



ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Διάλεξη 2^η: ΕΙΣΑΓΩΓΗ - UML - SQL



Unified Modeling Language (UML)

- Τι είναι
- Κλάσεις (Classes)
- Ορατότητα (Visibility)
- Αφηρημένες κλάσεις (Abstract classes)
- Σχέσεις (Associations)
 - Απλές σχέσεις (1 to 1, 1 to many, many to many)
 - Συνάθροιση (Aggregation)
 - Σύνθεση (Composition)
 - Γενίκευση/εξειδίκευση (Generalization/specialization)



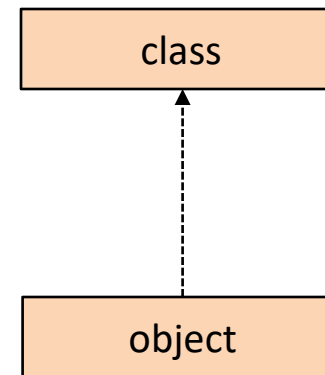
UML: Τι είναι

- Γλώσσα περιγραφής/απεικόνισης αντικειμενοστραφών μοντέλων
 - «Σχημάτων» Βάσεων Δεδομένων (DB schemata)
 - Κλάσεων/αντικειμένων στον αντικειμενοστραφή προγραμματισμό
 - ...
- ISO/IEC standard 19501:2005



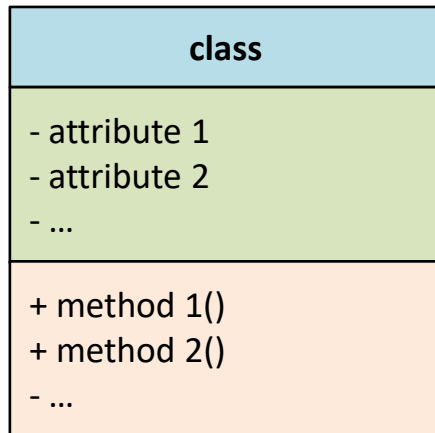
UML: classes

- Κλάση: «Φόρμα» για όμοια αλλά διακριτά αντικείμενα
- Τα μέλη μιας κλάσης μοιράζονται κοινές ιδιότητες (**attributes**) και συμπεριφορά (**methods**)
- Τα μέλη βρίσκονται προς την κλάση σε σχέση **“instance of”**





UML: visibility



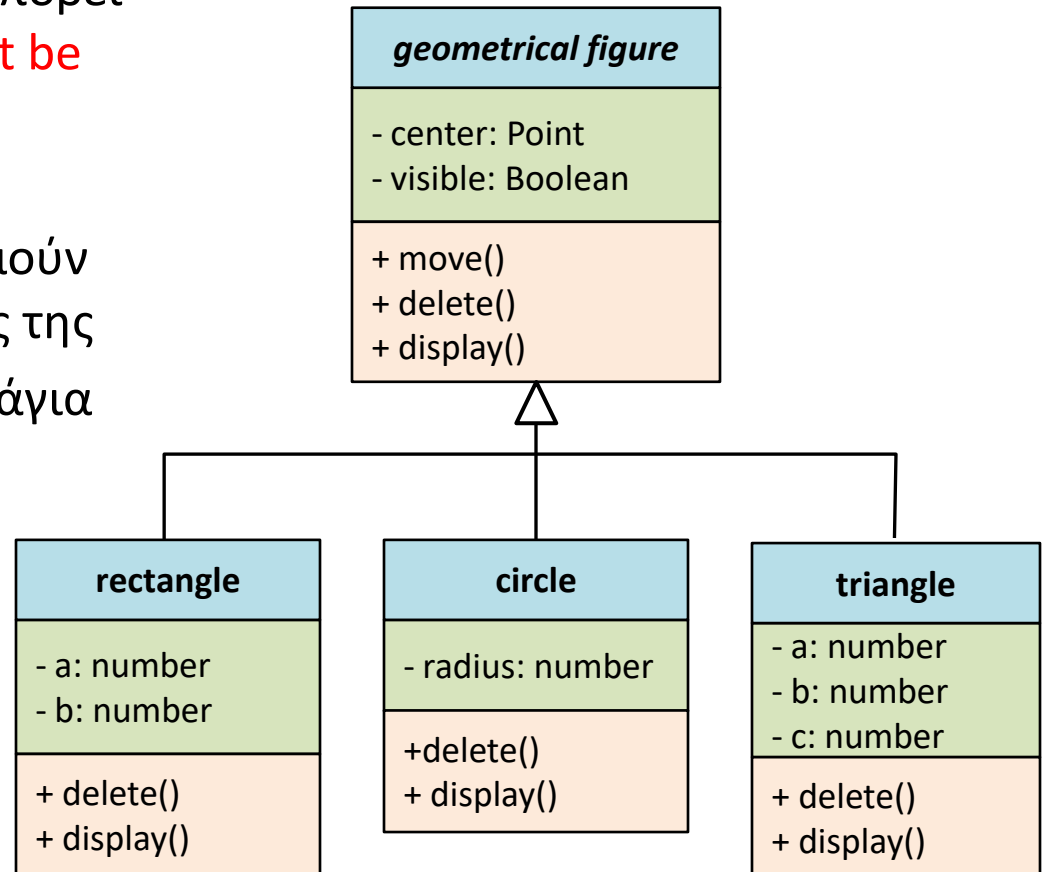
- Ορατά μέσα στην κλάση
 - **Private**
 - σύμβολο: -
- Ορατά έξω από την κλάση (από άλλες κλάσεις)
 - **Public**
 - σύμβολο: +
- Τα private attributes γράφονται ή διαβάζονται μέσω public methods

