραδείγματα:

8. Βρείτε τα ids των πελατών που παραγγέλνουν προϊόντα για τα οποία έχει γίνει παραγγελία μέσω του πράκτορα a06.

$$\left\{ cid \left| \begin{array}{l} (\exists ord1, mon1, aid1, pid, qty1, dl1) \\ orders(ord1, mon1, cid, aid1, pid, qty1, dl1) \wedge \\ (\exists ord2, mon2, cid2, qty2, dl2) \\ orders(ord2, mon2, cid2, a06, pid, qty2, dl2) \end{array} \right. \right\}$$

```
select distinct x.cid
from orders x, orders y
where x.pid=y.pid and x.aid='a06' ;
```

SQL – Data Manipulation Language

ordno	month	cid	aid	pid	qty	dollars
1011	jan	<u>c001</u>	a01	p01	1000	450.00
1012	jan	<u>c001</u>	a01	p01	1000	450.00
1019	feb	c001	a02	p02	400	180.00
1017	feb	<u>c001</u>	a06	<u>p03</u>	600	540.00
1018	feb	c001	a03	p04	600	540.00
1023	mar	c001	a04	p05	500	450.00
1022	mar	c001	a05	p06	400	720.00
1025	apr	c001	a05	p07	800	720.00
1013	jan	<u>c002</u>	a03	p03	1000	880.00
1026	may	<u>c002</u>	a05	p03	800	704.00
1015	jan	c003	a03	p05	1200	1104.00
1014	jan	c003	a03	p05	1200	1104.00
1021	feb	<u>c004</u>	a06	<u>p01</u>	1000	460.00
1016	jan	<u>c006</u>	a01	p01	1000	500.00
1020	feb	c006	a03	p07	600	600.00
1024	mar	c006	a06	<u>p01</u>	800	400.00

SQL – Data Manipulation Language

- Εμφωλευμένες επερωτήσεις (Nested Queries)

- Μια εντολή **select** μπορεί να περιέχει μια άλλη εντολή **select** αλλά όχι αυθαίρετα.
- Απαίτηση: μια σχέση στην πρόταση **from** δε μπορεί να είναι το αποτέλεσμα μιας εντολής **select**, πρέπει να είναι μια υπάρχουσα σχέση.
- 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**

SQL – Data Manipulation Language

- Το κατηγορημα **IN**

- Χρησιμοποιείται για να ελέγξει αν πλειάδες ανήκουν σε μια σχέση

- Παραδείγματα:

1. Βρείτε τα ids των πελατών που κάνουν παραγγελίες μέσω πρακτόρων με έδρα το Duluth ή το Dallas.

Η παρακάτω εντολή επιστρέφει όλα τα ids των πρακτόρων με έδρα το Duluth ή το Dallas:

```
select aid from agents  
where city='Duluth' or city='Dallas';
```

SQL – Data Manipulation Language

- Το κατηγορημα **IN**

- Χρησιμοποιείται για να ελέγξει αν πλειάδες ανήκουν σε μια σχέση

- Παραδείγματα:

1. Βρείτε τα ids των πελατών που κάνουν παραγγελίες μέσω πρακτόρων με έδρα το Duluth ή το Dallas.

Η παρακάτω εντολή επιστρέφει όλα τα ids των πελατών από εκείνες τις πλειάδες των οποίων η τιμή aid ανήκει στα αποτελέσματα της προηγούμενης εντολής η οποία έχει τοποθετηθεί ως subselect:

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

SQL – Data Manipulation Language

- Παραδείγματα:

2. Βρείτε όλη την πληροφορία για πράκτορες με έδρα το Duluth ή το Dallas. Εδώ το **in** χρησιμοποιείται με ρητά απαριθμημένο σύνολο:

```
select * from agents  
where city in ( 'Duluth' , 'Dallas' );
```

3. Βρείτε τα ονόματα και τις εκπτώσεις των πελατών που κάνουν παραγγελίες μέσω πρακτόρων με έδρα το Duluth ή το Dallas.

```
select cname, discnt from customers  
where cid in (select cid from orders  
where aid in (select aid from agents  
where city in ( 'Duluth' , 'Dallas' )) );
```

Σημείωση: τα ονόματα γνωρισμάτων τα οποία δεν είναι συσχετισμένα με όνομα σχέσης συσχετίζονται αυτόματα με τη σχέση της κοντινότερης εντολής select.

SQL – Data Manipulation Language

- Παραδείγματα:

4. Βρείτε τα ονόματα των πελατών που παραγγέλνουν το προϊόν p05.

```
select distinct cname from customers, orders  
where customers.cid=orders.cid and pid='p05';
```

ή

```
select cname from customers  
where 'p05' in (select pid from orders  
where cid=customers.cid);
```

SQL – Data Manipulation Language

- Παραδείγματα:

5. Βρείτε τα ονόματα των πελατών που παραγγέλλουν το προϊόν p07 μέσω του πράκτορα a03.

```
select cname from customers where  
orders.aid='a03' and 'p07' in (select pid  
from orders where cid=customers.cid);
```

Η παραπάνω εντολή δεν είναι σωστή διότι έχουμε αναφορά στο orders.aid σε σημείο εκτός του πεδίου της orders του subselect. Το σωστό είναι το εξής:

```
select cname from customers, orders where  
orders.aid='a03' and 'p07' in (select pid  
from orders where cid=customers.cid);
```

SQL – Data Manipulation Language

- Παραδείγματα:

6. Βρείτε τους αριθμούς παραγγελιών για όλες τις παραγγελίες που κάνουν πελάτες στο Duluth μέσω πρακτόρων στη Νέα Υόρκη.

```
select ordno from orders  
where cid,aid in (select cid,aid  
from customers c, agents a where  
c.city='Duluth' and a.city='New York' );
```

Η παραπάνω εντολή επιτρέπεται από την έκδοση SQL-92 και μετά. Στις προηγούμενες εκδόσεις δεν επιτρέπεται να ελέγχουμε ζεύγη με το κατηγορήμα **in** και η παραπάνω επερώτηση αντικαθίσταται με:

```
select ordno from orders where cid in  
(select cid from customers where city='Duluth' )  
and aid in  
(select aid from agents where city='New York' );
```

SQL – Data Manipulation Language

- Το κατηγορημα **NOT IN**

7. Βρείτε τα ids των πρακτόρων που κάνουν παραγγελίες για πελάτες που **δε** μένουν στη Νέα Υόρκη.

```
select aid from agents
where aid in
  (select aid from orders
where cid not in
  (select cid from customers
where city='New York' ));
```