

# Ιατρική Πληροφορική



Δρ. Π. ΑΣΒΕΣΤΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ Τ.Ε.Ε.

## Χρήσιμοι Σύνδεσμοι



- Σημειώσεις μαθήματος:  
<http://medisp.bme.teiath.gr/eclass/courses/TIO103/>  
<https://eclass.teiath.gr/courses/TIO100/>
- E-mail:  
pasv@teiath.gr

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εισαγωγή δεδομένων επιτυγχάνεται με χρήση της εντολής **INSERT INTO**
- Η εντολή αυτή παρέχει τη δυνατότητα για εισαγωγή μίας ή περισσότερων γραμμών σε έναν πίνακα
- Η σύνταξη της εντολής απαιτεί να οριστούν το όνομα του πίνακα στον οποίο θα γίνει η εισαγωγή δεδομένων, τα ονόματα των πεδίων στα οποία θα δοθούν τιμές και οι τιμές που θα δοθούν στα πεδία

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για παράδειγμα, έστω ο πίνακας για ασθενείς (astheneis)

Πεδίο	Τύπος	Ιδιότητες
kwdikos	INT UNSIGNED	AUTO_INCREMENT, PRIMARY KEY
onoma	VARCHAR(20)	NOT NULL
erwnymio	VARCHAR(30)	NOT NULL
hmniaGennhshs	DATE	NOT NULL

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εισαγωγή μιας γραμμής στον πίνακα γίνεται ως εξής:

```
INSERT INTO astheneis  
(ερwnυμο, ονομα, ημνιαGennhshs) VALUES  
('Doe', 'John', '1970-10-08');
```

- Το πεδίο `kwδικος` έχει την ιδιότητα `AUTO_INCREMENT`, το οποίο σημαίνει ότι λαμβάνει αυτόματα τιμή και δε χρειάζεται να οριστεί στην παραπάνω εντολή

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στην περίπτωση που δίνονται λιγότερες τιμές από τα πεδία, όπως στην ακόλουθη εντολή:

```
INSERT INTO astheneis  
(ερwnυμο, ονομα, ημνιαGennhshs) VALUES  
('Doe', 'John');
```

θα παράγει το μήνυμα σφάλματος:

```
Error Code: 1136. Column count doesn't match value count at  
row 1
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στην περίπτωση που δεν δίνεται τιμή σε κάποιο πεδίο που έχει την ιδιότητα NOT NULL και δεν έχει προεπιλεγμένη τιμή, όπως στην ακόλουθη εντολή :

```
INSERT INTO astheneis  
(epwnymo, onoma) VALUES  
('Doe', 'John');
```

η εισαγωγή θα πραγματοποιηθεί, αλλά θα εμφανιστεί η προειδοποίηση:

