

**Ανοικτά Ακαδημαϊκά Μαθήματα**

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας**

**Βάσεις Δεδομένων Ι**

**Ενότητα 16:** Εξετάσεις – Θέματα και κάποιες ενδεικτικές λύσεις

Χ. Σκουρλάς, Α. Τσολακίδης

Τμήμα Μηχανικών Πληροφορικής ΤΕ

|  |  |
| --- | --- |
| Το περιεχόμενο του μαθήματος διατίθεται με άδεια Creative Commons εκτός και αν αναφέρεται διαφορετικά | Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους. |

Περιεχόμενα

[Θέμα 1 3](#_Toc401579485)

[Θέμα 2 7](#_Toc401579486)

Στόχος του εργαστηρίου είναι η (αυτό)αξιολόγηση του φοιτητή σε θέματα σχεδίασης, δημιουργίας και διαχείρισης μίας σχεσιακής βάσης δεδομένων με πραγματικά Προϊόντα Διαχείρισης Βάσης Δεδομένων (Data Base Management Systems).

**Λέξεις Κλειδιά:** Σύστημα Διαχείρισης Βάσης Δεδομένων, Μοντελοποίηση, Κανονικοποίηση, SQL, mySQL, Oracle

Θέμα 1

**Έστω η παρακάτω βάση δεδομένων Νοσοκομείων**

Patient (πίνακας ασθενών)

|  |  |  |
| --- | --- | --- |
| P\_id | Pname | Job |
| 10 | ΣΠΥΡΟΥ | ΕΜΠΟΡΟΣ |
| 20 | ΙΩΑΝΝΟΥ | ΠΩΛΗΤΗΣ |
| 30 | ΠΕΤΡΟΥ | ΕΜΠΟΡΟΣ |

Hospital (πίνακας νοσοκομείων)

|  |  |
| --- | --- |
| H\_id | Hname  |
| 100 | ΜΕΤΑΞΑ |
| 200 | ΑΡΕΤΑΙΕΙΟ |

Ward (πίνακας κλινικών) κύριο κλειδί = (H\_id, W\_id)

|  |  |  |
| --- | --- | --- |
| W\_id | Wname | H\_id |
| 1000 | ΑΚΤΙΝΟΛΟΓΙΚΟ | 100 |
| 2000 | ΚΑΡΔΙΟΛΟΓΙΚΟ | 100 |
| 2000 | ΚΑΡΔΙΟΛΟΓΙΚΟ | 200 |
| 1000 | ΑΚΤΙΝΟΛΟΓΙΚΟ | 200 |

Occupy (εισαγωγή ασθενούς σε κλινική)

|  |  |  |  |
| --- | --- | --- | --- |
| H\_id | W\_Id | P\_id | Entry\_Date |
| 100 | 1000 | 10 | 1.1.2014 |
| 100 | 1000 | 20 | 1.1.2014 |
| 100 | 2000 | 10 | 5.1.2014 |
| 200 | 1000 | 10 | 29.1.2014 |
| 200 | 1000 | 30 | 2.1.2014 |

Υποτίθεται ότι το κύριο κλειδί είναι σύνθετο = (H\_id, W\_id, P\_id, Entry\_date)

1. **Η βάση είναι στην Τρίτη κανονική μορφή; Αν όχι γράψτε την Τρίτη κανονική μορφή.**

Στην Τρίτη κανονική μορφή ο πίνακας

Ward (πίνακας κλινικών) κύριο κλειδί = (H\_id, W\_id)

|  |  |  |
| --- | --- | --- |
| W\_id | Wname | H\_id |
| 1000 | ΑΚΤΙΝΟΛΟΓΙΚΟ | 100 |
| 2000 | ΚΑΡΔΙΟΛΟΓΙΚΟ | 100 |
| 2000 | ΚΑΡΔΙΟΛΟΓΙΚΟ | 200 |
| 1000 | ΑΚΤΙΝΟΛΟΓΙΚΟ | 200 |

Αντικαθίσταται από τους πίνακες:

