

**Πανεπιστήμιο Δυτικής Αττικής**

Βάσεις Δεδομένων Ι

**Ενότητα 4:** Δημιουργία και διαχείριση βάσης προσωπικού με χρήση απλών και σύνθετων δηλώσεων (statements) SQL στα προϊόντα mySQL και Oracle. Έμφαση σε θέματα αναζήτησης (queries)

Χ. Σκουρλάς

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Περιεχόμενα

[1. Δημιουργία βάσης 3](#_Toc528687865)

[2. Πώς αναζητούμε στοιχεία με τη γλώσσα sql. Σύνταξη της δήλωσης select. Πρακτικοί κανόνες 6](#_Toc528687866)

[2.1 Πως βλέπουμε όλα τα στοιχεία ενός πίνακα. 6](#_Toc528687867)

[2.2 Πώς αλλάζουμε τη σειρά των στηλών των αποτελεσμάτων. 6](#_Toc528687868)

[2.3 Πώς αλλάζουμε τα ονόματα των στηλών των αποτελεσμάτων. 6](#_Toc528687869)

[2.4 Αναζητήσεις που οδηγούν σε αποτελέσματα χωρίς επανάληψη τιμών - τελεστής DISTINCT 7](#_Toc528687870)

[2.5 Πως περιορίζεις τις στήλες των αποτελεσμάτων με απλές συνθήκες - Υποπρόταση WHERE. 7](#_Toc528687871)

[2.6 Αναζητήσεις που περιλαμβάνουν υπολογισμούς, πράξεις σε strings κλπ. 8](#_Toc528687872)

[2.7 Πως σχηματίζουμε σύνθετες συνθήκες σε υποπρόταση WHERE. Τελεστές σύγκρισης, Aριθμητικοί, Boole, LIKE, NOT LIKE, BETWEEN ...AND, NOT BETWEEN ...AND, σύνολα (ΙΝ). 10](#_Toc528687873)

[Παράδειγμα 1: Χρήση τελεστών σύγκρισης 10](#_Toc528687874)

[Παράδειγμα 2: Χρήση BETWEEN … AND 10](#_Toc528687875)

[Παράδειγμα 3: Χρήση συνόλων (τελεστών IN) 10](#_Toc528687876)

[Παράδειγμα 4: Χρήση συνάρτησης substr 11](#_Toc528687877)

[Παράδειγμα 5: Χρήση BETWEEN … AND με ημερομηνίες 12](#_Toc528687878)

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, Oracle, CREATE TABLE, INSERT INTO, SELECT

# 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 στις οποίες 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.

**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 new\_personnel;

**Χρήση βάσης**

USE new\_personnel;

**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 NEW\_PERSONNEL;

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 |

# Πώς αναζητούμε στοιχεία με τη γλώσσα sql. Σύνταξη της δήλωσης select. Πρακτικοί κανόνες

## Πως βλέπουμε όλα τα στοιχεία ενός πίνακα.

Η σύνταξη της εντολής είναι:

SELECT \* FROM όνομα-πίνακα;

***Παράδειγμα***

Η παρακάτω εντολή δείχνει τα στοιχεία του πίνακα DEPT.

SELECT \* FROM dept;

## Πώς αλλάζουμε τη σειρά των στηλών των αποτελεσμάτων.

Η συνηθισμένη σύνταξη της εντολής είναι:

SELECT όνομα-πίνακα.όνομα-στήλης, ...

FROM όνομα-πίνακα;

***Παράδειγμα***

Η παρακάτω εντολή δείχνει τις στήλες DNAME, LOC :

***Προσοχή!*** Αν είναι σαφές σε ποιές στήλες αναφερόμαστε χωρίς να χρειάζεται να εξετάζουμε το όνομα του πίνακα μπορούμε να το παραλείψουμε.

Έτσι η παραπάνω αναζήτηση μπορεί να τροποποιηθεί:

SELECT DNAME, LOC FROM DEPT;

## Πώς αλλάζουμε τα ονόματα των στηλών των αποτελεσμάτων.

Για να αλλάξουμε το όνομα μιας στήλης στα αποτελέσματα αρκεί να γράψουμε σε εισαγωγικά το νέο όνομα δίπλα στο όνομα της στήλης.

***Παράδειγμα***

Η παρακάτω εντολή δείχνει τους υπαλλήλους της εταιρείας με το μισθό τους.

SELECT ENAME "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", SAL "ΜΙΣΘΟΣ"

FROM EMP;

***Σημείωση 1:*** Οπως θα δούμε στη συνέχεια η SQL επιτρέπει εντολές της μορφής

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

SAL+NVL(COMM,0) "ΣΥΝΟΛΟ ΑΠΟΔΟΧΩΝ"

FROM EMP;

**Σε mySQL:**

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

SAL+IFNULL(COMM,0) "ΣΥΝΟΛΟ ΑΠΟΔΟΧΩΝ"

FROM EMP;

***Σημείωση 2:*** Αν το όνομα που δίνουμε είναι μόνο μια λέξη δεν είναι απαραίτητο να το γράφουμε σε εισαγωγικά.

SELECT ENAME "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", SAL ΜΙΣΘΟΣ

FROM EMP;

**Σε mySQL:**

SELECT ENAME "ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ", SAL "ΜΙΣΘΟΣ"

FROM EMP;

***Σημείωση 3:*** *Concatenation*

SELECT empno||ename FROM emp; -- oracle

**MySQL: select CONCAT(empno, ' ', ename) from emp;**

***Σημείωση 4:*** *Ανάλογα με το προϊόν θα χρησιμοποιήσετε απλά (*'*…*') *ή διπλά (“ ...”) εισαγωγικά*

## Αναζητήσεις που οδηγούν σε αποτελέσματα χωρίς επανάληψη τιμών - τελεστής DISTINCT

Η συνηθισμένη σύνταξη της εντολής είναι:

SELECT DISTINCT όνομα-πίνακα.όνομα-στήλης,

όνομα-πίνακα.όνομα-στήλης, ...

FROM όνομα-πίνακα κλπ.

***Παράδειγμα***

Η παρακάτω αναζήτηση προσδιορίζει τα επαγγέλματα των υπαλλήλων επαναλαμβάνοντας τις τιμές στα αποτελέσματα:

SELECT JOB FROM EMP;

Αντίθετα η αναζήτηση

SELECT DISTINCT JOB FROM EMP;

δεν τις επαναλαμβάνει, δηλαδή, αναγράφει όλα τα επαγγέλματα που υπάρχουν στο πίνακα EMP και μάλιστα μόνο μία φορά το καθένα.

***Παράδειγμα***

Παρατηρήστε τη διαφορά και στις επόμενες εντολές:

SELECT job, deptno

from emp;

SELECT DISTINCT job, deptno

from emp;

***Προσοχή!*** Η εντολή SELECT DISTINCT job, deptno from emp; θα εμφανίσει 3 στήλες επειδή το ζεύγος (ANALYST, 10) εμφανίζεται 1 φορά.

***Σημείωση:*** Ο τελεστής DISTINCT μπορεί να χρησιμοποιηθεί και με συναρτήσεις και γενικά σε σύνθετες αναζητήσεις.

## Πως περιορίζεις τις στήλες των αποτελεσμάτων με απλές συνθήκες - Υποπρόταση WHERE.

Η συνηθισμένη σύνταξη της εντολής είναι:

SELECT ...

FROM ...

WHERE συνθήκη;

***Παράδειγμα***

Η παρακάτω αναζήτηση βρίσκει ποια τμήματα έχουν έδρα NEW YORK.

SELECT DNAME "ΟΝΟΜΑ ΤΜΗΜΑΤΟΣ", LOC ΕΔΡΑ

FROM DEPT

WHERE LOC = 'NEW YORK';

**MySQL:**

SELECT DNAME "ΟΝΟΜΑ ΤΜΗΜΑΤΟΣ", LOC "ΕΔΡΑ"

FROM DEPT

WHERE LOC = 'NEW YORK';

***Παράδειγμα*** Διάλεξε μόνο τους υπάλληλους της Τμήματος 10.

SELECT \*

FROM EMP

WHERE DEPTNO = 10;

***Παράδειγμα*** Δείξε ονόματα,αριθμούς και Διευθύνσεις για τους αναλυτές.

SELECT ENAME,EMPNO,JOB, DEPTNO

FROM EMP

WHERE JOB = 'ANALYST';

***Παράδειγμα***  Δείξε ονόματα και αριθμό Τμημάτων για τις Διευθύνσεις με

κωδικό (αριθμό) μεγαλύτερο του 20.

SELECT DNAME,DEPTNO

FROM DEPT

WHERE DEPTNO > 20;

***Παράδειγμα*** Βρες ονόματα και αριθμούς Τμημάτων με κωδικό >= 20.

SELECT DNAME,DEPTNO

FROM DEPT

WHERE DEPTNO >= 20;

***Παράδειγμα*** Βρες υπαλλήλους με προμήθεια μεγαλύτερη του δέκατου του μισθού τους.

SELECT ENAME,SAL,SAL/10, COMM

FROM EMP

WHERE COMM > SAL/10;

## Αναζητήσεις που περιλαμβάνουν υπολογισμούς, πράξεις σε strings κλπ.

Για να υπολογίσεις αθροίσματα, γινόμενα, μέσους όρους κλπ. μπορείς να χρησιμοποιήσεις κατά την επιλογή στηλών:

1. αριθμητικούς τελεστές (+, -, \*, /),
2. συναρτήσεις (όπως NVL-Null Value, POWER-ύψωση σε δύναμη,ROUND-στρογγύλευση, TRUNC-αποκοπή, SUBSTR-substring, UPPER-μεταγραφή σε κεφαλαία, AVG-μέσος όρος κλπ),
3. παρενθέσεις,
4. τελεστή DISTINCT.

***Ανάλογα με το προϊόν οι συναρτήσεις μπορεί να διαφέρουν. Για παράδειγμα χρησιμοποιούμε τη συνάρτηση NVL στην oracle ενώ τη συνάρτηση IFNULL στη mySQL.***

Βλέπουμε τη σημερινή ημερομηνία με τη συνάρτηση sysdate χρησιμοποιώντας τον μονοθέσιο πίνακα dual του συστήματος.

SELECT sysdate FROM dual; -- oracle

**mySQL:**

SELECT current\_date;

Σε διάφορα προϊόντα, όπως για παράδειγμα στο προϊόν της **ORACLE**, είναι δυνατές κάποιες αριθμητικές πράξεις σε ημερομηνίες. Για παράδειγμα, η αναζήτηση

SELECT sysdate + 5 FROM dual;

προσθέτει πέντε ημέρες στην σημερινή ημερομηνία ενώ η αναζήτηση

mySQL: SELECT CURRENT\_DATE + 5;

SELECT sysdate - 5 FROM dual αφαιρεί 5 ημέρες.

mySQL: SELECT CURRENT\_DATE - 5;

Τέλος, η αναζήτηση

SELECT sysdate - hiredate, ename FROM emp;

εμφανίζει τον αριθμό των ετών, των μηνών, και των ημερών μεταξύ των δύο ημερομηνιών.

**MySQL: SELECT CURRENT\_DATE - hiredate, ename FROM emp;**

*Μια αντίστοιχη χρήσιμη συνάρτηση στη mySQL είναι η datediff.*

***Παράδειγμα***

Όπως είδαμε και στα προηγούμενα η παρακάτω αναζήτηση βρίσκει την ωριαία και τη συνολική αμοιβή υπαλλήλων

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

SAL+NVL (COMM,0) "ΣΥΝΟΛΙΚΗ ΑΜΟΙΒΗ"

FROM EMP;

Η συνάρτηση NVL εξασφαλίζει ότι το άθροισμα "Μισθός+Προμήθεια" έχει τιμή στην περίπτωση υπαλλήλων που δεν έχουν προμήθεια.

**MySQL:**

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

SAL+IFNULL(COMM,0) "ΣΥΝΟΛΙΚΗ ΑΜΟΙΒΗ"

FROM EMP;

Δείτε τι θα συμβεί αν δεν χρησιμοποιήσουμε τη συνάρτηση.

SELECT ENAME ΟΝΟΜΑ,

SAL/(25\*8) "ΩΡΙΑΙΑ ΑΜΟΙΒΗ",SAL+COMM,

SAL+NVL (COMM,0) "ΣΥΝΟΛΙΚΗ ΑΜΟΙΒΗ"

FROM EMP;

## Πως σχηματίζουμε σύνθετες συνθήκες σε υποπρόταση WHERE. Τελεστές σύγκρισης, Aριθμητικοί, Boole, LIKE, NOT LIKE, BETWEEN ...AND, NOT BETWEEN ...AND, σύνολα (ΙΝ).

Για να σχηματίσουμε συνθήκες μπορούμε να χρησιμοποιήσουμε:

**1)** τελεστές σύγκρισης (>, <, >=, <=, != , <>,**^** =), όπου οι τρεις τελευταίοι (ή και άλλοι ανάλογα με το προϊόν) συμβολίζουν το διάφορο.