Κλασική περίπτωση «ενθυλάκωσης»
(**encapsulation**)



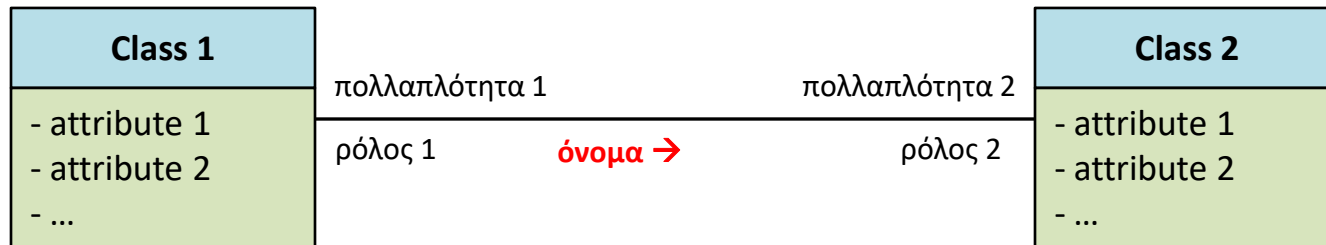
UML: abstract classes

- Μια αφηρημένη κλάση δεν μπορεί να υλοποιηθεί αμέσως (“**can’t be instantiated**”)
- Αποτελεί γενίκευση άλλων κλάσεων, οι οποίες κληρονομούν τις ιδιότητες και τις μεθόδους της
- Το όνομά της γράφεται σε πλάγια γραφή

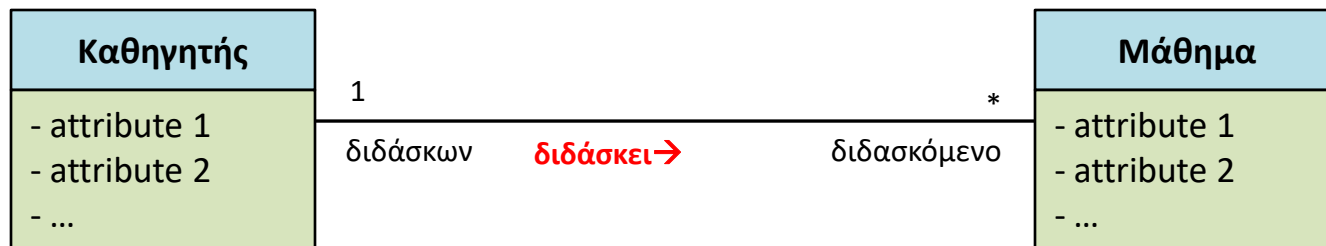




UML: simple relations

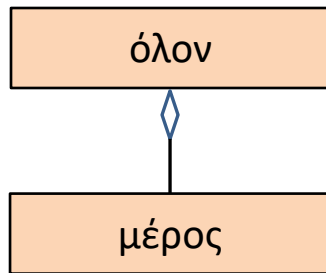


Παράδειγμα: Σχέση μεταξύ καθηγητή και διδασκόμενου μαθήματος



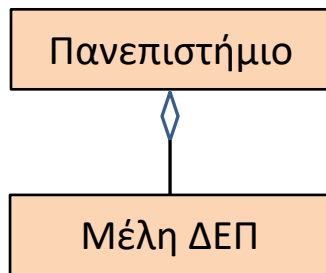


UML: aggregation



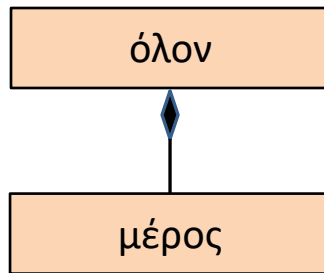
- Πρόκειται για την ειδική εκείνη σχέση μέρους-όλου
 - ολον: “consists of ...”
 - μέρος: “is part of ...”