Ward\_list (πίνακας κωδικοποίησης κλινικών) κύριο κλειδί = (W\_id)

|  |  |
| --- | --- |
| W\_id | Wname |
| 1000 | ΑΚΤΙΝΟΛΟΓΙΚΟ |
| 2000 | ΚΑΡΔΙΟΛΟΓΙΚΟ |

Hosp\_Ward (πίνακας κλινικών νοσοκομείων) κύριο κλειδί = (H\_id, W\_id)

|  |  |
| --- | --- |
| W\_id | H\_id |
| 1000 | 100 |
| 2000 | 100 |
| 2000 | 200 |
| 1000 | 200 |

Άλλες αλλαγές που προτάθηκαν από σπουδαστές και θα μπορούσαμε να δεχθούμε:

* Κωδικοποίηση της στήλης job. Άρα θα μπορούσαμε να έχουμε δύο πίνακες job(jobno, job), patient(p\_id, pname, jobno) αντί του πίνακα patient
* Εισαγωγή στήλης accno ως κύριου κλειδιού στον πίνακα occupy.
1. **Δημιουργήστε την παραπάνω βάση με κύρια και ξένα κλειδιά και γράψτε τις απαραίτητες εντολές INSERT INTO ώστε να καταχωρηθούν όλες οι εισαγωγές του ΠΕΤΡΟΥ σε νοσοκομεία.**

CREATE DATABASE test\_hospital;

USE test\_hospital;

CREATE TABLE Patient (P\_id int(5) NOT NULL, Pname varchar(20), Job varchar(15),

 PRIMARY KEY(P\_id));

INSERT INTO Patient (P\_id, Pname, Job)

 VALUES (10, 'ΣΠΥΡΟΥ', 'ΕΜΠΟΡΟΣ'),

 (20, 'ΙΩΑΝΝΟΥ', 'ΠΩΛΗΤΗΣ'),

 (30, 'ΠΕΤΡΟΥ', 'ΕΜΠΟΡΟΣ');

CREATE TABLE Hospital (H\_id int(3) NOT NULL, Hname varchar(20),

 PRIMARY KEY (H\_id));

 INSERT INTO Hospital VALUES

 (100, 'ΜΕΤΑΞΑ'), (200, 'ΑΡΕΤΑΙΕΙΟ');

CREATE TABLE Ward (W\_id int(4) NOT NULL, Wname varchar(20), H\_id int(3) NOT NULL, PRIMARY KEY(H\_id, W\_id), FOREIGN KEY(H\_id) REFERENCES Hospital (H\_id));

INSERT INTO Ward VALUES

 (1000, 'ΑΚΤΙΝΟΛΟΓΙΚΟ', 100),

 (2000, 'ΚΑΡΔΙΟΛΟΓΙΚΟ', 100),

 (2000, 'ΚΑΡΔΙΟΛΟΓΙΚΟ', 200),

 (1000, 'ΑΚΤΙΝΟΛΟΓΙΚΟ', 200);

CREATE TABLE Occupy (H\_id int(3) NOT NULL, W\_id int(4) NOT NULL,

 P\_id int(5) NOT NULL, Entry\_Date DATE NOT NULL,

 PRIMARY KEY(H\_id, W\_id, P\_id, Entry\_date),

 FOREIGN KEY(H\_id, W\_id)REFERENCES ward(H\_id, W\_id),

 FOREIGN KEY (P\_id)REFERENCES patient(P\_id));

INSERT INTO Occupy VALUES

 (100, 1000, 10, '2014/01/01'),

 (100, 1000, 20, '2014/01/01'),

 (100, 2000, 10, '2014/01/05'),

 (200, 1000, 10, '2014/01/29'),

 (200, 1000, 30, '2014/01/02');

SELECT \* FROM hospital;

SELECT \* FROM patient;

SELECT \* FROM ward;

SELECT \* FROM occupy;



1. **Αλλάξτε τον πίνακα Ward (δομή και περιεχόμενο)ώστε να έχει την παρακάτω μορφή:**

Ward (πίνακας κλινικών)

|  |  |  |  |
| --- | --- | --- | --- |
| **W\_id** | **Wname** | **H\_id** | **Beds** |
| **1000** | **ΑΚΤΙΝΟΛΟΓΙΚΟ** | **100** | **10** |
| **2000** | **ΚΑΡΔΙΟΛΟΓΙΚΟ** | **100** |  |
| **2000** | **ΚΑΡΔΙΟΛΟΓΙΚΟ** | **200** |  |

