

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 επιλογή δεδομένων από δύο ή περισσότερους πίνακες μπορεί να επιτευχθεί χρησιμοποιώντας τις λέξεις **INNER JOIN (ή μόνο JOIN)** σε μία εντολή **SELECT**
- Συνήθως, οι πίνακες από τους οποίους πρόκειται να επιλεγούν τα δεδομένα συνδέονται με μία σχέση ένα προς πολλά και το εξωτερικό κλειδί του ενός πίνακα συνδέεται με το πρωτεύον κλειδί του άλλου πίνακα

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Έστω, για παράδειγμα οι πίνακες για **ασθενείς**, **ιατρούς** και **εξετάσεις**

```
CREATE TABLE astheneis (
  kwdikos mediumint unsigned NOT NULL AUTO_INCREMENT PRIMARY KEY,
  onoma varchar(20) NOT NULL DEFAULT '',
  epwnymo varchar(30) NOT NULL DEFAULT '',
  hmniaGennhshs date NOT NULL,
  baros float DEFAULT 0.0,
  ypsos tinyint unsigned DEFAULT 0
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Έστω, για παράδειγμα οι πίνακες για ασθενείς, **ιατρούς** και εξετάσεις

```
CREATE TABLE iatroi (
  kwdikos mediumint unsigned NOT NULL AUTO_INCREMENT PRIMARY KEY,
  onoma varchar(20) NOT NULL DEFAULT '',
  epwnymo varchar(30) NOT NULL DEFAULT '',
  eidikothta varchar(20) NOT NULL DEFAULT ''
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Έστω, για παράδειγμα οι πίνακες για ασθενείς, **ιατρούς** και **εξετάσεις**

```
CREATE TABLE exetaseis (
  kwdikos int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  hmniaExetashs date NOT NULL,
  eidos varchar(50) NOT NULL,
  kwd_asth mediumint unsigned,
  kwd_iat mediumint unsigned,
  FOREIGN KEY (kwd_asth)
    REFERENCES astheneis (kwdikos),
  FOREIGN KEY (kwd_iat)
    REFERENCES iatroi (kwdikos)
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

- Για να επιλεγούν το όνομα και το επώνυμο ασθενή, καθώς και το είδος και η ημερομηνία εξέτασης, η εντολή SQL είναι

```
SELECT
  επωνυμο AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ',
  ονομα AS 'ΟΝΟΜΑ ΑΣΘΕΝΗ',
  ειδος AS 'ΕΙΔΟΣ ΕΞΕΤΑΣΗΣ',
  ημνιαExetashs AS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ'
FROM
  astheneis
  INNER JOIN
  exetaseis ON astheneis.kwdikos = exetaseis.kwd_asth;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

- Με την εντολή αυτή θα εμφανιστούν μόνο οι ασθενείς που έχουν κάνει κάποια εξέταση. Αν υπάρχουν ασθενείς που δεν έχουν καταχωρημένη εξέταση, αυτοί δεν θα εμφανιστούν

ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ	ΟΝΟΜΑ ΑΣΘΕΝΗ	ΕΙΔΟΣ ΕΞΕΤΑΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ
Facello	Georgi	CT	2010-01-13
Simmel	Bezalel	MRI	2010-02-14
Merlo	Arif	XRAY	2010-03-15
Casley	Mingsen	CT	2010-04-24
Randi	Hinrich	CT	2010-05-11
Nourani	Tze	MRI	2010-06-21
Dymetman	Snehasis	MRI	2020-10-07

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η σύνταξη της προηγούμενης εντολής μπορεί να απλοποιηθεί χρησιμοποιώντας τη λέξη **USING** με την προϋπόθεση ότι το εξωτερικό κλειδί του ενός πίνακα έχει το ίδιο όνομα με το πρωτεύον κλειδί του άλλου πίνακα
- Για παράδειγμα, εάν στον πίνακα με τους ασθενείς το πρωτεύον κλειδί ονομαζόταν `astheneis_id` και το ίδιο όνομα είχε δοθεί στο εξωτερικό κλειδί στον πίνακα με τις εξετάσεις, τότε η προηγούμενη εντολή θα μπορούσε να γραφτεί