Παράδειγμα: Πανεπιστήμιο και μέλη του

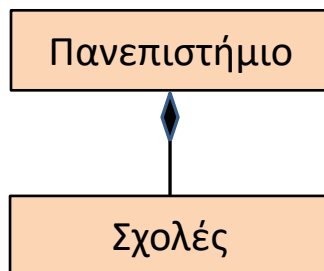




UML: composition



Παράδειγμα: Πανεπιστήμιο και Σχολές του

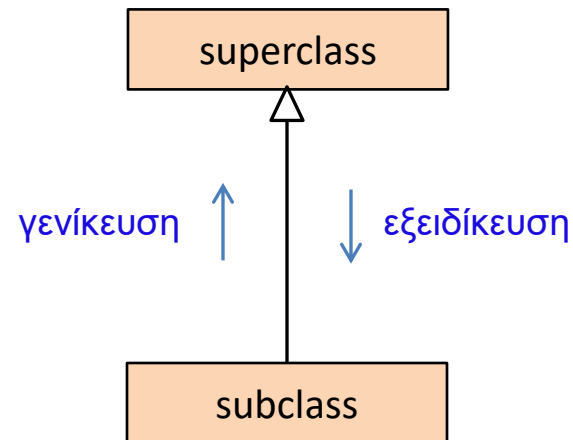


- Η ειδική εκείνη σχέση aggregation όπου όλα τα «μέρη» είναι απαραίτητα για τη σύνθεση του «όλου»
- Η πολλαπλότητα της κλάσης του «όλου» είναι ακριβώς 1
- Μια σχέση δεν είναι composition αν κάποιο «μέρος» συνδέεται με περισσότερα «όλα»
- Η διαγραφή του «όλου» συνεπάγεται αναγκαστική διαγραφή όλων των «μερών»

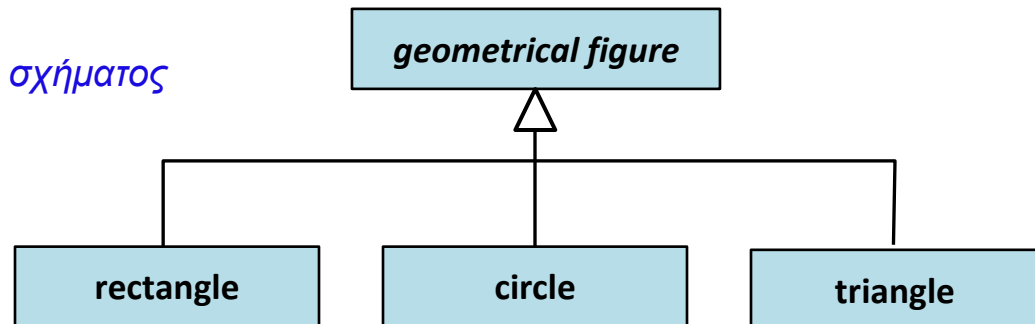


UML: Generalization/specialization

- Η γενική κλάση ονομάζεται **υπερκλάση (superclass)** και η ειδική, **υποκλάση (subclass)**.
- Η υποκλάση εξειδικεύει την υπερκλάση με πρόσθετες ιδιότητες.
- Μια υπερκλάση μπορεί να έχει πολλές διαφορετικές υποκλάσεις.



Παράδειγμα: Ιεραρχία γεωμετρικού σχήματος





Structured Query Language (SQL)

- Τι είναι
- Εντολές Data Definition Language (DDL)
- Εντολές Data Manipulation Language (DML)
- Εντολές Data Control Language (DCL)
- Εντολές Transaction Control Language (TCL)
- Οι τελεστές JOIN



SQL: Τι είναι

- Γλώσσα για τη διαχείριση δεδομένων και υποβολή ερωτημάτων σε σχεσιακές βάσεις δεδομένων:
 - Δημιουργία/αλλαγή «σχήματος» (schema creation)
 - Εισαγωγή/αλλαγή δεδομένων
 - Αναζήτηση/επεξεργασία δεδομένων με διατύπωση ερωτημάτων
- Δηλωτική (**declarative**) κατά βάση γλώσσα, όμως και με στοιχεία προγραμματισμού (**procedural**)
- Πρότυπο ANSI (1986) και ISO (**ISO/IEC 9075-1:2008**)



SQL: DDL

- Data Definition Language (DDL): εντολές που χρησιμοποιούνται για τον ορισμό της δομής της ΒΔ (schema). Μερικά παραδείγματα:
 - **CREATE** - για τη δημιουργία αντικειμένων στη βάση
 - **ALTER** - για την αλλαγή της δομής
 - **DROP** - για τη διαγραφή αντικειμένων από τη βάση
 - **TRUNCATE** - διαγράφει όλες τις εγγραφές από έναν πίνακα και ελευθερώνει όλο το χώρο που είχε διατεθεί για αυτές
 - **COMMENT** - προσθέτει σχόλια



SQL: DML

- Data Manipulation Language (DML): Εντολές για τη διαχείριση των δεδομένων των αντικειμένων της δομής. Μερικά παραδείγματα:
 - **SELECT** - ανακτά δεδομένα από τη βάση
 - **INSERT** - εισάγει δεδομένα σε κάποιον πίνακα
 - **UPDATE** - επικαιροποιεί υπάρχοντα δεδομένα σε πίνακα
 - **DELETE** - διαγράφει όλες τις εγγραφές ενός πίνακα, αλλά ο χώρος παραμένει δεσμευμένος