ALTER TABLE ward ADD beds int(2);

UPDATE ward

SET beds=10

WHERE w\_id=1000 AND h\_id=100;

SELECT \* FROM ward;



Θέμα 2

Έστω απλοποιημένη βάση δεδομένων Διεύθυνσης Προσωπικού. Οι στήλες των πινάκων είναι: Empno=Κωδικός υπαλλήλου, Name=όνομα, JobCode/JobNo=κωδικός θέσης, Job\_descr= θέση, Deptno=κωδικός τμήματος, Dname=τμήμα Sal=μισθός, Comm=προμήθεια. Υποτίθεται ότι κάθε υπάλληλος ανήκει σε ένα τμήμα και ότι ο μισθός του εξαρτάται από τη θέση του.

 Emp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Empno | Name | JobCode | DeptNo | Comm |
| 10 | ΣΠΥΡΟΥ | 100 | 50 | 450 |
| 20 | ΧΡΗΣΤΟΥ | 100 | 50 |  |
| 30 | ΝΙΚΟΥ | 300 | 60 |  |

 Job

|  |  |  |
| --- | --- | --- |
| JobNo | Job\_descr | Sal |
| 100 | ΠΩΛΗΤΗΣ | 2200 |
| 200 | ΑΝΑΛΥΤΗΣ | 2000 |
| 300 | ΧΕΙΡΙΣΤΗΣ | 1000 |

Dept

|  |  |  |
| --- | --- | --- |
| DeptNo | Dname | Loc |
| 50 | ΠΩΛΗΣΕΙΣ | ΑΘΗΝΑ |
| 60 | ΛΟΓΙΣΤΗΡΙΟ | ΑΘΗΝΑ |
| 70 | ΜΙΣΘΟΔΟΣΙΑ | ΒΟΛΟΣ |

CREATE DATABASE test\_personnel;

USE test\_personnel;

CREATE TABLE DEPT(DEPTNO INT(2) NOT NULL,

 DNAME VARCHAR(14), LOC VARCHAR(14),

 PRIMARY KEY(DEPTNO));

CREATE TABLE JOB (JOBNO INT(3) NOT NULL, JOB\_DESCR VARCHAR(15), SAL FLOAT(7,2),

 PRIMARY KEY(JOBNO));

CREATE TABLE EMP(EMPNO INT(4) NOT NULL,

 NAME VARCHAR(10), JOBCODE INT(3),

 DEPTNO INT(2), COMM FLOAT(7,2),

 PRIMARY KEY(EMPNO),

 FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO),

 FOREIGN KEY(JOBCODE) REFERENCES JOB(JOBNO));

DESCRIBE DEPT;

DESCRIBE JOB;

DESCRIBE EMP;

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (50, 'ΠΩΛΗΣΕΙΣ', 'ΑΘΗΝΑ');

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (60, 'ΛΟΓΙΣΤΗΡΙΟ', 'ΑΘΗΝΑ');

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (70, 'ΜΙΣΘΟΔΟΣΙΑ', 'ΒΟΛΟΣ');

INSERT INTO JOB VALUES (100, 'ΠΩΛΗΤΗΣ', 2200);

INSERT INTO JOB VALUES (200, 'ΑΝΑΛΥΤΗΣ', 2000);

INSERT INTO JOB VALUES (300, 'ΧΕΙΡΙΣΤΗΣ', 1000);

INSERT INTO EMP

 VALUES (10, 'ΣΠΥΡΟΥ', 100, 50, 450);

INSERT INTO EMP

 VALUES (20, 'ΧΡΗΣΤΟΥ', 100, 50, NULL);

INSERT INTO EMP

 VALUES (30, 'ΝΙΚΟΥ', 300, 60, NULL);

SELECT \* FROM emp;

SELECT \* FROM dept;

SELECT \* FROM job;



1. **Δείξτε (SELECT) name, empno, jobCode, job\_descr, sal, deptno, dname των υπαλλήλων που είναι αναλυτές ή πωλητές. Οι υπάλληλοι θα είναι ταξινομημένοι ανά θέση (job\_descr).**