```
1 row(s) affected, 1 warning(s): 1364 Field 'hmniaGennshs'  
doesn't have a default value
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή **INSERT INTO** μπορεί να χρησιμοποιηθεί για τη μαζική εισαγωγή πολλών γραμμών σε έναν πίνακα:

```
INSERT INTO astheneis  
(epwnymo, onoma, hmniaGennshs) VALUES  
('Doe', 'John', '1970-10-08'),  
('Doe', 'Jane', '1973-08-23');
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η MySQL υποστηρίζει τη χρήση συναρτήσεων, οι οποίες μπορεί να έχουν κάποια ορίσματα και να επιστρέφουν κάποια τιμή
- Δύο συναρτήσεις που χρησιμοποιούνται συχνά κατά την εισαγωγή δεδομένων είναι:
  - NOW(): επιστρέφει την τρέχουσα ημερομηνία και ώρα
  - SHA1('some text'): λαμβάνει ως όρισμα μια συμβολοσειρά και επιστρέφει μία συμβολοσειρά 40 δεκαεξαδικών ψηφίων που αντιπροσωπεύει την κρυπτογράφηση του ορίσματος εισόδου με βάση τον αλγόριθμο SHA1

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Έστω ο πίνακας *users* με τους χρήστες μιας εφαρμογής

Πεδίο	Τύπος	Ιδιότητες
userId	MEDIUMINT UNSIGNED	AUTO_INCREMENT, PRIMARY KEY
firstName	VARCHAR(25)	NOT NULL
lastName	VARCHAR(40)	NOT NULL
email	VARCHAR(60)	NOT NULL
username	VARCHAR(15)	NOT NULL
pass	CHAR(40)	NOT NULL (ΚΡΥΠΤΟΓΡΑΦΗΜΕΝΟ ΚΑΤΑ SHA1)
registrationDate	DATETIME	NOT NULL

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εισαγωγή μίας γραμμής στον πίνακα *users* γίνεται ως ακολούθως:

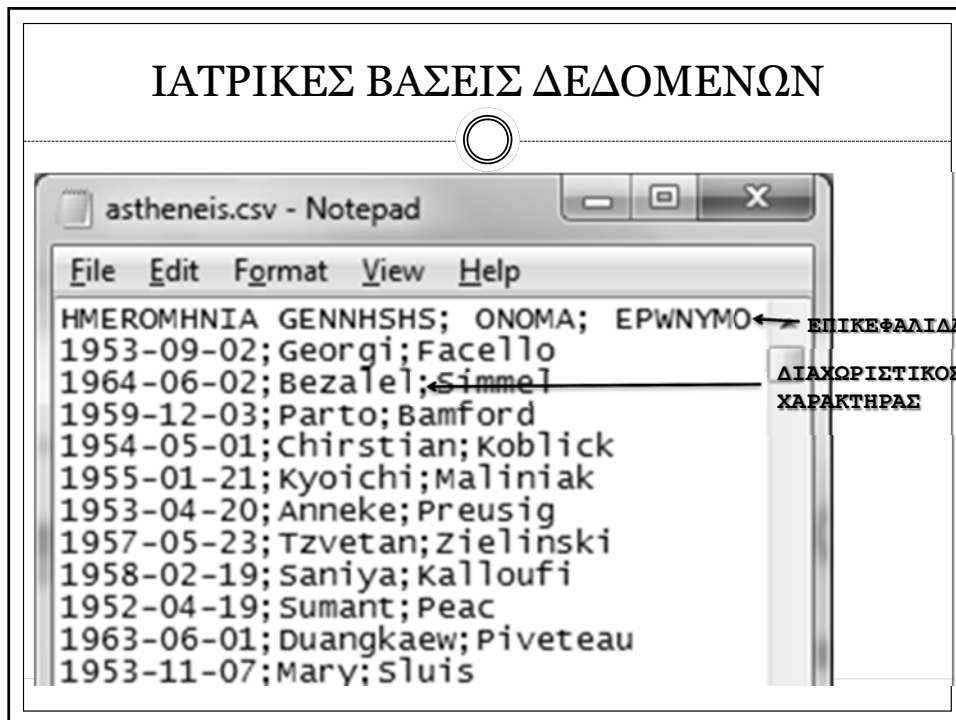
```
INSERT INTO users(firstName, lastName,  
email, username, pass, registrationDate)  
VALUES('John', 'Doe', 'j.doe@example.com',  
'johndoe', SHA1('qwerty!@#$'), NOW());
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η MySQL υποστηρίζει την εισαγωγή δεδομένων από αρχείο με χρήση της εντολής **LOAD DATA**
- Έστω το αρχείο *astheneis.csv*, που είναι στο φάκελο 'C:\labwork\iatrikh\_plhroforikh', το οποίο περιέχει την ημερομηνία γέννησης, το όνομα και το επώνυμο ασθενών:
  - Κάθε γραμμή του αρχείου αντιστοιχεί σε έναν ασθενή
  - Οι τιμές των πεδίων διαχωρίζονται με το χαρακτήρα ';'.
  - Η πρώτη γραμμή του αρχείου αποτελεί την επικεφαλίδα του αρχείου

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

- Η εντολή για την εισαγωγή των δεδομένων στον πίνακα astheneis είναι:

```
LOAD DATA LOCAL INFILE 'C:\labwork\ΙΑΤΡΙΚΗ_ΠΛΗΡΟΦΟΡΙΚΗ\astheneis.csv'
INTO TABLE astheneis
FIELDS TERMINATED BY ';'
IGNORE 1 LINES
(
  hmeromiaGennhshs,
  onoma,
  epwnymo
);
```