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



```
SELECT
    επωνυμο AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ',
    ονομα AS 'ΟΝΟΜΑ ΑΣΘΕΝΗ',
    ειδος AS 'ΕΙΔΟΣ ΕΞΕΤΑΣΗΣ',
    ημνιαExetashs AS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ'
FROM
    astheneis
    INNER JOIN
    exetaseis USING (astheneis_id);
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οποιοσδήποτε από τους δύο τρόπους και εάν επιλεγεί (χρήση λέξης ON ή χρήση λέξης USING), η εντολή μπορεί να περιλαμβάνει τις λέξεις WHERE και ORDER BY
- Έστω ότι στο προηγούμενο παράδειγμα θέλαμε τα εμφανιστούν οι εξετάσεις που έγιναν το πρώτο εξάμηνο του 2010 και τα αποτελέσματα να είναι ταξινομημένα με βάση την ημερομηνία εξέτασης από παλαιότερες σε νεώτερες,

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή θα ήταν

```
SELECT
  επωνυμο AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ',
  onoma AS 'ΟΝΟΜΑ ΑΣΘΕΝΗ',
  eidos AS 'ΕΙΔΟΣ ΕΞΕΤΑΣΗΣ',
  hmniaExetashs AS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ'
FROM
  astheneis
  INNER JOIN
  exetaseis ON astheneis.kwdikos = exetaseis.kwd_asth
WHERE
  hmniaExetashs BETWEEN '2010-01-01' AND '2010-06-30'
ORDER BY hmniaExetashs ASC;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Το αποτέλεσμα θα ήταν

ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ	ΟΝΟΜΑ ΑΣΘΕΝΗ	ΕΙΔΟΣ ΕΞΕΤΑΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ
Facello	Georgi	CT	2010-01-13
Simmel	Bezalel	MRI	2010-02-14
Merlo	Arif	XRAY	2010-03-15
Casley	Mingsen	CT	2010-04-24
Rendi	Hinrich	CT	2010-05-11
Nourani	Tze	MRI	2010-06-21

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στην περίπτωση που επιθυμείται να επιλεγούν δεδομένα από 3 πίνακες, για παράδειγμα επώνυμο ασθενή, επώνυμο ιατρού και ημερομηνία εξέτασης, η χρήση της λέξης JOIN γίνεται ως ακολούθως

```

SELECT
  astheneis.epwnymo AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ',
  iatroi.epwnymo AS 'ΕΠΩΝΥΜΟ ΙΑΤΡΟΥ',
  hmniaExetashs AS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ'
FROM
  astheneis
  INNER JOIN
  exetaseis ON astheneis.kwdikos = exetaseis.kwd_asth
  INNER JOIN
  iatroi ON iatroi.kwdikos = exetaseis.kwd_iat;

```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή θα παράγει το ακόλουθο αποτέλεσμα

ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ	ΕΠΩΝΥΜΟ ΙΑΤΡΟΥ	ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ
Facello	Pleszkun	2010-01-13
Simmel	Ghelli	2010-02-14
Merlo	Dymetman	2010-03-15
Casley	Garnier	2010-04-24
Randi	Kavanagh	2010-05-11
Nourani	Narlikar	2010-06-21

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Όπως και πριν, η σύνταξη της εντολής μπορεί να απλοποιηθεί με χρήση της λέξης `USING` ως εξής:
 - έστω ότι στον πίνακα με τους ασθενείς το πρωτεύον κλειδί ονομαζόταν `astheneis_id` και το ίδιο όνομα είχε δοθεί στο αντίστοιχο εξωτερικό κλειδί στον πίνακα με τις εξετάσεις
 - έστω ότι στον πίνακα με τους ιατρούς το πρωτεύον κλειδί ονομαζόταν `iatroi_id` και το ίδιο όνομα είχε δοθεί στο αντίστοιχο εξωτερικό κλειδί στον πίνακα με τις εξετάσεις

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



```
SELECT
  astheneis.epwnymo AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ',
  iatroi.epwnymo AS 'ΕΠΩΝΥΜΟ ΙΑΤΡΟΥ',
  hmniaExetashs AS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ'
FROM
  astheneis
  INNER JOIN
  exetaseis USING (astheneis_id)
  INNER JOIN
  iatroi USING (iatroi_id);
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Πρέπει να τονιστεί ότι εάν δύο πίνακες δεν συνδέονται άμεσα (όπως ασθενείς και ιατροί), αλλά μέσω ενός τρίτου πίνακα (εξετάσεις), τότε ακόμα και δεν ζητούνται δεδομένα από τον τρίτο πίνακα, ο πίνακας αυτός πρέπει να χρησιμοποιηθεί
- Για παράδειγμα, για να εμφανιστούν το επώνυμο ασθενή και το επώνυμο ιατρού

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

```

SELECT
  astheneis.epwnymo AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ', iatroi.epwnymo AS 'ΕΠΩΝΥΜΟ ΙΑΤΡΟΥ'
FROM
  astheneis
  INNER JOIN
  exetaseis ON astheneis.kwdikos = exetaseis.kwd_asth
  INNER JOIN
  iatroi ON iatroi.kwdikos = exetaseis.kwd_iat;

```

ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ	ΕΠΩΝΥΜΟ ΙΑΤΡΟΥ
Facello	Pleszkun
Simmel	Ghelli
Merlo	Dymetman
Casley	Garnier
Randi	Kavanagh
Nourani	Narlikar
Dymetman	Rodham

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

- Το πλεονέκτημα της εφαρμογής της JOIN για επιλογή δεδομένων από δύο πίνακες είναι ότι μπορεί να χρησιμοποιηθεί για την επιλογή δεδομένων από τον έναν πίνακα από τους δύο πίνακες ακόμα και αν οι εγγραφές του δεν σχετίζονται με εγγραφές του άλλου πίνακα.
- Για παράδειγμα, έστω ότι επιθυμείται να εμφανιστούν όνομα και επώνυμο για ασθενείς ακόμα και εάν δεν έχουν κάνει κάποια εξέταση και για όσους υπάρχουν εξετάσεις, να εμφανιστούν το είδος και η ημερομηνία εξέτασης.

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

```

SELECT
  επωνυμο, ονομα, ειδος, hmniaExetashs
FROM
  astheneis
  LEFT JOIN
  exetaseis ON astheneis.kwdikos = exetaseis.kwd_asth;

```

επωνυμο	ονομα	ειδος	hmniaExetashs
Facello	Georgi	CT	2010-01-13
Simmel	Bezalel	MRI	2010-02-14
Bamford	Parto	NULL	NULL
Koblick	Chirstian	NULL	NULL
Maliniak	Kyoichi	NULL	NULL
Preusig	Anneke	NULL	NULL
Zielinski	Tzvetan	NULL	NULL
Kalloufi	Saniya	NULL	NULL

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

- Η MySQL υποστηρίζει συναρτήσεις που μπορούν εφαρμοστούν στις τιμές μίας στήλης ενός πίνακα
- Παραδείγματα τέτοιων συναρτήσεων είναι:
 - ABS(col_name): υπολογίζει την απόλυτη τιμή κάθε τιμή της στήλης col_name
 - ROUND(col_name): υπολογίζει τον πλησιέστερο ακέραιο κάθε τιμή της στήλης col_name
 - ROUND(col_name, n): στρογγυλοποιεί κάθε τιμή της στήλης col_name χρησιμοποιώντας n δεκαδικά ψηφία
 -

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Μία ιδιαίτερη χρήσιμη συνάρτηση είναι η **DATE_FORMAT**(col_name, format), όπου col_name είναι το όνομα μίας στήλης που περιέχει ημερομηνίες και format είναι μία συμβολοσειρά που καθορίζει τη μορφοποίηση
- Έστω για παράδειγμα θέλαμε να εμφανιστούν το επώνυμο και το επώνυμο ασθενών και η ημερομηνία εξέτασης στη συνήθη μορφή (μέρα-μήνας-χρόνος)

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή θα είναι

```
SELECT
  epwnymo AS 'ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ',
  onoma AS 'ΟΝΟΜΑ ΑΣΘΕΝΗ',
  DATE_FORMAT(hmniaExetashs, '%d-%m-%Y') AS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ'
FROM
  astheneis
  INNER JOIN
  exetaseis ON astheneis.kwdikos = exetaseis.kwd_asth;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Το αποτέλεσμα θα είναι

ΕΠΩΝΥΜΟ ΑΣΘΕΝΗ	ΟΝΟΜΑ ΑΣΘΕΝΗ	ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ
Facello	Georgi	13-01-2010
Simmel	Bezalel	14-02-2010
Merlo	Arif	15-03-2010
Casley	Mingsen	24-04-2010
Randi	Hinrich	11-05-2010
Nourani	Tze	21-06-2010
Dymetman	Snehasis	07-10-2020

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η MySQL υποστηρίζει τις ακόλουθες συναρτήσεις, οι οποίες χρησιμοποιούνται με την εντολή **SELECT**:
 - **COUNT(field)**: επιστρέφει το πλήθος των γραμμών του πίνακα που έχουν τιμή στη στήλη field
 - **MIN(field)**: επιστρέφει την ελάχιστη τιμή της στήλης field
 - **MAX(field)**: επιστρέφει τη μέγιστη τιμή της στήλης field
 - **AVG(field)**: επιστρέφει το μέσο όρο των τιμών της στήλης field
 - **SUM(field)**: επιστρέφει το άθροισμα της στήλης field

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για παράδειγμα, για να υπολογιστεί η μέση τιμή του βάρους των ασθενών, η εντολή είναι:

```
SELECT
  AVG(baros) AS 'ΜΕΣΗ ΤΙΜΗ ΒΑΡΟΥΣ'
FROM
  astheneis;
```

ΜΕΣΗ ΤΙΜΗ ΒΑΡΟΥΣ

73.7

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για να υπολογιστούν η ελάχιστη και η μέγιστη τιμή του βάρους των ασθενών, η εντολή είναι:

```
SELECT
  MIN(baros) AS 'ΕΛΑΧΙΣΤΗ ΤΙΜΗ ΒΑΡΟΥΣ',
  MAX(baros) AS 'ΜΕΓΙΣΤΗ ΤΙΜΗ ΒΑΡΟΥΣ'
FROM
  astheneis;
```

	ΕΛΑΧΙΣΤΗ ΤΙΜΗ ΒΑΡΟΥΣ	ΜΕΓΙΣΤΗ ΤΙΜΗ ΒΑΡΟΥΣ
▶	56	93

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για να υπολογιστεί το πλήθος των γραμμών του πίνακα ασθενών, η εντολή είναι

```
SELECT  
  COUNT(*) AS 'ΠΛΗΘΟΣ ΑΣΘΕΝΩΝ'  
FROM  
  astheneis;
```

ΠΛΗΘΟΣ ΑΣΘΕΝΩΝ
1000

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Άσκηση**

Έστω μία βάση δεδομένων με πληροφορίες για πελάτες, κατηγορίες προϊόντων, προϊόντα, παραγγελίες και λεπτομέρειες παραγγελιών μιας εταιρείας, η οποία περιλαμβάνει τους ακόλουθους πίνακες:

- πίνακας πελατών *customers*:
 - κωδικός πελάτη
 - επώνυμο
 - όνομα

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Άσκηση**

Έστω μία βάση δεδομένων με πληροφορίες για πελάτες, κατηγορίες προϊόντων, προϊόντα, παραγγελίες και λεπτομέρειες παραγγελιών μιας εταιρείας, η οποία περιλαμβάνει τους ακόλουθους πίνακες:

- πίνακας κατηγοριών προϊόντων *product_categories*:
 - κωδικός κατηγορίας
 - περιγραφή

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Άσκηση**

Έστω μία βάση δεδομένων με πληροφορίες για πελάτες, κατηγορίες προϊόντων, προϊόντα, παραγγελίες και λεπτομέρειες παραγγελιών μιας εταιρείας, η οποία περιλαμβάνει τους ακόλουθους πίνακες:

- πίνακας προϊόντων *products*:
 - κωδικός προϊόντος
 - κωδικός κατηγορίας
 - ονομασία
 - τιμή
 - απόθεμα

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Άσκηση**

Έστω μία βάση δεδομένων με πληροφορίες για πελάτες, κατηγορίες προϊόντων, προϊόντα, παραγγελίες και λεπτομέρειες παραγγελιών μιας εταιρείας, η οποία περιλαμβάνει τους ακόλουθους πίνακες:

- πίνακας παραγγελιών *orders*:
 - κωδικός
 - κωδικός πελάτη
 - ημερομηνία παραγγελίας
 - σχόλια
 - κατάσταση

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Άσκηση**

Έστω μία βάση δεδομένων με πληροφορίες για πελάτες, κατηγορίες προϊόντων, προϊόντα, παραγγελίες και λεπτομέρειες παραγγελιών μιας εταιρείας, η οποία περιλαμβάνει τους ακόλουθους πίνακες:

- πίνακας λεπτομερειών παραγγελιών *order_details*:
 - κωδικός
 - κωδικός παραγγελίας
 - κωδικός προϊόντος
 - τεμάχια

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



Άσκηση

Να γραφτούν εντολές SQL ώστε να:

1. δημιουργούνται οι πίνακες
2. εμφανίζονται επώνυμο και όνομα πελατών και οι ημερομηνίες των αντίστοιχων παραγγελιών. Να γίνει ταξινόμηση με βάση την ημερομηνία παραγγελίας από παλαιότερες σε νεώτερες
3. εμφανίζονται οι ονομασίες των προϊόντων που είναι στην κατηγορία *Αναλώσιμα Εργαστηρίων*
4. εμφανίζονται οι ημερομηνίες των 5 τελευταίων παραγγελιών του πελάτη με ονοματεπώνυμο *John Doe*, ταξινομημένες από την πιο πρόσφατη προς την πιο παλιά
5. να υπολογίζεται το πλήθος των παραγγελιών για τον πελάτη με ονοματεπώνυμο *John Doe*

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



Λύση

1. Δημιουργία πινάκων

Πίνακας πελατών

```
CREATE TABLE customers (
  customer_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  last_name VARCHAR(30) NOT NULL,
  first_name VARCHAR(20) NOT NULL
) ENGINE=InnoDB;
```

Πίνακας κατηγοριών προϊόντων

```
CREATE TABLE product_categories (
  product_category_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  description TEXT NOT NULL
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Λύση

1. Δημιουργία πινάκων

Πίνακας προϊόντων

```
CREATE TABLE products (  
  product_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  product_category_id INT UNSIGNED,  
  product_name VARCHAR(50) NOT NULL,  
  price DOUBLE NOT NULL,  
  stock SMALLINT UNSIGNED NOT NULL,  
  FOREIGN KEY (product_category_id)  
    REFERENCES product_categories (product_category_id)  
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Λύση

1. Δημιουργία πινάκων

Πίνακας παραγγελιών

```
CREATE TABLE orders (  
  order_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  customer_id INT UNSIGNED,  
  order_date DATE NOT NULL,  
  comments TEXT,  
  order_status TINYINT UNSIGNED DEFAULT 0,  
  /*  
  order_status = 0 => order received  
  order_status = 1 => processing order  
  order_status = 2 => processing dispatched  
  */  
  FOREIGN KEY (customer_id)  
    REFERENCES customers (customer_id)  
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Λύση

1. Δημιουργία πινάκων

Πίνακας λεπτομέρειες παραγγελιών

```
CREATE TABLE order_details (  
  order_detail_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  order_id INT UNSIGNED,  
  product_id INT UNSIGNED,  
  quantity SMALLINT UNSIGNED,  
  FOREIGN KEY (order_id)  
    REFERENCES orders (order_id),  
  FOREIGN KEY (product_id)  
    REFERENCES products (product_id)  
) ENGINE=InnoDB;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Λύση

2. Επώνυμο και όνομα πελατών και οι ημερομηνίες των αντίστοιχων παραγγελιών. Ταξινόμηση με βάση την ημερομηνία παραγγελίας από παλαιότερες σε νεώτερες

```
SELECT  
  last_name, first_name, order_date  
FROM  
  customers  
  JOIN  
  orders USING (customer_id)  
ORDER BY order_date ASC;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Λύση**

3. Ονομασίες των προϊόντων που είναι στην κατηγορία *Αναλώσιμα Εργαστηρίων*

```
SELECT
    product_name
FROM
    product_categories
    JOIN
    products USING (product_category_id)
WHERE
    product_categories.description = 'ΑΝΑΛΩΣΙΜΑ ΕΡΓΑΣΤΗΡΙΩΝ';
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Λύση**

4. Ημερομηνίες των 5 τελευταίων παραγγελιών του πελάτη με ονοματεπώνυμο *John Doe*. Ταξινόμηση από την πιο πρόσφατη προς την πιο παλιά

```
SELECT
    orders.order_date
FROM
    customers
    JOIN
    orders USING (customer_id)
WHERE
    customers.first_name = 'John'
    AND customers.last_name = 'Doe'
ORDER BY orders.order_date DESC
LIMIT 5;
```

ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

**Λύση**

5. Πλήθος των παραγγελιών για τον πελάτη με ονοματεπώνυμο *John Doe*

```
SELECT
    COUNT(*)
FROM
    customers
    JOIN
    orders USING (customer_id)
WHERE
    customers.first_name = 'John'
    AND customers.last_name = 'Doe';
```