SQL: DCL

- Data Control Language (DCL): Παραδείγματα:
 - **GRANT** - δίνει στους χρήστες δικαιώματα πρόσβασης στη ΒΔ
 - **REVOKE** – αποσύρει τα δικαιώματα πρόσβασης που δόθηκαν με εντολή GRANT



SQL: TCL

- **Transaction Control Language (TCL)**: Εντολές που χρησιμοποιούνται για τη διαχείριση (μονιμοποίηση, αναίρεση) αλλαγών που έγιναν με εντολές DML. Επιτρέπει την ομαδοποίηση εντολών προς εκτέλεση:
 - **COMMIT** - αποθηκεύει ό, τι έχει γίνει
 - **SAVEPOINT** - σημειώνει ένα σημείο «συναλλαγής», όπου μπορεί αργότερα να επανέλθει η ροή εντολών με ROLLBACK
 - **ROLLBACK** - αναίρεση των αλλαγών που έγιναν από το τελευταίο COMMIT και έπειτα



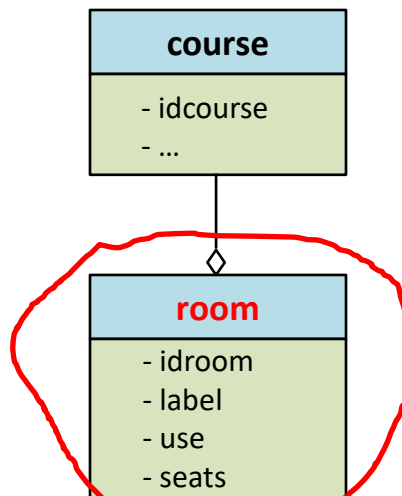
Σχεδιασμός ΒΔ με αφετηρία ένα μοντέλο UML

- Υπάρχει ένα σύνολο πρακτικών οδηγιών
- Γενικά, ακολουθούμε τα εξής βήματα:
 1. Κάθε κλάση, που δεν συμμετέχει σε κάποια ιεραρχία με κληρονομικότητα, αντιστοιχίζεται σε έναν πίνακα
 2. Κάθε σχέση, πλην της κληρονομικότητας, αντιστοιχίζεται σε έναν πίνακα
 3. Μερικοί πίνακες μπορούν στη συνέχεια να συμπτυχθούν
 4. Οι ιεραρχίες κληρονομικότητας μετατρέπονται χωριστά



Αντιστοίχιση κλάσεων

- Κάθε κλάση αντιστοιχίζεται σε έναν πίνακα
 - Όνομα του πίνακα: το όνομα της κλάσης
 - Στήλες: οι ιδιότητες της κλάσης (attributes στο διάγραμμα UML)
 - Οι τύποι δεδομένων μπορεί να χρειαστεί να προσαρμοστούν στους διαθέσιμους τύπους της SQL [7]



```
CREATE TYPE room_use AS ENUM ('office',  
'class');
```

```
CREATE TABLE room (  
    idroom INTEGER NOT NULL PRIMARY KEY,  
    label VARCHAR(50) DEFAULT NULL,  
    use room_use,  
    seats INTEGER  
);
```

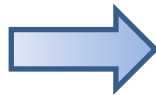
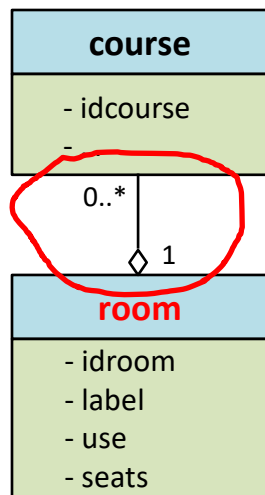
δικός μας τύπος



Αντιστοίχιση σχέσεων

- Κάθε σχέση μπορεί να αντιστοιχιστεί σε έναν πίνακα
 - **Όνομα του πίνακα**: το όνομα της σχέσης, ή, αν υφίσταται, το όνομα της σχετιζόμενης κλάσης συσχέτισης (οποιοδήποτε νέο όνομα)
 - **Στήλες**: κύριες ιδιότητες των σχετιζόμενων κλάσεων, καθώς και της σχετιζόμενης κλάσης συσχέτισης (αν υπάρχει). Απαραίτητα, τα κύρια κλειδιά των πινάκων των σχετιζόμενων κλάσεων.

Παράδειγμα: Η σχέση 1:n μεταξύ *room* και *course*



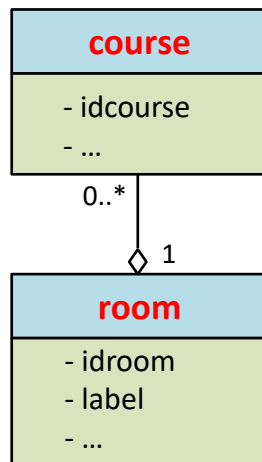
```
CREATE TABLE booking (
    roomid INTEGER NOT NULL,
    courseid VARCHAR(50),
    PRIMARY KEY (roomid,courseid),
    FOREIGN KEY (roomid) REFERENCES room(idroom),
    FOREIGN KEY (courseid) REFERENCES course(idcourse)
);
```



Συγχώνευση πινάκων

- Οι πίνακες που προκύπτουν από σχέσεις **1:1** ή **1:n** μπορούν να συγχωνευτούν με πίνακες των σχετιζόμενων κλάσεων
 - Κλάσεις A,B, σε σχέση 1:n Το κλειδί του πίνακα A μπορεί να προστεθεί, ως FOREIGN KEY, στον B (την πλευρά πολλαπλότητας n) και ο πίνακας της σχέσης να καταργηθεί
 - Στήλες: κύριες ιδιότητες των σχετιζόμενων κλάσεων, καθώς και της σχετιζόμενης κλάσης συσχέτισης (αν υπάρχει)
 - Στη σχέση 1:1, το FOREIGN KEY μπορεί να προστεθεί στον έναν ή τον άλλον πίνακα

Παράδειγμα: Η σχέση **1:n** μεταξύ **room** και **course**



Κατάργηση του πίνακα **booking**. Κλειδί του πίνακα **room** για τον **course**

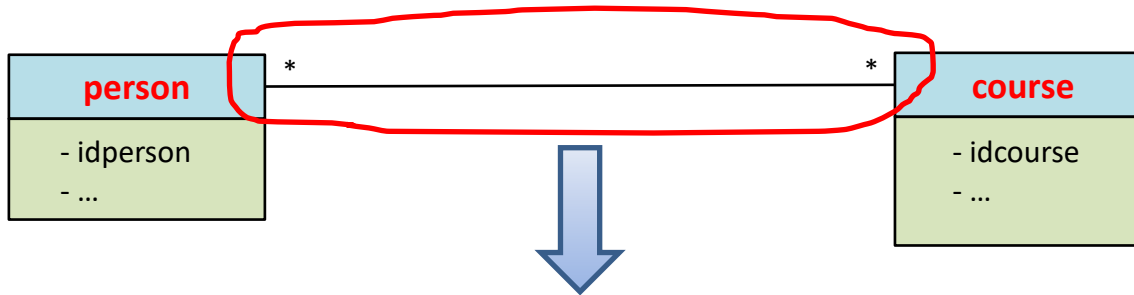
```
CREATE TABLE course (
  idcourse VARCHAR(50),
  roomid INTEGER NOT NULL,
  PRIMARY KEY (idcourse),
  FOREIGN KEY (roomid) REFERENCES room(idroom)
);
```



Συγχώνευση πινάκων (συνέχεια)

- Για τη σχέση **n:n** ο πίνακας θα πρέπει να διατηρηθεί!

Παράδειγμα: Η σχέση **n:n** μεταξύ **person** και **course**



```
CREATE TABLE personcourse (  
  personid  INTEGER NOT NULL,  
  courseid  INTEGER NOT NULL,  
  PRIMARY KEY (personid, courseid),  
  FOREIGN KEY (personid) REFERENCES person(idperson),  
  FOREIGN KEY (courseid) REFERENCES course(idcourse)  
);
```



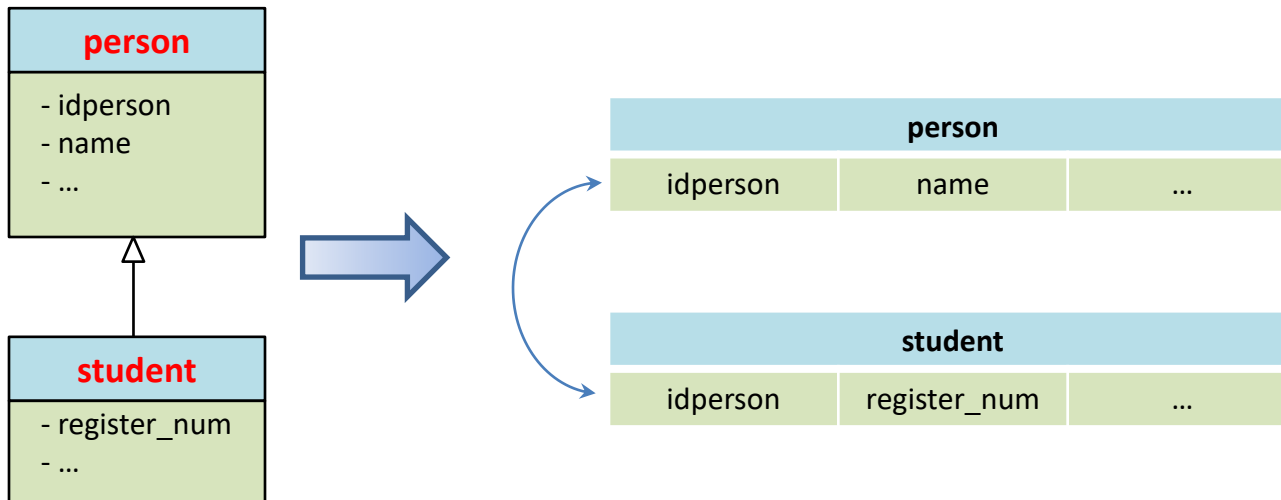
Σχέσεις ιεραρχίας (κληρονομικότητα)

- Σε ιεραρχίες υπερκλάσης-υποκλάσης, μόνο οι κλάσεις μετατρέπονται σε πίνακες, όχι οι σχέσεις
- Έχουμε τουλάχιστον δύο εναλλακτικές επιλογές:
 - Έναν πίνακα ανά κλάση της ιεραρχίας
 - Έναν και μόνο πίνακα για ολόκληρη την ιεραρχία
- Κριτήρια για την επιλογή:
 - Απαιτούμενος χώρος
 - Ευκολία στη διατύπωση ερωτημάτων



Σχέσεις ιεραρχίας (συνέχεια)

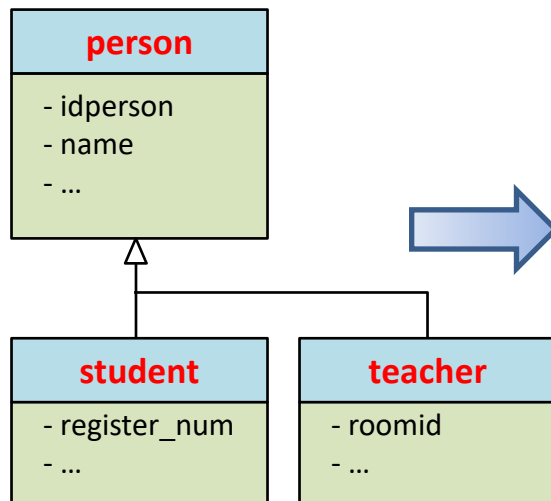
- **Επιλογή I:** Ένας πίνακας ανά κλάση της ιεραρχίας, με
 - **Όνομα πίνακα:** το όνομα της κλάσης
 - **Στήλες:**
 - τις ιδιότητες (attributes) της κλάσης στο διάγραμμα UML
 - Ιδιότητα-κλειδί της ρίζας της ιεραρχίας, που παραμένει κλειδί





Σχέσεις ιεραρχίας (συνέχεια)

- **Επιλογή II:** Ένας μόνον πίνακας για ολόκληρη την ιεραρχία.
 - **Όνομα πίνακα:** μπορεί να είναι το όνομα της μητρικής κλάσης (ρίζας)
 - **Στήλες:** όλες οι ιδιότητες των κλάσεων που εμφανίζονται στην ιεραρχία, με νέα ονόματα αν χρειάζεται. Ιδιότητες που δεν υπάρχουν σε κάποια υποκλάση παίρνουν τιμή **NULL**¹.



person				
idperson	name	register_num	roomID	...
52	ΜΗΤΡΟΥ	NULL	125	...
...
43	ΑΝΤΩΝΙΟΥ	2018005	NULL	...

¹ Γενικά, ο τύπος NULL χρησιμοποιείται στις περιπτώσεις: (α) άγνωστο, (β) μη διαθέσιμο, (γ) με εφαρμοσμένο



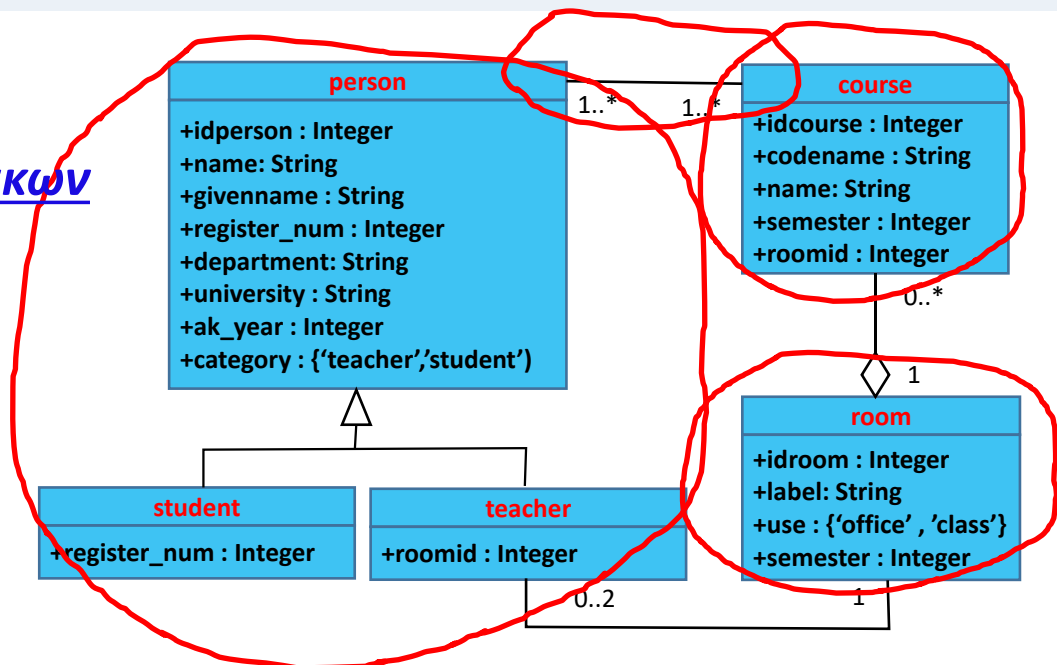
Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Ένα πιο προηγμένο μοντέλο