Annotations on the SQL code:

- An arrow points to the semicolon in the 'FIELDS TERMINATED BY' clause with the label "ΚΑΘΟΡΙΖΕΤΑΙ Ο ΔΙΑΧΩΡΙΣΤΙΚΟΣ ΧΑΡΑΚΤΗΡΑΣ".
- A bracket groups the field names in the parentheses, with the label "ΌΠΩΣ ΕΧΟΥΝ ΟΡΙΣΤΕΙ ΣΤΗΝ CREATE TABLE".

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Πολλές εντολές υποστηρίζουν τη χρήση συνθηκών στη σύνταξη τους (για παράδειγμα, όταν πρόκειται για αναζήτηση με κριτήρια, τη διαγραφή συγκεκριμένων γραμμών ενός πίνακα κ.λπ.)
- Μία συνθήκη εφαρμόζεται στις τιμές ενός πεδίου και εάν είναι αληθής η εντολή εφαρμόζεται στην αντίστοιχη γραμμή του πίνακα

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η MySQL υποστηρίζει τη χρήση των γνωστών τελεστών σύγκρισης:

Τελεστής	Περιγραφή
<code>val1 = val2</code>	Αληθής όταν το val1 είναι ίσο με val2
<code>val1 &lt;&gt; val2</code> ή <code>val1 != val2</code>	Αληθής όταν το val1 είναι διαφορετικό από το val2
<code>val1 &lt;= val2</code>	Αληθής όταν το val1 είναι μικρότερο ή ίσο από το val2
<code>val1 &lt; val2</code>	Αληθής όταν το val1 είναι μικρότερο από το val2
<code>val1 &gt;= val2</code>	Αληθής όταν το val1 είναι μεγαλύτερο ή ίσο από το val2
<code>val1 &gt; val2</code>	Αληθής όταν το val1 είναι μεγαλύτερο από το val2

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Επιπλέον, η MySQL υποστηρίζει τους ακόλουθους τελεστές:

Τελεστής	Περιγραφή
val <b>BETWEEN</b> val1 <b>AND</b> val2	Αληθές όταν το val είναι μεταξύ val1 και val2, ή ίσο με val1 ή val2
val <b>NOT BETWEEN</b> val1 <b>AND</b> val2	Αληθές όταν το val δεν είναι μεταξύ val1 και val2
val <b>IN</b> (val1, val2,...)	Αληθές όταν το val έχει μία από τις τιμές val1 ή val2 κ.λπ.
val <b>NOT IN</b> (val1, val2,...)	Αληθές όταν το val δεν έχει μία από τις τιμές της λίστας
val <b>LIKE</b> expr	Αληθές όταν το val ταιριάζει με μία έκφραση
val <b>NOT LIKE</b> expr	Αληθές όταν το val δεν ταιριάζει με μία έκφραση

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στους τελεστές σύγκρισης **LIKE** και **NOT LIKE** η έκφραση μπορεί να περιλαμβάνει τους ακόλουθους ειδικούς χαρακτήρες % ή \_
- Για παράδειγμα:
  - η έκφραση 'B%' περιγράφει όλες τις συμβολοσειρές που ξεκινούν από B και έχουν **έναν ή περισσότερους** χαρακτήρες στη συνέχεια
  - η έκφραση 'B\_' περιγράφει όλες της συμβολοσειρές που ξεκινούν από B και έχουν **ένα μόνο** χαρακτήρα στη συνέχεια

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Δύο ή περισσότερες συνθήκες μπορούν να συνδυαστούν με χρήση των λογικών τελεστών **AND**, **OR**
- Η συνθήκη (cond1) **AND** (cond2) είναι αληθής όταν οι δύο επιμέρους συνθήκες είναι αληθείς
- Η συνθήκη (cond1) **OR** (cond2) είναι αληθής όταν τουλάχιστον μία από τις δύο επιμέρους συνθήκες είναι αληθείς

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η **ανανέωση** των τιμών ενός πίνακα γίνεται με χρήση της εντολής **UPDATE**, ακολουθούμενη από το όνομα του πίνακα, τις νέες τιμές και προαιρετικά από κάποια συνθήκη
- Η εντολή  
**UPDATE astheneis SET onoma='Paul' WHERE kwdikos=1;**

αλλάζει την τιμή του πεδίου 'όνομα' σε Paul για τον ασθενή με κωδικό 1

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Αντίστοιχα, η εντολή

```
UPDATE astheneis SET epwnymo = 'Smith',  
onoma='Paul' WHERE kwdikos=10;
```

αλλάζει το όνομα και το επώνυμο σε Paul και Smith αντίστοιχα, για τον ασθενή με κωδικό 10

- Στα προηγούμενα παραδείγματα, εάν δεν υπάρχει η συνθήκη, η αλλαγή θα γίνει σε όλες τις γραμμές του πίνακα

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η **διαγραφή** γραμμών ενός πίνακα γίνεται με χρήση της εντολής **DELETE FROM**, ακολουθούμενη από το όνομα του πίνακα και προαιρετικά από κάποια συνθήκη

- Η εντολή

```
DELETE FROM astheneis WHERE kwdikos=1;
```

διαγράφει τη γραμμή του πίνακα όπου ο κωδικός είναι 1

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή  
**DELETE FROM astheneis WHERE επωνυμο LIKE 'Sm%';**  
διαγράφει τις γραμμές του πίνακα όπου το επώνυμο ξεκινάει από Sm και ακολουθείται από έναν ή περισσότερους χαρακτήρες
- Η εντολή  
**DELETE FROM astheneis;**  
διαγράφει όλες τις γραμμές του πίνακα.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η επιλογή και κατ' επέκταση η εμφάνιση δεδομένων από έναν πίνακα γίνεται με χρήση της εντολής SQL **SELECT**.
- Για τη χρήση της εντολής αυτής πρέπει να προσδιοριστούν τα πεδία τα οποία θέλουμε να εμφανιστούν και το όνομα του πίνακα.
- Προαιρετικά, μπορούμε να προσδιορίσουμε μία ή περισσότερες συνθήκες, καθώς και να ταξινομήσουμε τα δεδομένα με βάση κάποιο κριτήριο.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή για να επιλεγούν όλα τα δεδομένα του πίνακα `astheneis` είναι:

**SELECT \* FROM astheneis;**

η οποία παράγει:

kwdikos	epwnymo	onoma	hmniaGennhshs
1	Facello	Georgi	1953-09-02
2	Simmel	Bezalel	1964-06-02
3	Bamford	Parto	1959-12-03
4	Koblick	Chirstian	1954-05-01
5	Maliniak	Kyoichi	1955-01-21
6	Preusig	Anneke	1953-04-20
7	Zelinski	Tzvetan	1957-05-23
8	Kalloufi	Saniya	1958-02-19

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή για να εμφανιστούν οι τιμές που είναι στις στήλες όνομα και επώνυμο είναι:

**SELECT epwnymo, onoma FROM astheneis;**

η οποία παράγει:

epwnymo	onoma
Facello	Georgi
Simmel	Bezalel
Bamford	Parto
Koblick	Chirstian
Maliniak	Kyoichi
Preusig	Anneke
Zelinski	Tzvetan

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η MySQL υποστηρίζει τη χρήση ψευδώνυμου για το όνομα ενός πεδίου, με χρήση της λέξης **AS**

```
SELECT επωνυμο AS 'ΕΠΩΝΥΜΟ', ονομα AS 'ΟΝΟΜΑ' FROM astheneis;
```

η οποία παράγει:

ΕΠΩΝΥΜΟ	ΟΝΟΜΑ
Facello	Georgi
Simmel	Bezalel
Bamford	Parto
Koblick	Chirstian
Maliniak	Kyoichi
Preusig	Anneke
Zielinski	Tzvetan
Kalloufi	Saniya

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στην περίπτωση όπου πρέπει να υπάρχει μία ή περισσότερες συνθήκες (για παράδειγμα, το επώνυμο να ξεκινάει από Α ή η ημερομηνία γέννησης να είναι μετά τη 1-1-1960 ή ο συνδυασμός και των δύο), η εντολή πρέπει να περιλαμβάνει τη λέξη **WHERE** ακολουθούμενη από τη συνθήκη

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για παράδειγμα, η εντολή για να εμφανιστούν οι τιμές που είναι στις στήλες όνομα και επώνυμο για τους ασθενείς που το επώνυμο ξεκινάει από Α είναι:  
**SELECT επωνυμο, ονομα FROM astheneis  
 WHERE επωνυμο LIKE 'A%';**

η οποία παράγει

επωνυμο	ονομα
Awdeh	Satosi
Azuma	Mona
Aingworth	Eben
Auria	Yucel
Azumi	Douadi
Aloisi	Nigel

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή για να εμφανιστούν οι τιμές που είναι στις στήλες επώνυμο και ημερομηνία γέννησης για τους ασθενείς με ημερομηνία γέννησης μετά την 1-1-1960 είναι:

**SELECT επωνυμο, hμνιαGennshs FROM  
 astheneis WHERE hμνιαGennshs >= '1960-  
 01-01';**

η οποία παράγει

επωνυμο	hμνιαGennshs
Simmel	1964-06-02
Piveteau	1963-06-01
Bridgland	1960-10-04
Terkki	1963-06-07
Cappelletti	1961-05-02
Erde	1960-02-20

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή για να εμφανιστούν οι τιμές που είναι στις στήλες όνομα, επώνυμο και ημερομηνία γέννησης για τους ασθενείς ημερομηνία γέννησης μετά την 1-1-1960 και το επώνυμό τους ξεκινάει από Α είναι:

```
SELECT επωνυμο, ονομα, hμνιαGεnnhshs
FROM astheneis
WHERE επωνυμο LIKE 'A%'
      AND hμνιαGεnnhshs >= '1960-01-01';
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η προηγούμενη εντολή παράγει το αποτέλεσμα:

επωνυμο	ονομα	hμνιαGεnnhshs
Awdeh	Satosi	1963-04-14
Azuma	Mona	1964-04-18
Ananiadou	Aleksandar	1960-01-11
Awdeh	Vidya	1961-12-31
Alencar	Aksel	1964-08-29
Apsitis	Christfried	1961-05-04
Aumann	Leucio	1963-11-13
Ananiadou	Munehiko	1964-12-22

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Αν θέλουμε τα αποτελέσματα να είναι ταξινομημένα με βάση τις τιμές κάποιας στήλης πρέπει να χρησιμοποιηθούν οι λέξεις **ORDER BY** ακολουθούμενες από το όνομα της στήλης ως προς την οποία θα γίνει η ταξινόμηση
- Για παράδειγμα, αν επιθυμείται να εμφανιστεί το όνομα, το επώνυμο και η ημερομηνία γέννησης των ασθενών που έχουν γεννηθεί μετά την 1-1-1960 και τα αποτελέσματα να ταξινομηθούν με βάση την ημερομηνία γέννησης σε αύξουσα σειρά (από πιο παλιές προς πιο πρόσφατες), η εντολή είναι

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



```
SELECT επωνυμο, ονομα, hmniaGennhshs
FROM astheneis WHERE hmniaGennhshs >=
'1960-01-01' ORDER BY hmniaGennhshs ASC;
```

η οποία παράγει το αποτέλεσμα

επωνυμο	ονομα	hmniaGennhshs
Klerer	Xiping	1960-01-02
Ananiadou	Aleksandar	1960-01-11
Bahk	Tesuro	1960-01-12
Ishibashi	Shahaf	1960-01-17
Piazza	Momie	1960-01-25
Ohori	Erzsebet	1960-01-28

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Αν στο προηγούμενο παράδειγμα, θέλουμε τα αποτελέσματα να ταξινομηθούν με βάση την ημερομηνία γέννησης σε φθίνουσα σειρά (από πιο πρόσφατες προς πιο παλιές), η εντολή είναι:

```
SELECT epwnymo, onoma, hmniaGennhshs  
FROM astheneis WHERE hmniaGennhshs >=  
'1960-01-01' ORDER BY hmniaGennhshs DESC;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



η οποία παράγει το αποτέλεσμα

epwnymo	onoma	hmniaGennhshs
Baba	Make	1965-01-25
Seghrouchni	Juyoung	1965-01-24
Seghrouchni	Dipayan	1965-01-23
Esposito	Ohad	1965-01-19
Himler	Lech	1965-01-19
Gecsei	Takahito	1965-01-19
Noriega	Teunis	1965-01-09
Morton	Hilari	1965-01-03
Iisaka	Kokou	1965-01-01

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η MySQL υποστηρίζει την εκτέλεση συνηθισμένων αριθμητικών πράξεων μεταξύ των τιμών δύο ή περισσότερων πεδίων
- Για παράδειγμα, έστω ότι στον πίνακα με τους ασθενείς υπάρχουν επιπλέον τα πεδία baros και ypsos για το βάρος (σε kg) και του ύψος (σε cm) αντίστοιχα
- Για να εμφανιστεί το επώνυμο, το βάρος, το ύψος και ο δείκτης σωματικής μάζας ( $BMI = 10^4 \times \text{baros} / \text{ypsos}^2$ ), η εντολή είναι:

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



**SELECT**

```
epwnymo AS 'ΕΠΩΝΥΜΟ',
baros AS 'ΒΑΡΟΣ (kg)',
ypsos AS 'ΥΨΟΣ (cm)',
1e4 * baros / (ypsos * ypsos) AS 'BMI (kg/m^2)'
```

**FROM** astheneis;

η οποία παράγει:

ΕΠΩΝΥΜΟ	ΒΑΡΟΣ (kg)	ΥΨΟΣ (cm)	BMI (kg/m <sup>2</sup> )
Facello	80	180	24.691358024691358
Simmel	70	178	22.093170054286077
Bamford	78	171	26.674874320303683
Koblick	82	190	22.714681440443215
Maliniak	79	171	27.016859888512705
Preusig	91	200	22.75

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στην περίπτωση όπου δύο πίνακες συνδέονται (για παράδειγμα ασθενείς και εξετάσεις) και επιθυμείται να εμφανιστούν δεδομένα και από τους δύο πίνακες, η εντολή **SELECT** χρησιμοποιείται ως ακολούθως:

```
SELECT column(s) FROM tbl_name1, tbl_name2  
WHERE tbl_name1.column = tbl_name2.column;
```

όπου:

- `tbl_name1` : όνομα του πρώτου πίνακα
- `tbl_name2` : όνομα του δεύτερου πίνακα
- `columns`: οι στήλες που θα εμφανιστούν
- `tbl_name1.column tbl_name2.column`: οι στήλες μέσω των οποίων συνδέονται δύο πίνακες

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για παράδειγμα, έστω ο πίνακας με εξετάσεις:

```
CREATE TABLE exetaseis(  
    kwdikos INT AUTO_INCREMENT PRIMARY KEY,  
    hmniaExetashs DATE,  
    eidos VARCHAR(50),  
    kwd_asth INT,  
    FOREIGN KEY(kwd_asth)  
        REFERENCES astheneis(kwdikos)  
) ENGINE=InnoDB;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για να εμφανιστούν το όνομα και το επώνυμο κάθε ασθενή, καθώς και το είδος και η ημερομηνία των εξετάσεων κάθε ασθενή, η εντολή είναι:

```
SELECT
    επωνυμο, ονομα, ειδος, ημνιαExetashs
FROM
    astheneis,
    exetaseis
WHERE
    astheneis.kwdikos = exetaseis.kwd_asth;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Αν επιθυμείται να εμφανιστούν το όνομα και το επώνυμο κάθε ασθενή, καθώς και το είδος και η ημερομηνία των εξετάσεων κάθε ασθενή, για τους ασθενείς με ημερομηνία γέννησης μετά την 1-1-1960, η εντολή είναι:

```
SELECT
    επωνυμο, ονομα, ειδος, ημνιαExetashs
FROM
    astheneis,
    exetaseis
WHERE
    astheneis.kwdikos = exetaseis.kwd_asth
    AND ημνιαGennhshs >= '1960-01-01';
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στην περίπτωση όπου δύο πεδία στους πίνακες έχουν την ίδια ονομασία, η εντολή θα πρέπει να διαφοροποιηθεί ελαφρώς.
- Για παράδειγμα, αν στους πίνακες με τους ασθενείς η ημερομηνία γέννησης είχε οριστεί ως hmnia και το ίδιο όνομα είχε δοθεί και στο πεδίο ημερομηνία εξέτασης στον πίνακα εξετάσεις, τότε η προηγούμενη εντολή θα έπρεπε να γραφτεί:

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



```

SELECT
    επωνυμο, ονομα, ειδος, exetaseis.hmnia
FROM
    astheneis,
    exetaseis
WHERE
    astheneis.kwdikos = exetaseis.kwd_asth
    AND astheneis.hmnia >= '1960-01-01';

```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η παραπάνω διαδικασία εφαρμόζεται όταν είναι να επιλέξουμε δεδομένα από τρεις πίνακες.
- Για παράδειγμα, έστω ο πίνακας με ιατρούς:

```
CREATE TABLE iatroι (
kwdikos MEDIUMINT UNSIGNED AUTO_INCREMENT PRIMARY
KEY,
onoma VARCHAR(20) NOT NULL,
erwnymo VARCHAR(30) NOT NULL,
eidikohta VARCHAR(30) NOT NULL DEFAULT ''
)ENGINE=InnoDB;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Επίσης, έστω ότι ο πίνακας με τις εξετάσεις έχει δημιουργηθεί ως ακολούθως

```
CREATE TABLE exetaseis(
kwdikos INT AUTO_INCREMENT PRIMARY KEY,
hmniaExetashs DATE,
eidos VARCHAR(50),
kwd_asth INT,
kwd_iat MEDIUMINT UNSIGNED,
FOREIGN KEY(kwd_asth)
REFERENCES astheneis(kwdikos),
FOREIGN KEY(kwd_iat)
REFERENCES iatroι(kwdikos)
) ENGINE=InnoDB;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Αν επιθυμείται να εμφανιστούν το όνομα και το επώνυμο κάθε ασθενή, το είδος και η ημερομηνία των εξετάσεων κάθε ασθενή καθώς και το όνομα και επώνυμο του ιατρού που ζήτησε την εξέταση, η εντολή είναι:

```
SELECT astheneis.onoma, astheneis.epwnymo,
       eidos, hmniaExetashs, iatroi.onoma,
       iatroi.epwnymo
FROM   astheneis, exetaseis, iatroi
WHERE
       astheneis.kwdikos = exetaseis.kwd_asth
AND   iatroi.kwdikos = exetaseis.kwd_iat;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΑΣΚΗΣΗ 9

1. Για τον πίνακα με τους χρήστες, περιγράψτε αναλυτικά με ποιον τρόπο SQL μπορεί να διαπιστωθεί:
  - A. εάν υπάρχει χρήστης με συγκεκριμένο όνομα χρήστη και συνθηματικό
  - B. εάν τα στοιχεία που πρόκειται να εισαχθούν κατά την εγγραφή ενός νέου χρήστη είναι έγκυρα
2. Στον πίνακα με τους ασθενείς, να εμφανιστούν:
  - A. Το όνομα, το επώνυμο και η ημερομηνία γέννησης των ασθενών που είναι μεταξύ 40 και 50 ετών (με δύο τρόπους)
  - B. Το επώνυμο και η ηλικία κάθε ασθενή (Υπόδειξη: χρησιμοποιήστε τις συναρτήσεις NOW, DATEDIFF και ROUND).
  - C. Το ονοματεπώνυμο ως ένα πεδίο π.χ. 'John Doe' (Υπόδειξη: χρησιμοποιήστε τη συνάρτηση CONCAT).

(Για επαλήθευση, μπορείτε να χρησιμοποιήσετε το αρχείο με δεδομένα ασθενών που υπάρχει στο eclass στο φάκελλο ΔΕΔΟΜΕΝΑ ΑΣΚΗΣΕΩΝ)