SELECT name, empno, jobCode, job\_descr, sal, emp.deptno, dname

FROM emp, dept, job

WHERE emp.jobcode=job.jobno

AND emp.deptno=dept.deptno

AND job\_descr IN ('ΠΩΛΗΤΗΣ', 'ΑΝΑΛΥΤΗΣ')

ORDER BY job\_descr;



1. **Δείξτε (SELECT) name, empno, jobCode, job\_descr, sal, deptno, dname των υπαλλήλων που ανήκουν σε ένα από τα τμήματα: «ΠΩΛΗΣΕΙΣ» ή «ΛΟΓΙΣΤΗΡΙΟ» (με 2 τρόπους) (select … select, χρήση join)**

SELECT name, empno, jobCode, job\_descr, sal, emp.deptno, dname

FROM emp, dept, job

WHERE emp.jobcode=job.jobno

AND emp.deptno=dept.deptno

AND dname IN ('ΠΩΛΗΣΕΙΣ', 'ΛΟΓΙΣΤΗΡΙΟ')

ORDER BY job\_descr;

SELECT name, empno, jobCode, job\_descr, sal, emp.deptno, dname

FROM emp, dept, job

WHERE emp.jobcode=job.jobno

AND emp.deptno=dept.deptno

AND EMP.deptno IN (SELECT deptno FROM dept

 WHERE dname IN ('ΠΩΛΗΣΕΙΣ', 'ΛΟΓΙΣΤΗΡΙΟ'))

ORDER BY job\_descr;



1. **Δείξτε name, empno, job\_descr, sal των υπαλλήλων με σύνολο αμοιβών (άθροισμα μισθού και προμήθειας) μεγαλύτερο των 1000 ευρώ και μικρότερο των 3000 ευρώ.**

SELECT name, empno, job\_descr, sal, SAL+IFNULL(COMM,0)

FROM emp, job

WHERE emp.jobcode=job.jobno

AND SAL+IFNULL(COMM,0)>1000

AND SAL+IFNULL(COMM,0)<3000

ORDER BY job\_descr;

* **Προσοχή! Η επόμενη εντολή δεν υπολογίζει το ζητούμενο.**

SELECT name, empno, job\_descr, sal, SAL+IFNULL(COMM,0)

FROM emp, job

WHERE emp.jobcode=job.jobno

AND SAL+IFNULL(COMM,0) BETWEEN 1000 AND 3000

ORDER BY job\_descr;

* **Επιπλέον θυμηθείτε ότι υποχρεωτικά γράφουμε SAL+IFNULL(COMM,0)**

 **Και όχι SAL+COMM.**



1. **Διορθώστε και συμπληρώστε την παρακάτω εντολή SELECT ώστε να δείχνει στοιχεία πωλητών και αναλυτών που έχουν μισθό μεγαλύτερο των 1500 ευρώ. Πρέπει να έχουμε δύο επίπεδα ταξινόμησης: Οι υπάλληλοι θα τυπώνονται ανά θέση δηλαδή πρώτα οι αναλυτές, μετά οι πωλητές και οι υπαλληλοι που έχουν την ίδια θέση θα εμφανίζονται αλφαβητικά.**

SELECT empno, name, job, sal, comm, deptno, dname

FROM

WHERE

ORDER BY

SELECT name, empno, jobCode, job\_descr, sal, emp.deptno, dname

FROM emp, dept, job

WHERE emp.jobcode=job.jobno

AND emp.deptno=dept.deptno

AND job\_descr IN ('ΠΩΛΗΤΗΣ', 'ΑΝΑΛΥΤΗΣ')

AND sal>1500

ORDER BY job\_descr, name;



1. **Γράψτε εντολή SELECT η οποία υπολογίζει πόσοι υπάλληλοι έχουν κωδικό θέσης 100 (είναι πωλητές), πόσοι 300 (είναι χειριστές) κ.λπ. Να πως θα φαίνονται τα αποτελέσματα.**

|  |  |
| --- | --- |
| **JobCode** | **No\_of\_employees.** |
| **100** | **2** |
| **300** | **1** |

