

**Ανοικτά Ακαδημαϊκά Μαθήματα**

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας**

Βάσεις Δεδομένων

**Database Tour One:** Δημιουργία και διαχείριση βάσης προσωπικού με χρήση απλών και σύνθετων δηλώσεων (statements) SQL στο προϊόν mySQL. Έμφαση σε θέματα αναζήτησης (queries)

Χ. Σκουρλάς

Τμήμα Μηχανικών Πληροφορικής ΤΕ

Στόχος του εργαστηρίου είναι η εκμάθηση απλών και σύνθετων δηλώσεων (statements) SQL με χρήση του προϊόντος mySQL. Μετά την επεξεργασία του εργαστηρίου ο ενδιαφερόμενος θα γνωρίζει σύνταξη και χρήση δηλώσεων της γλώσσας SQL και θα εφαρμόζει σειρά πρακτικών κανόνων για τη διαχείριση βάσης δεδομένων με έμφαση σε θέματα αναζήτησης (queries).

**Λέξεις Κλειδιά**: SQL, mySQL, CREATE TABLE, INSERT INTO, SELECT

|  |  |
| --- | --- |
| 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. |

# Δημιουργία βάσης

**Βάση δεδομένων που θα χρησιμοποιηθεί στα παραδείγματα αναζήτησης.**

 (πίνακας στοιχείων υπαλλήλου) EMP

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |

(πίνακας στοιχείων Τμημάτων στα οποία ανήκουν οι υπάλληλοι)DEPT

|  |  |  |
| --- | --- | --- |
| **DEPTNO** | **DNAME** | **LOC** |

(πίνακας στοιχείων έργων της εταιρείας) PROJ

|  |  |  |
| --- | --- | --- |
| **PROJNO** | **PNAME** | **BUDGET** |

(πίνακας απασχόλησης υπαλλήλων σε έργα της εταιρείας) ASSIGN

|  |  |  |
| --- | --- | --- |
| **EMPNO** | **PROJNO** | **PTIME** |

Όπου EMPNO=κωδικός υπαλλήλου, ENAME=ονοματεπώνυμο υπαλλήλου, JOB = θέση στην εταιρεία, MGR = ο επικεφαλής του, SAL= μισθός, COMM = προμήθεια, DEPTNO = κωδικός Τμήματος. Υποτίθεται ότι κάθε υπάλληλος ανήκει σε ένα Τμήμα, DNAME = όνομα Τμήματος, LOC = έδρα Τμήματος, PROJNO = κωδικός έργου, PNAME = περιγραφή έργου, BUDGET = προϋπολογισμός έργου, PTIME = ποσοστό χρόνου απασχόλησης υπαλλήλου σε έργο. Υποτίθεται ότι κάθε υπάλληλος μπορεί να εργάζεται σε περισσότερα από ένα έργα.

**Μοντέλο οντοτήτων συσχετίσεων με συμβολισμό Navathe-Elmasri**

****

****

SELECT \* FROM dept

|  |  |  |
| --- | --- | --- |
| **DEPTNO** | **DNAME** | **LOC** |
| 10 | ACCOUNTING | NEW YORK  |
| 20 | RESEARCH | DALLAS  |
| 30 | SALES | CHICAGO  |
| 40 | OPERATIONS | BOSTON  |

SELECT \* FROM emp

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 10 | CODD | ANALYST | 15 | 01/01/1989 | 3000 | -  | 10 |
| 15 | ELMASRI | ANALYST | 15 | 02/05/1995 | 1200 | 150 | 10 |
| 20 | NAVATHE | SALESMAN | 20 | 07/07/1977 | 2000 | -  | 20 |
| 30 | DATE | PROGRAMMER | 15 | 04/05/2004 | 1800 | 200 | 10 |

SELECT \* FROM proj

|  |  |  |
| --- | --- | --- |
| **PROJNO** | **PNAME** | **BUDGET** |
| 100 | PAYROLL | 100000 |
| 200 | PERSONNEL | 200000 |
| 300 | SALES | 150000 |

SELECT \* FROM assign

|  |  |  |
| --- | --- | --- |
| **EMPNO** | **PROJNO** | **PTIME** |
| 10 | 100 | 40 |
| 10 | 200 | 60 |
| 15 | 100 | 100 |
| 20 | 200 | 100 |
| 30 | 100 | 100 |

1. **Πώς δημιουργούμε τη βάση δεδομένων με τη γλώσσα sql.**

**Δημιουργία της βάσης με mySQL. Οι πίνακες δημιουργούνται με κύρια και ξένα κλειδιά.**

**Αν θέλουμε να δούμε τις υπαρχουσες βάσεις γράφουμε εντολή SHOW.**

SHOW DATABASES;