- Περιλαμβάνει όλες τις εμπλεκόμενες οντότητες στο ΔΠΜΣ
 - Καθηγητές, σπουδαστές
 - Μαθήματα
 - Χώρους (αίθουσες, γραφεία)
- ... και τις μεταξύ τους σχέσεις

Δημιουργία τεσσάρων πινάκων

- ✓ **person**
- ✓ **course**
- ✓ **room**
- ✓ **personcourse**





Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Πίνακας person

```
CREATE TYPE example2.person_category
AS ENUM('student', 'teacher')
CREATE TABLE example2.person (
  idperson INTEGER NOT NULL PRIMARY KEY,
  name VARCHAR(50) DEFAULT NULL,
  givenname VARCHAR(50) DEFAULT NULL,
  register_num INTEGER DEFAULT NULL,
  department VARCHAR(50) DEFAULT NULL,
  university VARCHAR(50) DEFAULT NULL,
  ak_year INTEGER DEFAULT 2020,
  category person_category DEFAULT 'student'
);
```

Έχοντας τον πίνακα student, ενημερώνουμε τον person ως εξής:

```
INSERT INTO example2.person (idperson, name, givenname, department, university, ak_year)
SELECT idstudent, name, givenname, department, university, ak_year
FROM example1.student
-- WHERE ak_year=2022; -- αν θέλουμε μόνο τους νέους σπουδαστές
```



idperson [PK]	serial	name character varying(50)	givenname character varying(50)	register_ integer	department character varying(50)	university character varying(50)	ak_year integer	category example2.person_catego
202201		ΚΩΝΣΤΑΝΤΙΝΟΣ	ΑΓΗΣΤΟΣ		ΑΤΜ	ΑΠΘ	2022	student
202202		ΑΝΔΡΟΝΟΠΟΥΛΟΥ	ΣΟΦΙΑ		ΑΤΜ-ΜΓ	ΕΜΠ	2022	student
202203		ΒΑΣΙΛΕΙΑΔΗΣ	ΣΑΒΒΑΣ		ΜΗΧ. ΧΩΡΟΤΑΞΙΑΣ ΚΑΙ ΑΝ	ΑΠΘ	2022	student
202204		ΒΛΑΧΟΥ	ΑΡΙΣΤΕΑ		ΑΤΜ-ΜΓ	ΕΜΠ	2022	student
202205		ΔΕΜΕΡΤΖΗ	ΓΕΩΡΓΙΑ ΙΣΙΣ		ΓΕΩΓΡΑΦΙΑΣ	ΧΑΡΟΚΟΠΕΙΟ. ΠΑΝ.	2022	student
202206		ΔΗΜΗΤΡΗ	ΙΩΑΝΝΑ-ΕΡΑΣΜΙΑ		ΓΕΩΓΡΑΦΙΑΣ	ΧΑΡΟΚΟΠΕΙΟ. ΠΑΝ.	2022	student
202207		ΛΑΒΙΔΑΣ	ΑΛΕΞΑΝΔΡΟΣ-ΔΗΜΗ		ΑΤΜ-ΜΓ	ΕΜΠ	2022	student



Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Πίνακες course και personcourse

```
CREATE TABLE example2.course (  
  idcourse INTEGER NOT NULL PRIMARY KEY,  
  code_name VARCHAR(15),  
  name VARCHAR(80),  
  semester INTEGER NOT NULL,  
  roomid INTEGER NOT NULL  
);
```

	idcourse [PK] integer	name character varying(80)	semester integer	roomid integer
1	1	ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	2	1
2	2	ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΣΤΗ ΓΠ	1	2
*				

```
CREATE TABLE example2.personcourse (  
  personid INTEGER NOT NULL,  
  courseid INTEGER NOT NULL,  
  PRIMARY KEY (personid, courseid)  
);
```

personid [PK] integer	courseid [PK] integer
202003	2
202004	1
202004	2
202006	2
202007	2
202008	2

και τα δύο μαθήματα

μόνο το 2^ο μάθημα

```
ALTER TABLE example2.personcourse  
  ADD CONSTRAINT FK_personcourse_person  
  FOREIGN KEY (personid)  
  REFERENCES person (idperson)  
  ON DELETE CASCADE;
```

```
ALTER TABLE example2.personcourse  
  ADD CONSTRAINT FK_personcourse_course  
  FOREIGN KEY (courseid)  
  REFERENCES course (idcourse)  
  ON DELETE CASCADE;
```

καθηγητές

ΟΝΟΜΑ character varying(50)	ΕΠΩΝΥΜΗΜΟ character varying(50)	ΠΑΝΕΠ character vari
ΒΑΣΙΛΕΙΟΣ	ΒΕΣΚΟΥΚΗΣ	ΕΜΠ
ΑΝΑΣΤΑΣΙΟΣ	ΖΑΦΕΙΡΟΠΟΥΛΟΣ	ΕΜΠ
ΙΩΑΝΝΗΣ	ΘΕΟΔΩΡΙΔΗΣ	ΠΑΠΕΙ
ΜΑΡΤΑΡΙΤΑ	ΚΟΚΛΑ	ΕΜΠ
ΝΙΚΟΛΑΟΣ	ΜΗΤΡΟΥ	ΕΜΠ



Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Πίνακας room

```
CREATE TYPE example2.room_use AS ENUM ('office', 'class');

CREATE TABLE example2.room (
  idroom INTEGER NOT NULL PRIMARY KEY,
  label VARCHAR(50) DEFAULT NULL,
  use example2.room_use,
  -- use ENUM('office', 'class'), -- for oracle
  seats INTEGER
);
INSERT INTO example2.room VALUES (1, 'ΣΑΤΜ-ΜΑ', 'class', 80);
...
```




Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Ερωτήματα

1. Να βρεθούν οι σπουδαστές (όνομα & επώνυμο) που δήλωσαν το μάθημα 'ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ' και το επώνυμό τους ξεκινάει με γράμμα μέχρι και το Κ

```
SELECT DISTINCT givenname AS ΟΝΟΜΑ, name AS ΕΠΩΝΥΜΟ FROM
example2.person,
(SELECT personid FROM example2.personcourse, example2.course
WHERE courseid=idcourse AND course.name=
'ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ') s1
WHERE s1.personid=person.idperson
AND person.category= 'student'
AND name < 'Λ'
ORDER BY name;
```



ΟΝΟΜΑ character varying(50)	ΕΠΩΝΥΜΟ character varying(50)
ΦΩΤΕΙΝΗ	ΒΑΚΑΛΟΠΟΥΛΟΥ
ΣΩΤΗΡΙΟΣ-ΜΑΡΙΟΣ	ΚΑΛΑΜΠΟΚΗΣ
ΑΝΑΣΤΑΣΙΟΣ	ΚΑΤΣΟΣ
ΠΕΤΡΟΣ	ΚΥΡΙΑΚΟΥ



Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Ερωτήματα (συνέχεια)

2. Να βρεθούν οι σπουδαστές οι οποίοι δήλωσαν 'ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ' και όχι 'ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΣΤΗ ΓΠ'

```
SELECT givenname AS ΟΝΟΜΑ, name AS ΕΠΩΝΥΜΟ FROM example2.person,  
      (SELECT personid FROM example2.personcourse, example2.course  
       WHERE courseid=idcourse AND course.name='ΧΩΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ') s1  
WHERE s1.personid=person.IDperson AND person.category= 'student'  
      AND s1.personid NOT IN  
      (SELECT personid FROM example2.personcourse, example2.course  
       WHERE courseid=idcourse AND course.name='ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΣΤΗ ΓΠ')  
ORDER BY name;
```



ΟΝΟΜΑ character varying(50)	ΕΠΩΝΥΜΟ character varying(50)
ΠΕΤΡΟΣ	ΚΥΡΙΑΚΟΥ
ΜΙΧΑΗΛ	ΠΡΩΤΟΓΕΡΟΣ
ΕΙΡΗΝΗ	ΤΣΕΚΟΥΡΑ
ΒΑΣΙΛΕΙΟΣ	ΨΗΡΟΥΚΗΣ



Παράδειγμα 2: Βάση Δεδομένων ΔΠΜΣ (συνέχεια)

Ασκηση 1α (για παράδοση)

Θα αναρτηθεί στην ιστοσελίδα του μαθήματος



Βιβλιογραφία

[1] Elmasri, Navathe, *Fundamentals of Database Systems*, 6th edition, Addison-Wesley, 2011, (μπορεί να βρεθεί και σε ηλεκτρονική μορφή)

[1.a] “ Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων, 7^η Έκδοση, ΔΙΑΥΛΟΣ Α.Ε. ΕΚΔΟΣΕΙΣ ΒΙΒΛΙΩΝ, 2016

[2] Βερούκιος, Β., & Βασιλακόπουλος, Μ. (2022). *Συστήματα Βάσεων Δεδομένων* [Προπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις.

<https://dx.doi.org/10.57713/kallipos-36>

[3] Clare Churcher, *Beginning Database Design – From Novice to Professional*, Apress 2007

[4] Hector Garcia-Molina, Jeffrey Ullman, *Database Systems: The Complete Book*, Pearson Education, 2009, 2013

[5] Jason Price, *Oracle Database 11g SQL*, Oracle Press, 2007

[6] Hans-Petter Halvorsen, *Structured Query Language*, 2017,

<https://www.halvorsen.blog/documents/tutorials/resources/Structured%20Query%20Language.pdf>

[7] Unified Modeling Language® (UML®) Resource Page, <http://www.uml.org/>

[8] SQL Data types (στην Postgres) <https://www.postgresql.org/docs/9.5/datatype.html>