SELECT jobcode, COUNT(\*)

FROM emp

GROUP BY jobcode

ORDER BY jobcode;

**Προσοχή! Αν χρησιμοποιώ GROUP BY έχω περιορισμούς στο τι επιλέγω στο SELECT!**

1. **Τροποποιήστε την εντολή SELECT έτσι ώστε αν και υπολογίζει πόσοι υπάλληλοι έχουν κωδικό θέσης 100 (είναι αναλυτές), πόσοι 300 (είναι χειριστές) κ.λπ. να δείχνει μόνο θέσεις που απασχολούν περισσότερους από έναν υπαλλήλους:**

|  |  |
| --- | --- |
| **JobCode** | **No\_of\_employees** |
| **100** | **2** |

SELECT jobcode, COUNT(\*)

FROM emp

GROUP BY jobcode

HAVING COUNT(\*)>1

ORDER BY jobcode;

1. **Δείξτε τους υπάλληλους με έδρα 'ΑΘΗΝΑ' ή 'ΒΟΛΟΣ' και θέση σαν του ΣΠΥΡΟΥ.**

**SELECT name, empno, jobCode, job\_descr, sal, emp.deptno, dname, loc**

**FROM emp, dept, job**

**WHERE emp.jobcode=job.jobno**

**AND emp.deptno=dept.deptno**

**AND loc IN ('ΑΘΗΝΑ', 'ΒΟΛΟΣ')**

**AND jobcode=(SELECT jobcode FROM emp WHERE name='ΣΠΥΡΟΥ')**

**ORDER BY job\_descr;**



1. **Διαφέρουν οι παρακάτω αριθμοί AVG(comm), SUM(comm)/COUNT(comm), SUM(comm)/COUNT(\*) για τους υπάλληλους του πίνακα emp;**

SELECT AVG(comm), SUM(comm)/COUNT(comm), SUM(comm)/COUNT(\*)

FROM EMP;



1. **Εμφάνισε τον εργαζόμενο που έχει τον μεγαλύτερο μισθό στο τμήμα 'ΠΩΛΗΣΕΙΣ'**

SELECT name, empno, jobCode, job\_descr, sal, emp.deptno, dname, loc

FROM emp, dept, job

WHERE emp.jobcode=job.jobno

AND emp.deptno=dept.deptno

AND sal= (SELECT MAX(sal)

 FROM emp, job

 WHERE emp.jobcode=job.jobno

 AND deptno IN (SELECT deptno FROM dept WHERE dname='ΠΩΛΗΣΕΙΣ'))

ORDER BY job\_descr;

**Λύνεται και χωρίς την υποπρόταση deptno IN (SELECT deptno FROM dept WHERE dname='ΠΩΛΗΣΕΙΣ'). Απλά χρησιμοποιούμε τον κωδικό για dname='ΠΩΛΗΣΕΙΣ' ή σύνδεση**.



1. **Δείξτε υπάλληλους που το όνομά τους αρχίζει με γράμμα Ν ή Σ.**

SELECT \*

FROM EMP

WHERE SUBSTR(name,1, 1) IN ('Ν','Σ');



|  |
| --- |
| **Ανοικτά Ακαδημαϊκά Μαθήματα****Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας** |
| **Τέλος Ενότητας** |
| **Χρηματοδότηση*** Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Αθήνας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

 |

**Σημειώματα**

**Σημείωμα Αναφοράς**

Copyright ΤΕΙ Αθήνας, Χ. Σκουρλάς, Α. Τσολακίδης 2014. Χ. Σκουρλάς, Α. Τσολακίδης. «Βάσεις Δεδομένων Ι. Ενότητα 16: Εξετάσεις – Θέματα και κάποιες ενδεικτικές λύσεις». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: [ocp.teiath.gr](https://ocp.teiath.gr/).

**Σημείωμα Αδειοδότησης**

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

* που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
* που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
* που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

**Διατήρηση Σημειωμάτων**

* Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
* το Σημείωμα Αναφοράς
* το Σημείωμα Αδειοδότησης
* τη δήλωση Διατήρησης Σημειωμάτων
* το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

**Σημείωμα Χρήσης Έργων Τρίτων**

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων: