



Διάλεξη 02: Βάσεις Δεδομένων - Εισαγωγή

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:
Εισαγωγή στις έννοιες:

- Αρχιτεκτονική Τριών Επιπέδων
- Χρήστες Βάσεων Δεδομένων
- Μοντέλα Δεδομένων και Γλώσσες
- Πότε να μην χρησιμοποιούμε Βάσεις Δεδομένων

Διδάσκων: Παναγιώτης Ανδρέου

Moodle enrollment key

moodle-epl342

Αρχιτεκτονική 3 Επιπέδων

• Εσωτερικό Επίπεδο

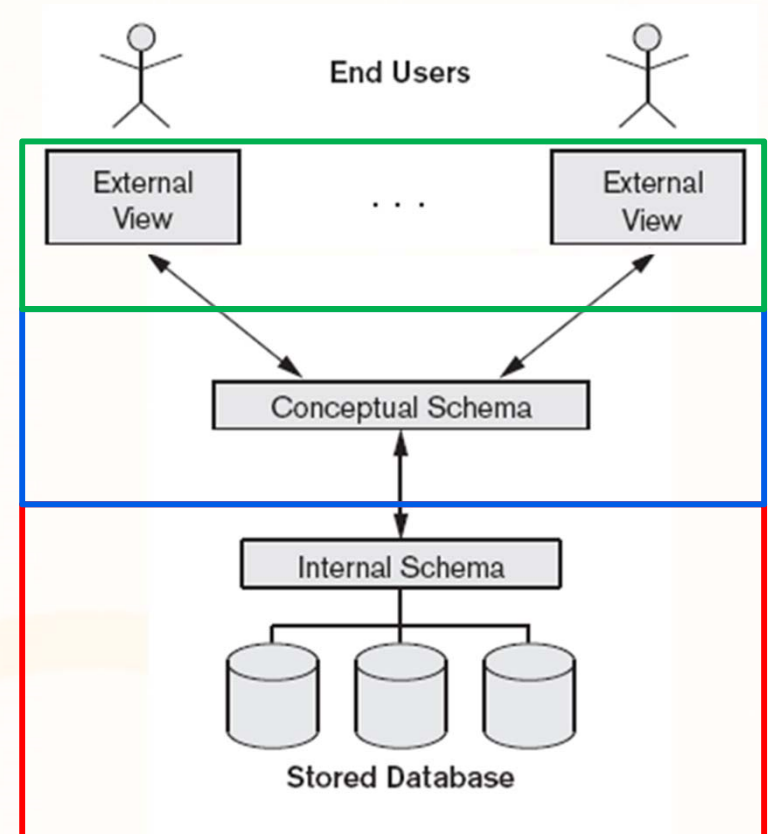
- Περιγράφει τη φυσική δομή αποθήκευσης της βάσης δεδομένων
- Π.χ., Πίνακας STUDENT αποθηκευμένος στο αρχείο X.dat

• Εννοιολογικό Επίπεδο

- Περιγράφει τη εννοιολογική δομή της βάσης δεδομένων στους εξωτερικούς χρήστες
- Π.χ., Ο πίνακας STUDENT αποτελείται από 3 στήλες [Ταυτότητα (int), Όνομα (String), Έτος (int)]

• Εξωτερικό Επίπεδο (όψεις)

- Περιγράφει μέρος της βάσης δεδομένων σε συγκεκριμένους χρήστες
- Π.χ., Όψη-1: Πρωτοετής φοιτητές (select * from STUDENT where Έτος=1),
Όψη-2: Δευτεροετής φοιτητές (select * from STUDENT where Έτος=2), ...