**Δημιουργία της βάσης**

CREATE DATABASE DATABASE\_TOUR\_ONE;

**Χρήση βάσης**

USE DATABASE\_TOUR\_ONE;

**Δημιουργία πινάκων**

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

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

 PRIMARY KEY(DEPTNO));

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

 ENAME VARCHAR(10), JOB VARCHAR(25),

 HIREDATE DATE, MGR INT(4), SAL FLOAT(7,2), COMM FLOAT(7,2),

 DEPTNO INT(2),

 PRIMARY KEY(EMPNO),

 FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO));

CREATE TABLE PROJ (projno INT(3) NOT **NULL**,
                   pname VARCHAR(15),
                   budget FLOAT(12,2),

 PRIMARY KEY(projno));
CREATE TABLE ASSIGN(

 EMPNO INT(4) NOT NULL, PROJNO INT(3) NOT **NULL**,

 PTIME INT(3),

PRIMARY KEY(EMPNO,PROJNO),

FOREIGN KEY(EMPNO) REFERENCES EMP(EMPNO),

FOREIGN KEY(PROJNO) REFERENCES PROJ(PROJNO));

SHOW TABLES;

**Αν θέλουμε να δούμε τη δομή των πινάκων γράφουμε εντολή describe.**

**describe emp;**

**Εισαγωγή στοιχείων στους πίνακες**

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (20, 'RESEARCH', 'DALLAS');

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO DEPT(DEPTNO, DNAME, LOC)

 VALUES (40, 'OPERATIONS', 'BOSTON');

INSERT INTO EMP

 VALUES (10, 'CODD', 'ANALYST', '1989/01/01', 15, 3000, NULL, 10);

INSERT INTO EMP

 VALUES (15, 'ELMASRI', 'ANALYST', '1995/05/02', 15, 1200, 150, 10);

INSERT INTO EMP

 VALUES (20, 'NAVATHE', 'SALESMAN', '1977/07/07', 20, 2000, NULL, 20);

INSERT INTO EMP

 VALUES (30, 'DATE', 'PROGRAMMER', '2004/05/04', 15, 1800, 200, 10);

INSERT INTO proj(projno, pname, budget)
            VALUES(100, 'PAYROLL', 100000);
INSERT INTO proj(projno, pname, budget)
            VALUES(200, 'PERSONNEL',200000 );
INSERT INTO proj(projno, pname, budget)
            VALUES(300, 'SALES', 150000);
INSERT INTO assign(empno, projno, ptime)
                 VALUES(10,100, 40);
INSERT INTO assign(empno, projno, ptime)
                 VALUES(10, 200, 60);
INSERT INTO assign(empno, projno, ptime)
                 VALUES(15, 100, 100);
INSERT INTO assign(empno, projno, ptime)
                 VALUES(20, 200, 100);
INSERT INTO assign(empno, projno, ptime)
                 VALUES(30, 100, 100);

**Πως βλέπουμε τα δεδομένα.**

SELECT \* FROM DEPT;

SELECT \* FROM EMP;

SELECT \* FROM PROJ;

SELECT \* FROM ASSIGN;

**Διαγραφή των πινάκων**

DROP TABLE assign;

DROP TABLE emp;

DROP TABLE proj;

DROP TABLE dept;

**Η διαγραφή της βάσης γίνεται με την εντολή:**

DROP DATABASE DATABASE\_TOUR\_ONE;

# Πώς αναζητούμε στοιχεία με τη γλώσσα sql.

#1 Δείξτε υπαλλήλους της εταιρείας με το μισθό τους.

SELECT \* FROM EMP;

SELECT EMPNO "ΚΩΔΙΚΟΣ ΥΠΑΛΛΗΛΟΥ", ENAME "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", SAL "ΜΙΣΘΟΣ"

FROM EMP;

SELECT EMPNO "ΚΩΔΙΚΟΣ ΥΠΑΛΛΗΛΟΥ", ENAME "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", SAL "ΜΙΣΘΟΣ"

FROM EMP;

#2 Δείξτε υπαλλήλους της εταιρείας με το μισθό τους, την ωριαία αμοιβή και το σύνολο # # # αποδοχών.

SELECT EMPNO, ENAME, SAL/(25\*8), SAL+IFNULL(COMM,0) FROM EMP;

SELECT EMPNO, ENAME, SAL/(25\*8) "ΩΡΙΑΙΑ ΑΜΟΙΒΗ",

 SAL+IFNULL(COMM,0) "ΣΥΝΟΛΟ ΑΠΟΔΟΧΩΝ"

FROM EMP;

#3 Δείξτε όνομα και μισθό υπαλλήλου με τίτλους στο αποτέλεσμα "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", # # "ΜΙΣΘΟΣ". Ταξινομήστε τα αποτελέσματα κατά όνομα

SELECT EMPNO, ENAME "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", SAL "ΜΙΣΘΟΣ"

FROM EMP

ORDER BY ENAME;

# Δείτε και τα εξής

SELECT EMPNO, ENAME, SAL, DEPTNO

FROM EMP

ORDER BY JOB, ENAME;

SELECT EMPNO, ENAME, SAL, DEPTNO

FROM EMP

ORDER BY DEPTNO, ENAME;

#4 Δείξτε μαζί κωδικό και όνομα υπαλλήλου

select CONCAT(empno, ' ', ename) from emp;

#5 Που διαφέρουν οι δηλώσεις;

SELECT job

from emp;

SELECT DISTINCT job

from emp;

#6 Τι δείχνει η δήλωση;

SELECT DISTINCT job, deptno

from emp;

#7 Τι δείχνει η δήλωση;

SELECT DISTINCT job, deptno

from emp

ORDER BY job, deptno;

#8 Τι δείχνει η δήλωση;

SELECT DISTINCT job, deptno

from emp

ORDER BY deptno, job;

#9 Δείξε τα τμήματα που έχουν έδρα NEW YORK.

SELECT DNAME, LOC

FROM DEPT

WHERE LOC = 'NEW YORK';

#10 Δείξε τους υπάλληλους του Τμήματος 10.

SELECT \*

FROM EMP

WHERE DEPTNO = 10;

#11 Δείξε κωδικό, όνομα, θέση υπαλλήλου και κωδικό τμήματος για τους αναλυτές # # # # # # # (ANALYST).

SELECT ENAME, EMPNO, JOB, DEPTNO

FROM EMP

WHERE JOB = 'ANALYST';

#12 Δείξε όνομα, αριθμό Τμήματος για τμήματα με κωδικό (αριθμό) μεγαλύτερο του 20.

SELECT DNAME, DEPTNO

FROM DEPT

WHERE DEPTNO > 20;

#13 Δείξε όνομα, αριθμό Τμήματος για τμήματα με κωδικό >= 20.

SELECT DNAME,DEPTNO

FROM DEPT

WHERE DEPTNO >= 20;

#14 Δείξε υπαλλήλους με προμήθεια μεγαλύτερη του ενός δεκάτου του μισθού τους.

SELECT ENAME,SAL,SAL/10, COMM

FROM EMP

WHERE COMM > SAL/10;

# Χρήση τελεστών σύγκρισης

#15 Δείξτε πωλητές με μισθό πάνω από 1,300 ή προμήθεια μεγαλύτερη του ενός δεκάτου του μισθού τους

# Λανθασμένη λύση

SELECT ENAME "ΟΝΟΜΑ", SAL "ΜΙΣΘΟΣ", COMM "ΠΡΟΜΗΘΕΙΑ"

FROM EMP

WHERE JOB = 'SALESMAN'

 AND (SAL> 1300) OR (COMM > SAL/10);

# ορθή λύση

SELECT ENAME "ΟΝΟΜΑ", SAL "ΜΙΣΘΟΣ", COMM "ΠΡΟΜΗΘΕΙΑ"

FROM EMP

WHERE JOB = 'SALESMAN'

 AND ((SAL> 1300) OR (COMM > SAL/10));

# Χρήση BETWEEN … AND

#16 Δείξτε πωλητές που έχουν μισθό μεταξύ 1800 και 2000.

SELECT ENAME, JOB, SAL

FROM EMP

WHERE JOB = 'SALESMAN'

AND SAL BETWEEN 1800 AND 2000;

# Εναλλακτική λύση

SELECT ENAME, JOB, SAL

FROM EMP

WHERE JOB = 'SALESMAN'

AND (SAL>=1800 AND SAL<=2000);

# Χρήση συνόλων (τελεστών IN)

#17 Δείξτε πωλητές ή αναλυτές που έχουν μισθό μεγαλύτερο από 1250

SELECT ENAME, JOB, SAL

FROM EMP

WHERE JOB IN ('SALESMAN', 'ANALYST')

AND SAL> 1250;

# Εναλλακτική λύση

SELECT ENAME, JOB, SAL

FROM EMP

WHERE (JOB='SALESMAN' OR JOB='ANALYST')

AND SAL> 1250;

# Χρήση συνάρτησης substr

#18 Δείξτε αναλυτές με όνομα που αρχίζει από C, D, E.

SELECT ename, job FROM emp

WHERE (ename >= 'C' AND ename < 'F' )

AND job = 'ANALYST';

# χρήση LIKE

#19 Δείξτε υπάλληλους με επώνυμο που αρχίζει από το γράμμα C

SELECT ename, job

FROM emp

WHERE ename LIKE 'C%';

# Χρήση BETWEEN … AND με ημερομηνίες

#20 Δείξτε υπάλληλους που το επώνυμο τους αρχίζει από C ή D και προσλήφθηκαν από το 1989 έως και το 1995

SELECT ename, hiredate from emp

where ename between 'C' and 'F'

and hiredate Between '1989/01/01' and '1995/12/31';

#21 Δείξτε όνομα,θέση και μισθό των υπαλλήλων του Τμήματος 20 που κερδίζουν πάνω από 2,000.

SELECT ENAME,JOB,SAL, DEPTNO

FROM EMP

WHERE DEPTNO = 10

AND SAL > 2000;

#22 Δείξτε τους πωλητές της Τμήματος 20 με μισθό μεγαλύτερο ή ισο των 1,500.

SELECT ENAME, SAL, JOB, DEPTNO FROM EMP

WHERE JOB = 'ANALYST'

 AND DEPTNO = 10

 AND SAL >= 1500;

#23 Δείξτε υπάλληλους που είναι πωλητές ή αναλυτές (με 2 τρόπους, IN, χρήση OR)

# Ταξινομήστε ανά θέση κατά φθίνουσα σειρά

SELECT ENAME, JOB, SAL, DEPTNO

FROM EMP

WHERE JOB = 'ANALYST'

OR JOB = 'SALESMAN'

ORDER BY JOB DESC;

SELECT ENAME, JOB, SAL, DEPTNO

FROM EMP

WHERE JOB IN('ANALYST', 'SALESMAN')

ORDER BY JOB DESC;

#24 Δείξτε τους τους πωλητές ή αναλυτές που ανήκουν στο τμήμα 10.

SELECT ENAME, JOB, SAL, DEPTNO

FROM EMP

WHERE (JOB = 'ANALYST' OR JOB = 'SALESMAN')

AND DEPTNO=10;

#25 Δείξτε τους πωλητές ή αναλυτές που δεν ανήκουν στο τμήμα 10.

SELECT EMPNO, ENAME, JOB, SAL, DEPTNO

FROM EMP

WHERE (JOB = 'ANALYST' OR JOB = 'SALESMAN')

AND DEPTNO <> 10;

#26 Δείξτε τους πωλητές οποιουδήποτε τμήματος και τους αναλυτές του Τμήματος 10.

SELECT \*

FROM EMP

WHERE JOB = 'SALESMAN'

OR (JOB = 'ANALYST' AND DEPTNO = 10);

#27 Δείξτε τους υπάλληλους του Τμήματος 10 που δεν είναι πωλητές ή αναλυτές.

SELECT \*

FROM EMP

WHERE NOT (JOB = 'SALESMAN' OR JOB = 'ANALYST')

AND DEPTNO = 10;

#28 Δείξτε τους υπαλλήλους που έχουν μισθό μεταξύ 1,500 και 1,800 ταξινομημένους ανά # # όνομα.

SELECT ENAME, JOB, SAL

FROM EMP

WHERE SAL BETWEEN 1500 AND 1800

ORDER BY ENAME;

#29 Χρησιμοποιήστε τελεστή BETWEEN ... AND και τελεστή NOT για να βρείς τους υπόλοιπους # # # υπάλληλους.

SELECT ENAME,JOB,SAL

FROM EMP

WHERE SAL NOT BETWEEN 1500 AND 1800

ORDER BY ENAME;

#30 Δείξτε υπάλληλους που είναι αναλυτές ή πωλητές ή DBA.

SELECT ENAME,JOB,DEPTNO

FROM EMP

WHERE JOB IN ('ANALYST','SALESMAN', 'DBA');

#31 Δείξτε τους υπόλοιπους υπάλληλους.

SELECT ENAME,JOB,DEPTNO

FROM EMP

WHERE JOB NOT IN ('ANALYST','SALESMAN', 'DBA');

#32 Δείξτε τους υπάλληλους που το επώνυμο τους αρχίζει με γράμμα C ή E.

SELECT ENAME,JOB,DEPTNO

FROM EMP

WHERE (ENAME LIKE 'C%' OR ENAME LIKE 'E%');

#33 Δείξτε τους υπάλληλους που το όνομα τους τελειώνει σε THE και είναι Πωλητές.

SELECT ENAME, JOB, DEPTNO

FROM EMP

WHERE ENAME LIKE '%THE'

AND JOB = 'SALESMAN';

|  |
| --- |
| **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 τμήματος 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 2017. Χ. Σκουρλάς. «Βάσεις Δεδομένων και Web Development. Ενότητα: Database Tour One». Έκδοση: 1.0. Αθήνα 2017. Διαθέσιμο από τη δικτυακή διεύθυνση: [pyles.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: