

# Βάσεις Δεδομένων




Σχεσιακή Σχεδίαση ΒΔ:  
Συναρτησιακές Εξαρτήσεις





# Επεκτάσεις του Σχεσιακού Μοντέλου

# Παραδείγματα

- **KMBC  $\subset$  3KM  $\subset$  2KM  $\subset$  1KM**
- Είναι σε 1KM αλλά όχι σε 2KM (διότι έχει μερική Σ.Ε.)
  - ΠΡΟΜΗΘΕΙΑ (κωδπ, κωδε, κωδα, όνομαπ, όνομαε, ποσότητα)
- Σε 2KM αλλά όχι σε 3KM (διότι έχει μεταβατική Σ.Ε.)
  - ΥΤ (κωδυ, όνομα, μισθός, κωδτ, όνοματ, όροφ, διευθ)
- 3KM, όχι KMBC (διότι έχει Σ.Ε. που οδηγεί σε μέρος υποψ. κλειδ.)
  - ΓΕΥΜΑ (ανθρ, τροφή, #απόδειξης)
- KMBC
  - ΥΠ (κωδυ, όνομα, μισθός, κωδτ)



# «Αλγόριθμος» Κανονικοποίησης

- Για κάθε βασική Σ.Ε. που είναι μερική ή μεταβατική (A-?B) στη σχέση R:
  - δημιουργούμε νέα σχέση **R'** με χαρακτηριστικά AUB και πρωτεύον κλειδί το A,
  - αφαιρούμε από την **R** τα χαρακτηριστικά B.

Αλγόριθμος  
Κανονικοποίησης—  
ξένα κλειδιά

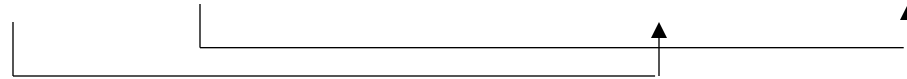
Τα υπάρχοντα ξένα κλειδιά ακολουθούν το που πηγαίνει το πρωτεύον κλειδί από το οποίο παίρνουν τιμές.

Καθώς εξαλείφεται η  $A \rightarrow B$ , το  $A$  που μένει στην παλιά σχέση  $R$  γίνεται ξένο κλειδί στη  $R'$ .

Κάθε Σ.Ε. (συμπεριλαμβανομένων και των κλειδιών) ελέγχεται σε μια σχέση στο νέο σχήμα.

# Αλγόριθμος Κανονικοποίησης: Παράδειγμα

- 1 NF
- ΠΡΟΜΗΘΕΙΑ (κωδπ, κωδε, κωδα, όνομαπ, όνομαε, ποσότητα)



- **BCNF**
- ΠΡΟΜΗΘΕΙΑ (κωδπ, κωδε, κωδα, ποσότητα)
- ΠΡΟΜΗΘΕΥΤΗΣ (κωδπ, όνομαπ)      ΕΡΓΟ (κωδε, όνομαε)


# Αλγόριθμος Κανονικοποίησης: Παράδειγμα

- 2NF
- ΥΤ (κωδυ, όνομα, μισθός, κωδτ, όνοματ, όροφ, διευθ)



- **BCNF**
- R ΥΠ (κωδυ, όνομα, μισθός, κωδτ)
- R' ΤΜΗΜΑ (κωδτ, όνοματ, όροφ, διευθ)

# Αλγόριθμος Κανονικοποίησης: Παράδειγμα

- 3NF
- ΓΕΥΜΑ (κωδα, πιάτο, #απόδειξης)  

- Είναι 3NF διότι υπάρχει μια προβληματική Σ.Ε. η οποία είναι από αυτές που οδηγούν σε μέρος υποψήφιου (πρωτεύοντος) κλειδιού.
- Είναι από αυτές που «ανεχόμαστε» στην 3NF, αλλά όχι στην BCNF.
- Αν εφαρμόσουμε τον αλγόριθμο, παίρνουμε τη Σ.Ε. που είναι προβληματική, βάζουμε το AUB (όλα) σε νέα σχέση (π.χ. R') κι αφαιρούμε το B από την πρώτη σχέση. Το A της γίνεται πρωτεύον κλειδί.



Αλγόριθμος  
Κανονικοποίησης:  
Παράδειγμα

- **3NF**
- ΓΕΥΜΑ (κωδα, πιάτο, #απόδειξης)
  
- **BCNF**
- ΓΕΥΜΑ (κωδα, #απόδειξης)
- ΠΙΑΤΟ (#απόδειξης, πιάτο)
  
- Υπάρχει κάποιο πρόβλημα στο παραπάνω;
  - Το πρωτεύον κλειδί κωδα, πιάτο είναι υποψήφιο αλλά όχι το μόνο.
  - Το κωδα, #αποδειξης είναι επίσης υποψήφιο κλειδί.



## Κανονικοποίηση —κόστος ελέγχου

- Όταν έχουμε Σ.Ε. που οδηγούν σε μέρη υποψήφιων κλειδιών, αν μετατραπούν σε BCNF, ο έλεγχος των Σ.Ε. αυτών απαιτεί τη ζεύξη πολλών πινάκων που **κοστίζει**.
- Άρα, ενώ, ιδανικά, τα σχήματα πρέπει να είναι BCNF, σε κάποιες περιπτώσεις, **συμβιβάζομαστε και προτιμάμε την 3NF** όπου αυτές οι Σ.Ε. ελέγχονται σε έναν πίνακα.
- Δηλαδή προτιμάμε την **ταχύτητα** ελέγχου από την **καθαρότητα** και την **μη επανάληψη** πληροφορίας.

# Άλλες εξαρτήσεις & κανονικές μορφές

- Υπάρχουν κι άλλες εξαρτήσεις που δημιουργούν πλεονασμούς, π.χ. πλειότιμες, επαγωγικές, κ.λπ.
- Η απαλοιφή αυτών οδηγεί σε υψηλότερες κανονικές μορφές, π.χ 4NF, 5NF, ...
- Στην πράξη, οι Σ.Ε. είναι αυτές που έχουν σημασία.



# Ασκηση 1

Δίδεται ο παρακάτω πίνακας της απασχόλησης του προσωπικού σε μια κατασκευαστική εταιρεία

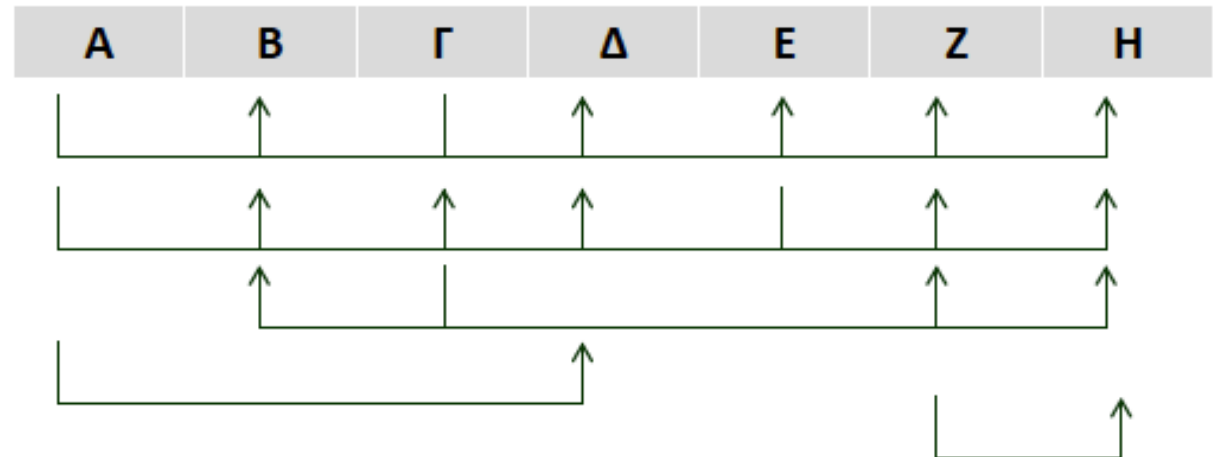
ΑΜΚΑ	Όνομ/μο	Διεύθυνση	Ειδικότητα	Κωδ. Έργου	Τίτλος Έργου	Προϋπολογισμός	Αντικείμενο Εργασίας	Από	Έως	Κωδικός Τμήματος που εργάζεται	Τίτλος Μονάδας
ΑΜΚΑ	Sname	Address	Speciality	ProjectID	ProjectTitle	Budget	Job	Start	Finish	BranchNo	BranchTitle
1010001101	Κώτσος Κ.	Αθηνών 101	Ηλεκτρολόγος	101	Κτήριο Εταιρείας ABC	12000000	Ηλεκτρολογικό Δίκτυο	2/3/2013	31/3/2013	T1	Τεχνική Μονάδα
				102	Νοσοκομείο A	40000000	Φωτισμοί Εξ. Χώρων	1/4/2013	10/4/2013	T1	Τεχνική Μονάδα
				101	Κτήριο Εταιρείας ABC	12000000	Τοποθέτηση ηλ. Συσκευών	2/3/2013	31/3/2013	T1	Τεχνική Μονάδα
1010001101	Σωτηρίου Π.	Πάτρας 201	Πολιτικός Μηχανικός	102	Νοσοκομείο A	40000000	Μελέτη	1/1/2013	31/1/2013	T3	Μελετών
				101	Κτήριο Εταιρείας ABC	12000000	Μελέτη	1/9/2012	10/1/2012	T3	Μελετών
				105	Νοσοκομείο A	40000000	Επίβλεψη	1/1/2013	31/12/2013	T3	Μελετών

# Ασκηση 1

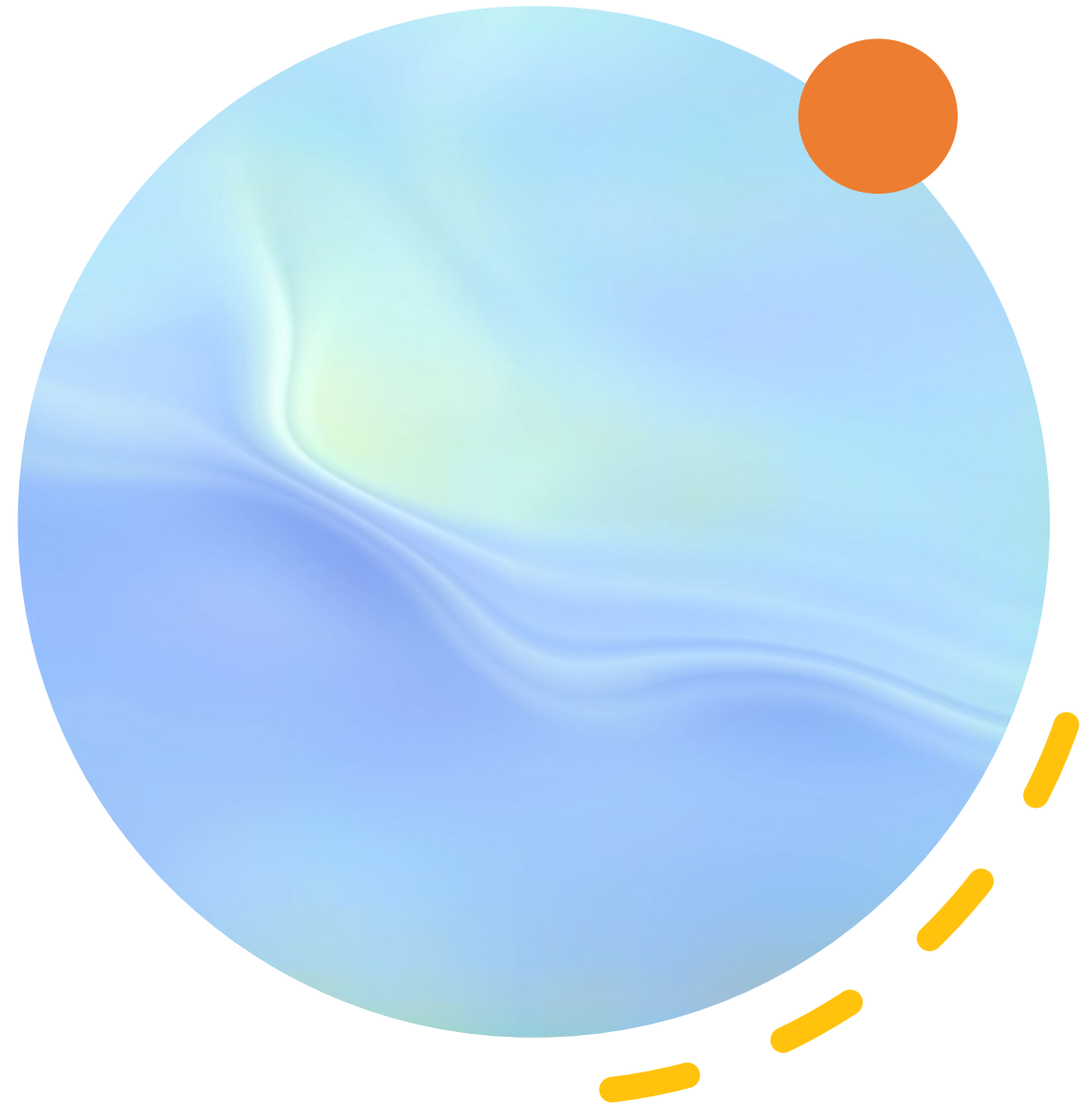
1. Γιατί ο πίνακας δεν είναι κανονικοποιημένος κατά 1NF;
2. Ποιές είναι οι δυσλειτουργίες που θα έχουμε αν αφήσουμε τη Σχέση σε αυτή τη μορφή;
3. Ποιες από τις παρακάτω Εξαρτήσεις είναι πραγματικές;
  1. AMKA -> Speciality, ProjectTitle ->job, AMKA ->BranchTitle
  2. AMKA -> Address, Address -> BranchTitle, ProjectTitle -> Budget
  3. ProjectID -> Job, AMKA -> Job, Job -> ProjectID
4. Να φερετε στον πίνακα σε κανονικοποιημένη μορφή κατά 1 NF
5. Ποιό είναι το πρωτεύον κλειδί που θα επιλέξετε;
6. Ποιές είναι συναρτησιακές εξαρτήσεις στον πίνακα 1 (Ολικές, μερικές, μεταβατικές)
7. Κανονικοποιήστε τον πίνακα κατά 2NF.

# Άσκηση 2

1. Δίδεται η παραπάνω σχέση με τις συναρτησιακές εξαρτήσεις:  
Προσδιορίστε τα πιθανά κλειδιά και καθορίστε το πρωτεύον κλειδί
2. Υποθέστε ότι η σχέση είναι σε 1NF (πρώτη κανονική μορφή),  
Περιγράψτε και επεξηγήστε τη διαδικασία της κανονικοποίησης των σχεσιακών σχημάτων στη δεύτερη (2NF) και τρίτη (3NF) κανονική μορφή.
3. Προσδιορίστε τα πρωτεύοντα και ξένα κλειδιά στην τελική μορφή (3NF).



# Επεκτάσεις Σχεσιακού Μοντέλου





# Αναφορικός περιορισμός

- Όταν τιμές ενός πεδίου πρέπει να υπάρχουν και ως τιμές σε άλλο πεδίο(άλλης) σχέσης.
- Τα ξένα κλειδιά είναι ειδική περίπτωση όπου το άλλο πεδίο είναι πρωτεύον κλειδί.
- Π.χ. Έστω γονείς & παιδιά, ποιο είναι το βασικό αυτοκίνητο με το οποίο μετακινούνται;
  - ΓΟΝΙΟΣ (κωδ, βασικό\_αυτοκίνητο)
  - ΠΑΙΔΙ (κωδ, βασικό\_αυτοκίνητο)
- ΠΑΙΔΙ.αυτοκίνητο  $\subseteq$  ΓΟΝΙΟΣ.αυτοκίνητο

# Αναφορικός περιορισμός

- Στη συντριπτική πλειοψηφία τους, οι χρήσιμοι αναφορικοί περιορισμοί είναι ξένα κλειδιά
- (όμως είναι χρήσιμο να ξέρουμε ότι υπάρχουν κι άλλες περιπτώσεις)

# Μηδενικές τιμές (NULL values)

- Μηδενική τιμή = μια μη καταχωρημένη τιμή όταν:
  - **Άγνωστη**(τιμή υπάρχει αλλά δεν είναι γνωστή),
  - **Μη διαθέσιμη** (η τιμή είναι γνωστή αλλά μη διαθέσιμη),
  - **Δεν ορίζεται** (δεν υπάρχει, δεν ισχύει)
- Π.χ. ο Λάκης δεν έχει τηλέφωνο
  - ή έχει αλλά δεν το γνωρίζουμε
  - ή έχει, το γνωρίζουμε, αλλά δεν θέλουμε να το αποκαλύψουμε.
- Ένα πρωτεύον κλειδί δεν έχει ποτέ μηδενική τιμή NULL



## Ιεραρχίες γενίκευσης – ειδίκευσης

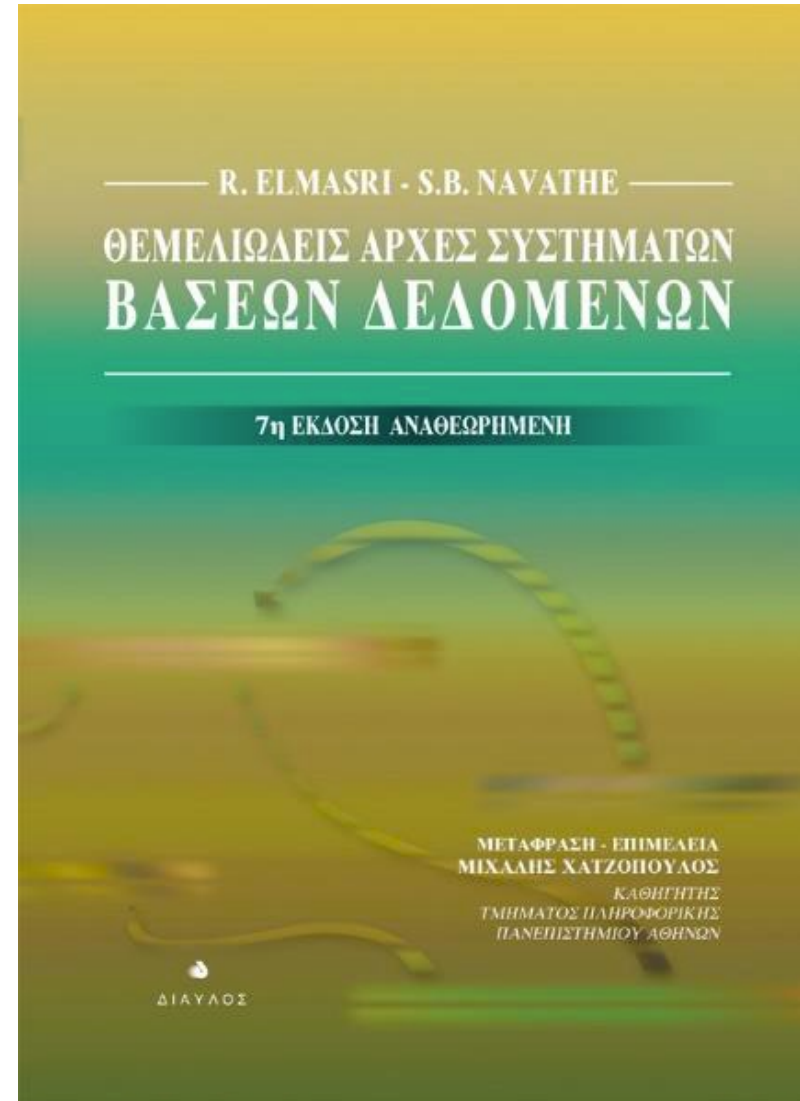
- Νοητικές ιεραρχίες κατηγοριών & υποκατηγοριών «αντικειμένων».
- Μπορούν να εκφραστούν:
  - είτε ως τιμές πεδίων
  - είτε ως διαφορετικοί πίνακες με ανάλογες συνδέσεις κλειδιών
  - είτε ως διαφορετικοί πίνακες και επανάληψη πεδίων
  - ... κ.λπ. ανάλογα με τις ανάγκες (και τον οίστρο του σχεδιαστή!)

# Ιεραρχίες γενίκευσης –ειδίκευσης - παράδειγμα

- Αλκοόλ -> ποτό -> τροφές (το αλκοόλ είναι μια κατηγορία ποτού που είναι μια κατηγορία τροφής).
- Toyota -> αυτοκίνητο -> όχημα -> αντικείμενο
  
- ΟΧΗΜΑ (κωδ, βάρος, max-ταχύτητα...)
- ΑΥΤΟΚΙΝΗΤΟ (κωδ, 0-100, θέσεις)
- ΕΛΙΚΟΠΤΕΡΟ (κωδ, ύψος, θέσεις, #ελίκων)
  
- ΟΧΗΜΑ (κωδ, βάρος, max-ταχύτητα)
- ΑΥΤΟΚΙΝΗΤΟ (κωδ, βάρος, 0-100, θέσεις)
- ΕΛΙΚΟΠΤΕΡΟ (κωδ, βάρος, ύψος, θέσεις, #ελίκων)

# Ευχαριστώ!

- <https://eclass.uoa.gr/courses/DIND136/> Έγγραφα > Διαλέξεις
- Βιβλία:
  - Ενότητα 6
    - Κεφάλαιο 14 και 15  
(Βασικά για  
Συναρτησιακές  
Εξαρτήσεις και  
Κανονικοποίηση ΒΔ)



Ταξινομημένες  
σχέσεις

Όταν η σειρά των πλειάδων έχει σημασία

Π.χ. τιμές μετοχών, μετρήσεις παραμέτρων,  
κείμενο λέξεων, ...στο χρόνο, ...στο χώρο.



Η οποιαδήποτε ταξινόμηση πλειάδων  
γίνεται πέρα από το σχεσιακό μοντέλο κι  
απαιτεί ειδική εντολή

Structured  
Query  
Language

SQL



# Structured Query Language

- Γλώσσα Δομημένων Ερωτημάτων

# Structured Query Language

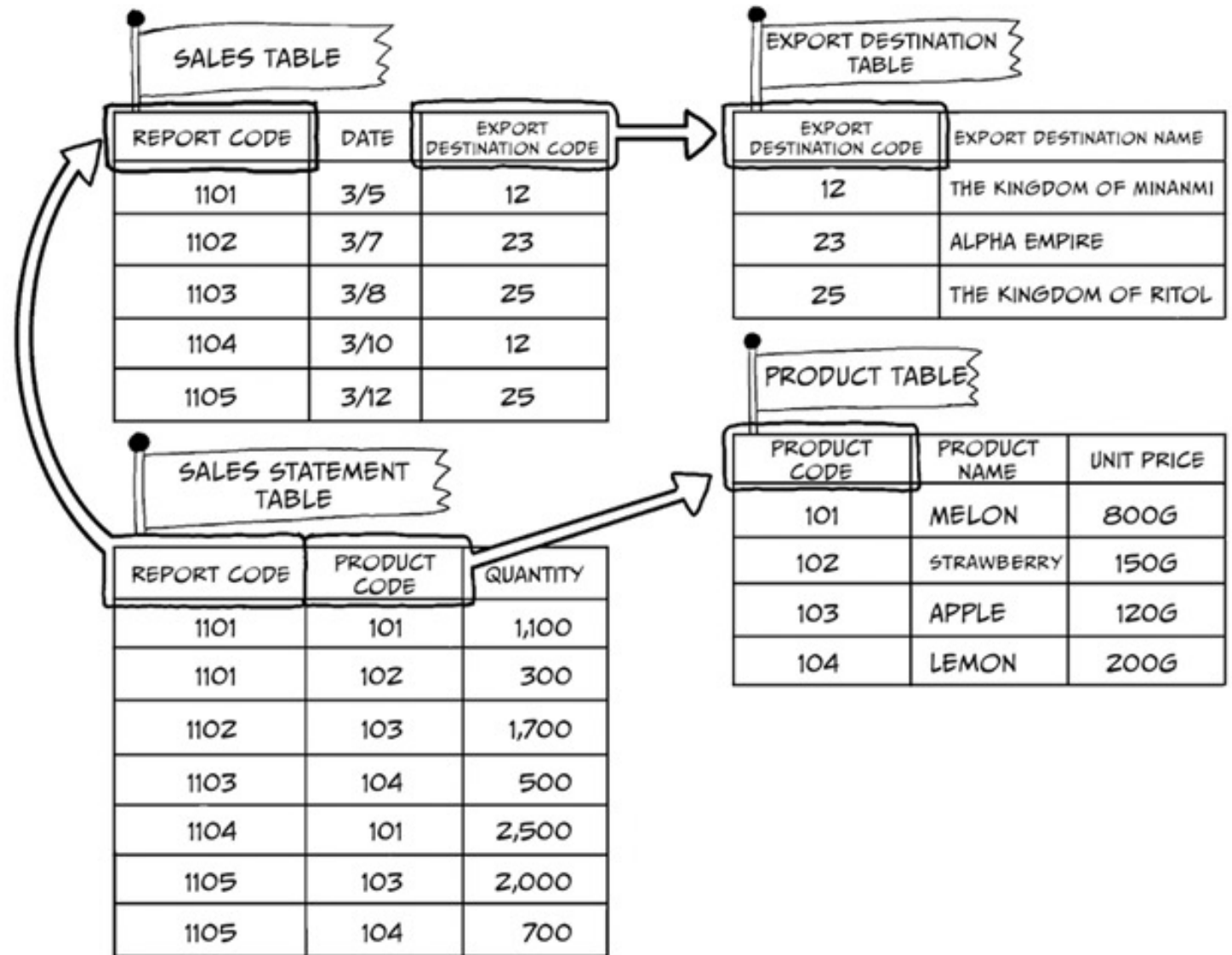
Η γλώσσα που χρησιμοποιούμε  
για τη:

- Δημιουργία σχέσεων (πινάκων) μιας ΒΔ
- Ενημερώσεις (εισαγωγή, κ.λπ. δεδομένων)
- Ανάκτηση δεδομένων

**Ερωτήματα!**

# Παράδειγμα

- Με SQL:
  - δημιουργούμε το σχήμα στη ΒΔ
  - γεμίζουμε τους πίνακες με δεδομένα
  - κάνουμε ερωτήματα



Structured  
Query  
Language

---

Είναι δηλωτική γλώσσα  
(declarative) **Τι!**

---

Σε αντίθεση με γλώσσες  
όπως η C, C++, Java κ.λπ.

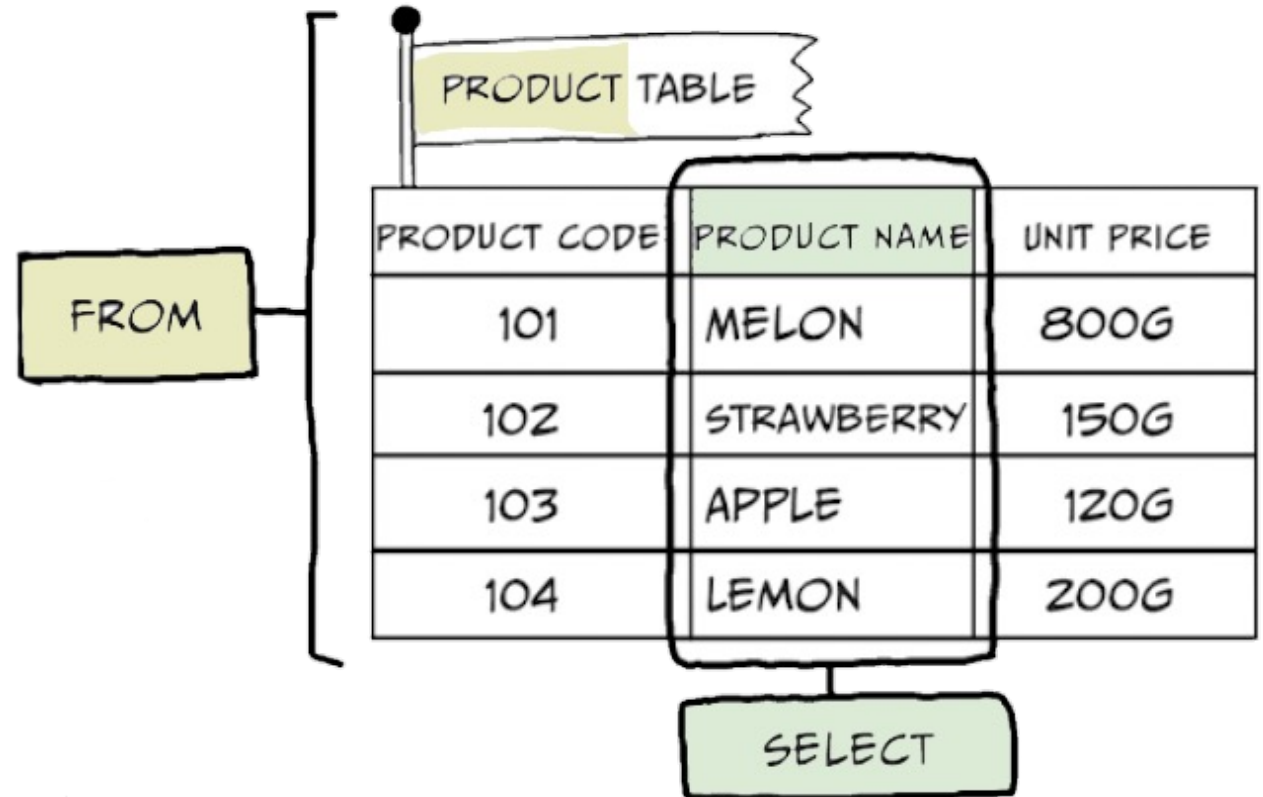
**Τι και πως!**

# SQL: βασικές εντολές

- **Select** → για ερωτήματα
  - Insert
  - Delete
  - Update
- για ενημερώσεις
- Create table → για λογική σχεδίαση
  - Create view → για σχεδίαση εξωτ. σχημάτων
  - Create index → για φυσική σχεδίαση
- 
- Κάθε ερώτημα εφαρμόζεται σε ένα σύνολο σχέσεων και παράγει **μια** σχέση

# SQL: SELECT-FROM

- Έστω ότι θέλουμε να δούμε όλα τα προϊόντα (τα ονόματά τους):
- **SELECT product\_name FROM product**



PRODUCT NAME
MELON
STRAWBERRY
APPLE
LEMON

## SQL: SELECT-FROM

---

- `SELECT product_name FROM product`
- Επιστρέφει το αποτέλεσμα:

# SQL: SELECT-FROM-WHERE

Όλες τις πληροφορίες (χαρακτηριστικά) για τα φρούτα με τιμή μονάδας μεγαλύτερη ή ίση με 200:

- SELECT \*
- FROM product
- WHERE unit\_price >= 200

```
SELECT *  
  
FROM product  
  
WHERE unit_price >= 200
```

PRODUCTS THAT COST  
200G OR MORE

PRODUCT CODE	PRODUCT NAME	UNIT PRICE
101	MELON	800G
104	LEMON	200G



# SQL: SELECT-FROM-WHERE

- Έστω ΥΠ (όνομα, ηλικία, τμήμα) και θέλουμε τα ονόματα υπαλλήλων άνω των 30 ετών:
- SELECT u.όνομα
- FROM ΥΠ u
- WHERE u.ηλικία >30

# SQL: τμήμα FROM

- Το **u** είναι **μεταβλητή πλειάδας**. Δηλώνεται στο τμήμα **FROM** της εντολής
- **Δεικτοδοτημένη** μεταβλητή πλειάδας: *μεταβλητή πλειάδας.χαρακτηριστικό* (του αντίστοιχου πίνακα)
- Αντιπροσωπεύει την τιμή του πεδίου της πλειάδας:
  - **u.όνομα**
  - **u.ηλικία**

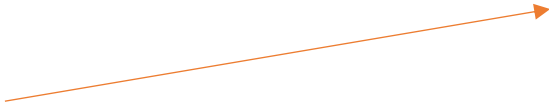
# SQL: τμήμα WHERE

- Μια **μεταβλητή πλειάδας** διατρέχει τον πίνακα στον οποίο αναφέρεται κι αντιπροσωπεύει την κάθε πλειάδα του.
- Το  $u.ηλικία > 30$  στο **WHERE** αποτελεί τη **συνθήκη** που πρέπει να ικανοποιεί μια πλειάδα για να συμμετέχει στο αποτέλεσμα.
- Το  $u.όνομα$  στο **SELECT** αποτελεί τη **λίστα αποτελέσματος**, δηλ. τις τελικές στήλες που θα δώσει η εντολή ως αποτέλεσμα

# SQL: τμήμα SELECT

- SELECT <λίστα χαρακτηριστικών>
- FROM <λίστα πινάκων>
- WHERE <συνθήκη>

- Τελεστές
- = < <= > >= <>



Έκφραση αναζήτησης που προσδιορίζει τις πλειάδες που πρέπει να ανακτηθούν από την ερώτηση

- SQL standard: <>
- SQLnon-standard: !=

# SQL: συνθήκες

- Μια **συνθήκη επιλογής** είναι μια σύγκριση μεταξύ ενός χαρακτηριστικού μιας μεταβλητής πλειάδας και μιας σταθεράς ή μεταξύ δύο χαρακτηριστικών της ίδιας μεταβλητής πλειάδας.
  - μετ.χαρασταθερά (π.χ. u.ηλικία > 30)
  - μετ.χαρ1 μετ.χαρ2 (π.χ. u.όνομα <> u.τμήμα)
- Όπου τ ανήκει {= < <= > >= <>}

# SQL: συνθήκες

- Μια **συνθήκη ζεύξης** είναι μια σύγκριση μεταξύ χαρακτηριστικών δύο διαφορετικών μεταβλητών πλειάδων
  - μετ.χαρ1 τ μετ.χαρ2 (π.χ. u.κωδτ= τ.τμήμα)
- Όπου τ ανήκει {= < <= > >= <>}

# SQL: συνθήκες

- Η συνθήκη ενός ερωτήματος στο **WHERE** είναι ένας λογικός συνδυασμός **συνθηκών επιλογής** και **ζεύξης** μέσω των λογικών τελεστών **and, or, not**.
- **(u.κωδτ= τ.τμ)and ((τ.όροφος=2)or (τ.προϋπ>1M))**
- **WHERE** ζευγάρια υπαλλήλου -τμήματος **ΚΑΙ** ή ο όροφος του τμήματος να είναι ο 2 ή ο προϋπολογισμός του τμήματος να είναι > από 1 εκατομμύριο

# SQL: συνθήκες

- **WHERE(u.ηλικία>30) or ((u.ηλικία <= 35) and (u.τμήμα=τ.όνομα) and (τ.όροφος<7))**



## SQL:καρτεσιανό γινόμενο

- Δύο σύνολα:
- $A = \{x,y,z\}$
- $B = \{1,2,3\}$
- Το καρτεσιανό γινόμενο  $A \times B$ :

$(x,1)$	$(x,2)$	$(x,3)$
$(y,1)$	$(y,2)$	$(y,3)$
$(z,1)$	$(z,2)$	$(z,3)$

# SQL: σημασιολογία εντολής SELECT

1. Δημιουργία καρτεσιανού γινομένου των σχέσεων κάθε μεταβλητής που ορίζεται στο **FROM**
2. Ελέγχουμε τη συνθήκη του **WHERE** σε κάθε πλειάδα του καρτ. γινομένου και διατηρούμε μόνο αυτές που ικανοποιούν τη συνθήκη.
3. Διαλέγουμε τα χαρακτηριστικά του **SELECT** από τις πλειάδες που έχουν μείνει και δημιουργούμε το αποτέλεσμα.
4. Προαιρετικά, διαμορφώνουμε το τελικό αποτέλεσμα έτσι ώστε να είναι ταξινομημένο (ORDER BY) ή να μην έχει διπλότυπα (προσδιορισμός DISTINCT)

## Βασική σύνταξη

---

SELECT [DISTINCT] <λίστα αποτελεσμάτων>

---

FROM <λίστα δηλώσεων μεταβλ. πλειάδων>

---

WHERE <συνθήκη>

---

[ORDER BY] <λίστα ταξινόμησης>

---

# Ευχαριστώ!

- <https://eclass.uoa.gr/courses/DIND136/> Έγγραφα > Διαλέξεις
- Βιβλία:
  - Κεφάλαιο 6
  - SQL