Αρχιτεκτονική (Οργάνωση) Υπολογιστών/Δικτύων

A. Κεντρική Αρχιτεκτονική (1-tier)

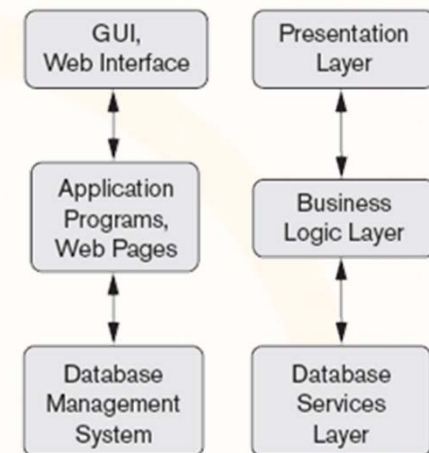
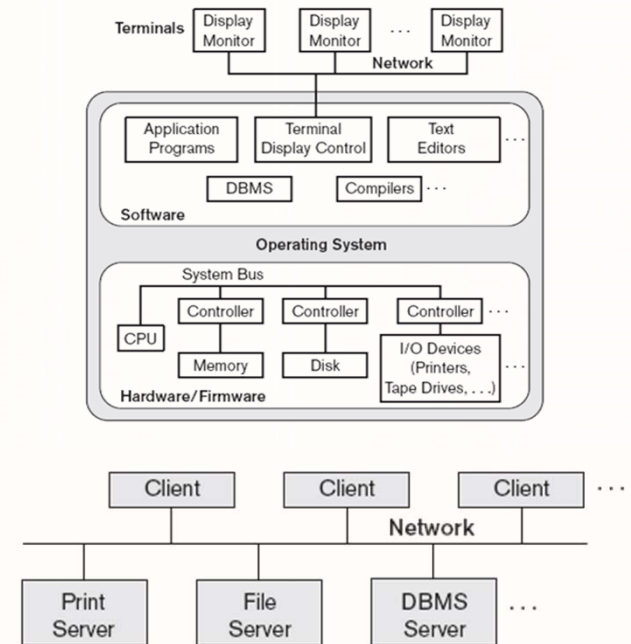
- Όλα τα συστατικά (εφαρμογή, βάση δεδομένων) βρίσκονται στην ίδια μηχανή

B. Αρχιτεκτονική Πελάτη-Εξυπηρετητή (2-tier)

- Πελάτης: Διαπροσωπείες, Εφαρμογές και μερική τοπική επεξεργασία
- Εξυπηρετητής: Σύστημα επεξεργασίας Επερωτήσεων, DBMS

C. Αρχιτεκτονική 3 ή περισσότερων επιπέδων (3-tier or n-tier)

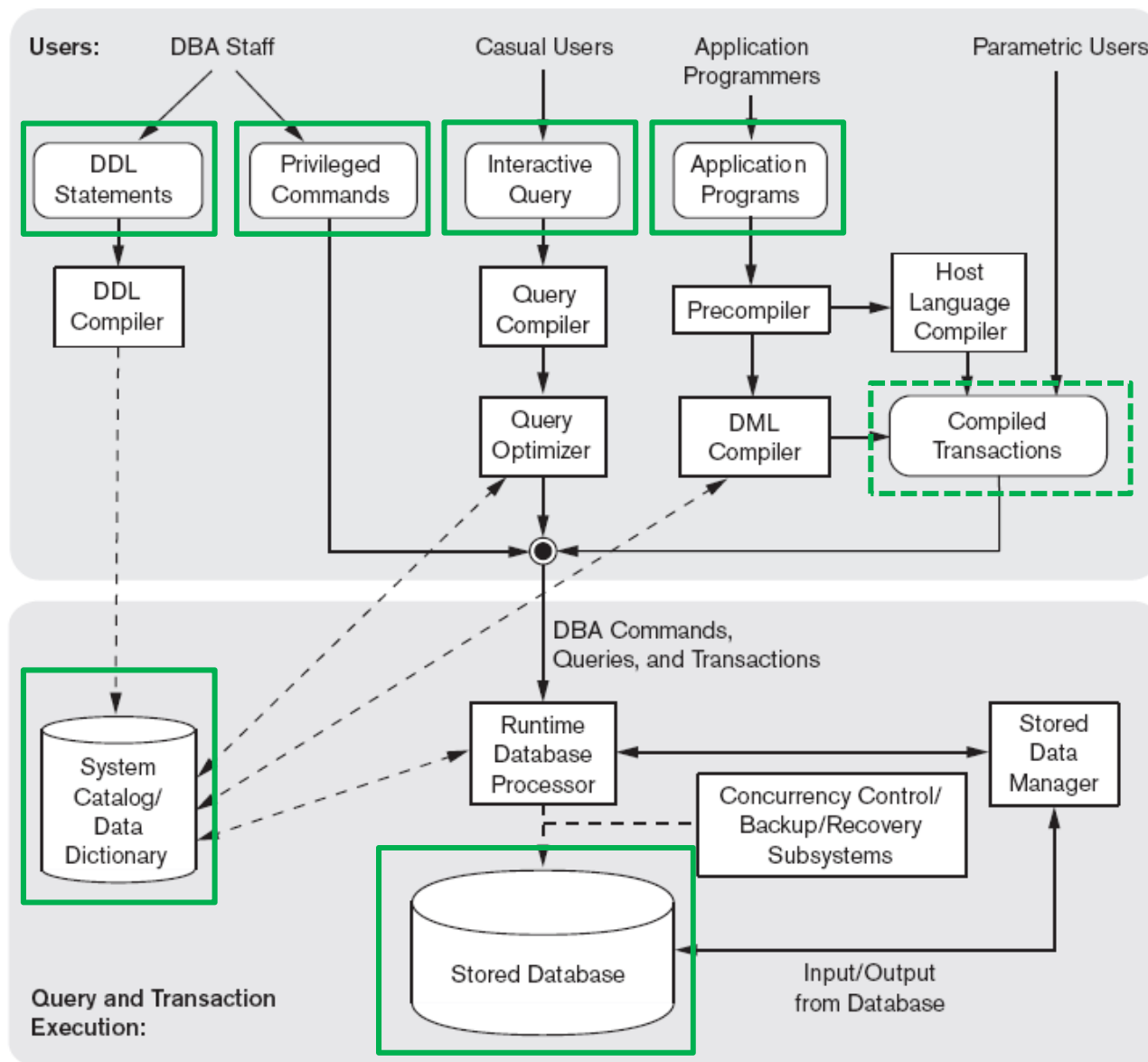
- Εκτός από πελάτη και εξυπηρετητή υπάρχουν και ενδιάμεσα επίπεδα
- Π.χ., Ο Web server παρέχει στον πελάτη τη διαπροσωπεία μέσω του δικού του browser



Χρήστες Συστημάτων Βάσεων Δεδομένων

- Οι Χρήστες Βάσεων μπορούν να χωριστούν σε δυο κατηγορίες:
- Αυτοί που Εργαζόμενοι Στο **Προσκήνιο**
 - Αναλυτές
 - Σχεδιαστές Βάσεων Δεδομένων
 - Προγραμματιστές Βάσεων Δεδομένων
 - Διαχειριστές Βάσεων Δεδομένων
 - Προγραμματιστές Εφαρμογών
 - Τελικοί Χρήστες
- Αυτοί που Εργάζονται στο **Παρασκήνιο**
 - Σχεδιαστές και προγραμματιστές DBMS
 - Προγραμματιστές Εργαλείων
 - Διαχειριστές Υλικού και Λογισμικού Εξυπηρετητή

Χρήστες Συστημάτων Βάσεων Δεδομένων (συν.)



Επιπρόσθετα Πλεονεκτήματα Χρήσης ΒΔ

Αναφέραμε ήδη **4 πλεονεκτήματα** των Βάσεων σε σχέση με **συμβατικά αρχεία**.

1. Αυτό-περιγραφική Φύση της ΒΔ (Κατάλογος)
2. Ανεξαρτησία Προγράμματος-Δεδομένων
3. Παροχή Πολλαπλών Όψεων
4. Διαμοιρασμός Φόρτου και Επεξεργασία Δοσοληψιών

Επιπρόσθετα πλεονεκτήματα μπορεί να αξιοποιήσει ο DBA

5) Έλεγχος των πλεονασμών (redundancy)

- Π.χ., Στο παράδειγμα του University μπορεί να υπάρχουν πολλές ανεξάρτητες βάσεις: π.χ., LIBRARY-DB, CS-DB, UCY-DB, etc.
- Αυτό δημιουργεί προβλήματα: Σπατάλη Χώρου (Storage Space Waste), Περιπτώσεις Ασυνέπειας (Inconsistencies), Σπατάλη Χρόνου (Duplication of effort)
- Ιδανικά θα θέλαμε η πληροφορία να **αποθηκεύεται σε μια μόνο τοποθεσία**.
- **Πως;** Κατά τη φάση της **σχεδίασης και κανονικοποίησης**.

Επιπρόσθετα Πλεονεκτήματα Χρήσης ΒΔ (συν.)

6) Περιορισμός της Μη-Εξουσιοδοτημένης Προσπέλασης

- Δυνατότητα δημιουργίας ρόλων, ομάδων, κτλ. όπως σε συμβατικά Λειτουργικά Συστήματα.

7) Παροχή Εφεδρικών Αντιγράφων (Backup) και Μηχανισμών Ανάκαμψης (Recovery)

- Π.χ., Εάν έχουμε διακοπή ρεύματος (χωρίς UPS) δεν θα βρεθεί σε ασυνεπή κατάσταση η βάση δεδομένων.

8) Υποστήριξη Δομών Αποθήκευσης για Αποτελεσματική Επεξεργασία Επερωτήσεων

- Π.χ., Ευρετήρια Κατακερματισμού, Δενδρικά Ευρετήρια, Χωρικά Ευρετήρια και πολλά άλλα.

Επιπρόσθετα Πλεονεκτήματα Χρήσης ΒΔ (συν.)

9) Δυνατότητα Επιβολής Κανόνων Ακεραιότητας

- Π.χ., Επιβολή κανόνα: «Σε κανένα μάθημα δεν μπορούν να εγγραφούν πάνω από 99 φοιτητές»
- Παράδειγμα σε SQL-DDL*:

```
CREATE TABLE COURSE
```

```
( CourseID INT NOT NULL DEFAULT 1
```

```
Positions INT NOT NULL CHECK (Pos<100)....)
```

- Η δηλωτική αυτή γλώσσα θα μελετηθεί αργότερα στο μάθημα

10) Δυνατότητα Εκτέλεσης Σκανδαλών (Triggers)

Π.χ., Όποτε αλλάξει κάτι στον λογαριασμό του χρήστη στείλε του SMS!

```
CREATE TRIGGER change AFTER INSERT OR DELETE OR UPDATE ON  
CustomerAccount
```

```
FOR EACH ROW EXECUTE PROCEDURE sendSMS ();
```

Επιπρόσθετα Πλεονεκτήματα Χρήσης ΒΔ (συν.)

11) Δυνατότητα Επιβολής Κανόνων στην Αναπαράσταση Δεδομένων σε ένα οργανισμό:

- Π.χ., Σχεδιάζοντας και Υλοποιώντας τη βάση UNIVERSITY προτυποποιείται ότι το STUDENT.Name είναι μέχρι 30 χαρακτήρες.

12) Μειωμένος χρόνος ανάπτυξης εφαρμογών

- Υπολογίζεται ότι ο χρόνος ανάπτυξης εφαρμογών με βάσεις μειώνεται από **75-84%** του χρόνου που απαιτείται.

13) Ευελιξία

- Εύκολη μετάβαση σε άλλη κωδικοποίηση βάσης (με εργαλεία EXPORT).
- Όχι τόσο απλό εάν είναι σε ειδική κωδικοποίηση

Μοντέλα Δεδομένων

- Ένα **Μοντέλο Δεδομένων (Data Model)** είναι ένα **σύνολο εννοιών** που μπορούν να χρησιμοποιηθούν για την περιγραφή της **δομής** ενός **database**
- Περιλαμβάνουν **έννοιες Δομής**
 - **Οντότητες (Elements)** και τους **τύπους δεδομένων τους**
 - *Ομάδες Οντοτήτων (Εγγραφές, Πίνακες, κτλ)*
 - **Συσχετίσεις (Relationships)** μεταξύ των Οντοτήτων.
 - **Περιορισμούς (Constraints)** τα οποία περιλαμβάνουν κάποιους κανόνες οι οποίοι πρέπει να τηρούνται πάντα για να είναι η ΒΔ σε μια συνεπή κατάσταση.
- Επίσης περιλαμβάνουν **έννοιες Τελεστών**
 - Τελεστές **Ανάκτησης** και **Ενημέρωσης** Δεδομένων
 - Χωρίζεται σε **βασικούς τελεστές** (insert, delete, update) και τελεστών χρηστών (π.χ., compute_student_gra, update_inventory)

Μοντέλα Δεδομένων (συν.)

- **Υψηλού Επιπέδου Μοντέλα**, (Εννοιολογικό)
(High-level or Conceptual)
 - Παρέχει έννοιες κοντά στον τρόπο που **πολλοί χρήστες καταλαβαίνουν** τα διάφορα δεδομένα
 - Π.χ., **Entity-Relationship Model**
- **Ενδιάμεσου Επίπεδου Μοντέλα** (Αναπαραστατικό)
(Representational)
 - Παρέχει έννοιες που είναι **μεν κατανοητές** από τους **χρήστες** αλλά **όχι πολύ απομακρυσμένες** από το τρόπο αποθήκευσης
 - Π.χ., **Relational Model** and DB Schemas
- **Χαμηλού Επιπέδου Μοντέλα** (Low-level or Physical)
 - Παρέχει έννοιες που περιγράφουν τις λεπτομέρειες του πως τα δεδομένα είναι αποθηκευμένα στη **δευτερεύουσα μνήμη**
 - Π.χ., Specific Storage Model

Παραδείγματα Μοντέλων

- Παράδειγμα Εννοιολογικού Μοντέλου

- **Οντότητα (Entity)**

- Αντιπροσωπεύει πραγματικό αντικείμενο ή έννοια

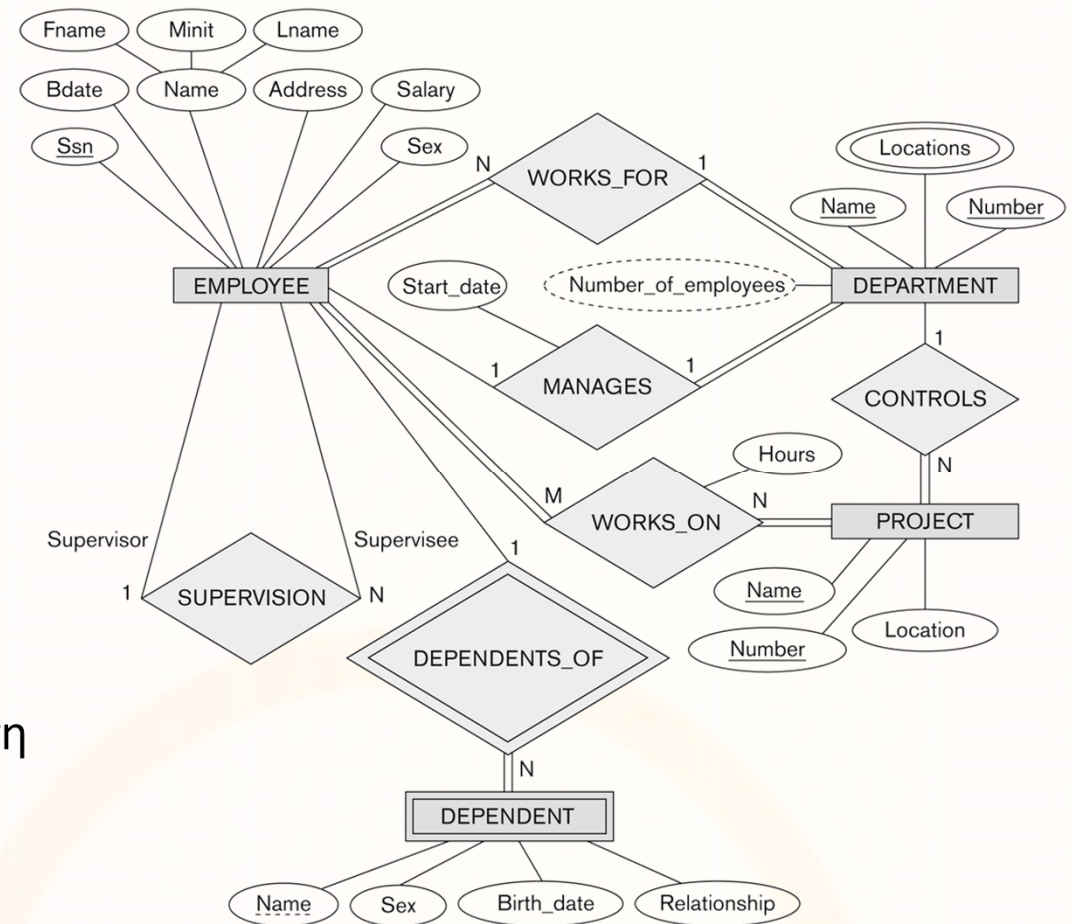
- **Χαρακτηριστικό (Attribute)**

- Αντιπροσωπεύει ιδιότητα κάποιου αντικειμένου

- **Σχέση (Relationship)**

- Αντιπροσωπεύει μία συσχέτιση μεταξύ οντοτήτων

- **Entity-Relationship model**



Παραδείγματα Μοντέλων (συν.)

- Παράδειγμα Αναπαραστατικού Μοντέλου

- Πολύ κοντά στο φυσικό μοντέλο

- **Σχέση**

- Αντιπροσωπεύει ένα σύνολο από συσχετιζόμενα χαρακτηριστικά
- Αντιπροσωπεύει μία συσχέτιση μεταξύ δύο συνόλων

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

- **Relational Model**

Σχήματα και Στιγμιότυπα

- **Σχήμα Βάσης (Database Schema)**
 - Η περιγραφή μιας βάσης.
 - Περιλαμβάνει περιγραφές της **δομής, τύπων δεδομένων**, και **περιορισμούς**.
- **Διάγραμμα Σχήματος:**
 - Ένας γραφικός τρόπος αναπαράστασης των πληροφοριών ενός σχήματος.
- **Κατάσταση Βάσης (Database State):**
 - Η πραγματική πληροφορία που αποθηκεύεται σε μια **δεδομένη στιγμή**
 - Ονομάζεται επίσης **Στιγμιότυπο ΒΔ (DB instance, occurrence or snapshot)**.
- Το **Σχήμα** μιας βάσης **ΔΕΝ** αλλάζει συχνά
- Η **Κατάσταση** μια βάσης **αλλάζει** συχνά.

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Λεξικό Βάσης Δεδομένων

- Το λεξικό της βάσης δεδομένων περιγράφει με λεπτομέρεια το φυσικό σχήμα των δεδομένων.

setName	fldName	fldType	fldMin	fldMax	Dec	fldTitle	Order1	Group1	TwoColum	Default	Pickl
a_users	username	C	3	20		User Name	2	all	<input type="checkbox"/>		
a_users	password	C	4	15		Password	4	all	<input type="checkbox"/>		
a_users	password2	C		15		Verify Password	4.5	all	<input type="checkbox"/>		
a_users	Zodiac	L					5	all	<input type="checkbox"/>	--	Aries, Lib
a_users	fullname	C	2	40		Name	7	all	<input type="checkbox"/>		
a_users	phone1	C	0	40		Phone	9	read	<input type="checkbox"/>		
a_users	email	C	5	80		E-Mail Address	11	all	<input type="checkbox"/>		
a_users	ccn	C	0	20		Credit Card Number	13	all	<input type="checkbox"/>		
a_users	othercontact	C	0	60		Other Contact Info	17	all	<input type="checkbox"/>	test default	
a_users	whenAdded	D					19	none	<input type="checkbox"/>		
a_users	State	L					23	all	<input type="checkbox"/>	--	
a_users	BirthMonth	L				Birth Month	30	all	<input checked="" type="checkbox"/>	Mar.	Jan.,Feb.
a_users	BirthDay	L				Birth Day	31	all	<input checked="" type="checkbox"/>		1,2,3,4,5
a_users	BirthYear	N		2999	0	Birth Year	33	all	<input checked="" type="checkbox"/>		
a_users	BirthTime	C	0	15		Birth Time	35	all	<input checked="" type="checkbox"/>		
a_users	userExpires	D	1/1/1990	12/31/2999		End Date	37	all	<input type="checkbox"/>		
a_users	Notes	M		1000			40	all	<input type="checkbox"/>		
a_users	subscribed	Y				Wants Promo Email	43	all	<input checked="" type="checkbox"/>		
a_users	discount	Y				Discounted	48	all	<input checked="" type="checkbox"/>	True	
sample2	orderID	n			0	Order ID	1	R	<input checked="" type="checkbox"/>		
sample2	Descript	c	2	30		Description	2	A	<input checked="" type="checkbox"/>		
sample2	price1	n	0.00	9000	2	Amount	3	A	<input checked="" type="checkbox"/>		
sample2	Tax	n		9000	2		4	A	<input checked="" type="checkbox"/>		
sample2	price2	n			2	Alternate Price	6	R	<input checked="" type="checkbox"/>		
sample2	nullMe	c		250		Intro	7	A	<input checked="" type="checkbox"/>		
sample2	nonNull	c		99		Mfr.	8	A	<input checked="" type="checkbox"/>		

Record: 6 of 27

Γλώσσες Βάσεων Δεδομένων

- Όταν ολοκληρωθεί η φάση της **εννοιολογικής μοντελοποίησης** των απαιτήσεων του χρήστη τότε ένας DBA ή DB Designer προχωρεί στην υλοποίηση της βάσης δεδομένων με τα ακόλουθα:

A) Γλώσσα Ορισμού (Δομής) Δεδομένων (DDL)

- Χρησιμοποιείται από τον DBA και τον DB Σχεδιαστή για τον ορισμό του **Εννοιολογικού Σχήματος (Αναπαραστατικού Μοντέλου)** μιας βάσης.

B) Γλώσσα Χειρισμού (Επεξεργασίας) Δεδομένων (DML)

- Χρησιμοποιείται στον ορισμό **ανακτήσεων (retrievals)** και **ενημερώσεων (updates)**

Γ) Γλώσσα Ορισμού Ελέγχων (DCL)

- Χρησιμοποιείται από τον DBA για τον ορισμό ελέγχων ασφαλείας

Δ) Γλώσσα Χειρισμού Εντολών (TCL)

- Χρησιμοποιείται στον ορισμό **εντολών (transactions)**

Γλώσσα Ορισμού (Δομής) Δεδομένων (DDL)

- Παραδείγματα Εντολών: CREATE, DROP, ALTER
- Παράδειγμα σε SQL-DDL*:
 - **CREATE TABLE** products (
 product_no integer,
 name text,
 price numeric);
 - *Δημιουργεί ένα πίνακα products με 3 πεδία (γνωρίσματα)*
- Η DDL χρησιμοποιείται επίσης για τον ορισμό των **εσωτερικών** (π.χ., indexes) και **εξωτερικών σχημάτων** (π.χ., views).

Π.χ., **CREATE VIEW** expensive_products AS

 SELECT name, price

 FROM products

 WHERE price>100;

* *Δημιουργεί ένα νοητό πίνακα που περιλαμβάνει μόνο τα ακριβά προϊόντα*

Γλώσσα Χειρισμού Δεδομένων (DML):

- Παραδείγματα Εντολών: SELECT, UPDATE, INSERT, DELETE
- Παράδειγμα SQL-DML*:
 - SELECT * FROM products;
 - Επιστρέφει όλα τα προϊόντα στον πίνακα products;
- Οι εντολές DML μπορούν να ενσωματωθούν σε μια **γλώσσα προγραμματισμού** (π.χ., C, C++, C#, Java, κτλ.).
- Εναλλακτικά, μπορούμε να εκτελέσουμε τέτοιες εντολές απευθείας από τη γραμμή εντολών κάποιου **κελύφους SQL** (psql/PostgreSQL, *SQL*Plus*/Oracle, *SQLCMD/SQLServer* κτλ) ή ακόμα και μέσω γραφικού περιβάλλοντος διαπροσωπείας

Γλώσσα Ορισμού Ελέγχων (DCL)

- Παραδείγματα Εντολών: GRANT, DENY, REVOKE

- Παράδειγμα σε SQL-DCL*:

```
DENY    DELETE
ON      Employees
TO      Andreas;
```

- *Δεν επιτρέπει την διαγραφή δεδομένων από τον πίνακα Employees στο χρήστη Andreas*

Γλώσσα Χειρισμού Εντολών (TCL)

- Παραδείγματα Εντολών: COMMIT, ROLLBACK
- Ομαδοποίηση εντολών και εκτέλεση τους σαν μία εντολή. Σε περίπτωση αποτυχίας, μπορούν να αναιρεθούν όλες οι εντολές.

- Παράδειγμα σε SQL-TCL*:

BEGIN TRANSACTION

```
INSERT INTO STUDENTS(ID, NAME) VALUES(13, Andreas)
```

```
INSERT INTO STUDENT_INTERESTS(ID, INTEREST) VALUES(13, 'Football')
```

```
INSERT INTO STUDENT_INTERESTS(ID, INTEREST) VALUES(13, 'Rugby')
```

COMMIT TRANSACTION

```
IF @@ERROR<>0
```

ROLLBACK TRANSACTION

Πότε να μην χρησιμοποιούμε Βάσεις;

A) Υπάρχει συνήθως ψηλό κόστος αγοράς, π.χ.,

- Oracle Standard Edition (per CPU): **5,900\$**
- Oracle Enterprise Edition (per CPU): **47,500\$**
- IBM DB2 v9.7 Enterprise: **25,000\$**
- SQL Server 2008 Enterprise: **25,000\$**
- Τα πιο πάνω ΔΕΝ περιλαμβάνουν κόστος αγοράς υλικού (server), λειτουργικού συστήματος, training, κτλ.!
- Όταν ο προϋπολογισμός ενός project είναι περιορισμός τότε λαμβάνεται σημαντικά υπόψη μια τέτοια παράμετρος.
- Ακόμη και εάν χρησιμοποιηθεί DBMS **ανοικτού πηγαίου κώδικα** (π.χ., SQL Server, MySQL, PostgreSQL, κτλ.), οι οποίες είναι **δωρεάν**, τότε υπάρχουν και πάλι λόγοι για τους οποίους δεν θέλουμε να χρησιμοποιήσουμε τέτοιες βάσεις

Πότε να μην χρησιμοποιούμε Βάσεις; (συν.)

- Β)** Εάν η βάση δεδομένων είναι **απλή**, καλά ορισμένη και δεν πρόκειται να αλλάξει εύκολα
- Π.χ., ένα απλό κατάλογο τηλεφώνων ενός γραφείου.
- Γ)** Εάν η πληροφορία είναι **πάρα πολύ μεγάλη**
- π.χ., το Google ΔΕΝ αποθηκεύει το ευρετήριο σε μια DBMS. **Γιατί;**
 - Αυτό βέβαια δεν είναι κανόνας. Λέγεται ότι το Facebook χρησιμοποιεί την MySQL:
 - Το Facebook έχει πάνω από 1.5 petabytes (10^{15}) σε φωτογραφίες (περίπου 10 δισεκατομμύρια φωτογραφίες!)
- Δ)** Εάν δεν υπάρχει το **προσωπικό («DBA»)** για παροχή ασφάλειας, ανάκαμψης, αντιγράφων, κτλ.
- Ε)** Εάν δεν υπάρχει πολλή **ταυτόχρονη πρόσβαση** στα δεδομένα, εάν δεν μπορεί εύκολα να αναπαρασταθεί κάποια πληροφορία σε μια βάση.