**2)** αριθμητικούς (+, -, /, \*),

**3)** τελεστές Boole (AND, OR, NOT),

**4)** τελεστή LIKE ή NOT LIKE (π.χ. ENAME NOT LIKE '%ΑΣ'),

**5)** τελεστές BETWEEN ...AND ή NOT BETWEEN ...AND

(π.χ. SAL BETWEEN 2500 AND 3000)

**6)** σύνολα τιμών (τελεστής ΙΝ),

(π.χ. JOB IN ('ΠΩΛΗΤΗΣ', 'ΑΝΑΛΥΤΗΣ ΣΥΣΤΗΜΑΤΩΝ')),

**7)** παρενθέσεις.

### Παράδειγμα 1: Χρήση τελεστών σύγκρισης

Η παρακάτω αναζήτηση βρίσκει πωλητές με μισθό πάνω από 1,300 ή προμήθεια μεγαλύτερη του ενός δεκάτου του μισθού τους

SELECT ENAME ΟΝΟΜΑ, SAL ΜΙΣΘΟΣ,COMM ΠΡΟΜΗΘΕΙΑ

FROM EMP

WHERE JOB = 'SALESMAN'

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

**mySQL:**

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

FROM EMP

WHERE JOB = 'SALESMAN'

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

### Παράδειγμα 2: Χρήση BETWEEN … AND

Η παρακάτω αναζήτηση βρίσκει πωλητές που έχουν μισθό μεταξύ 1800 και 2000.

SELECT ENAME, JOB, SAL

FROM EMP

WHERE JOB = 'SALESMAN'

AND SAL BETWEEN 1800 AND 2000;

### Παράδειγμα 3: Χρήση συνόλων (τελεστών IN)

Η παρακάτω αναζήτηση βρίσκει πωλητές ή αναλυτές συστημάτων που έχουν μισθό μεγαλύτερο από 1250

SELECT ENAME, JOB, SAL

FROM EMP

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

AND SAL> 1250;

### Παράδειγμα 4: Χρήση συνάρτησης substr

Η αναζήτηση βρίσκει αναλυτές με όνομα που αρχίζει από C, D, E.

SELECT ename, job FROM emp

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

AND job = 'ANALYST';

**mySQL:**

SELECT ename, job FROM emp

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

AND job = 'ANALYST';

Xρησιμοποιεί τη συνάρτηση substr για να απομονώσει το 1 γράμμα.

SELECT ename, job

FROM emp

WHERE (substr(ename,1,1) IN ('C', 'D', 'E'))

AND job = 'ANALYST';

***Σημείωση***

Η χρήση LIKE για την σύγκριση ορμαθών (ή συμβολοσειρών) (strings) συνδέεται συνήθως με το σύμβολο % όπως φαίνεται στα παραδείγματα:

1. Η παρακάτω εντολή ψάχνει την εργασία του CODD

SELECT ename, job

FROM emp

WHERE ename LIKE 'CODD';

1. ... like 'C%' ; ψάχνει την εργασία αυτών με επώνυμο που αρχίζει από το γράμμα C

SELECT ename, job

FROM emp

WHERE ename LIKE 'C%';

1. ... like '%TE' ; ψάχνει την εργασία αυτών που το όνομα τους τελειώνει σε TE

SELECT ename, job

FROM emp

WHERE ename LIKE'%TE';

1. ... like '%D%' ; ψάχνει την εργασία αυτών με ονοματεπώνυμο που περιέχει το γράμμα D

SELECT ename, job

FROM emp

WHERE ename LIKE '%D%';

1. select distinct job

from emp

where job like 'A%';

ψάχνει τις εργασίες που αρχίζουν από Α .

**mySQL:**

select DISTINCT job

from emp

where job like 'A%';

### Παράδειγμα 5: Χρήση BETWEEN … AND με ημερομηνίες

Η παρακάτω αναζήτηση βρίσκει αυτούς 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 '01/01/1989' and '12/31/1995';

**mySQL:**

SELECT ename, hiredate from emp

where ename between 'C' and 'F'

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

Στο Παράδειγμα μπορείτε να χρησιμοποιήσετε και τη συνάρτηση substr.

***Παράδειγμα*** Βρες 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;

***Παράδειγμα*** Βρες 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;

***Παράδειγμα*** Βρες τους υπάλληλους που είναι πωλητές ή αναλυτές

SELECT ENAME, JOB, SAL, DEPTNO

FROM EMP

WHERE JOB = 'ANALYST'

OR JOB = 'SALESMAN';

***Παράδειγμα*** Βρες τους πωλητές ή αναλυτές που δεν ανήκουν στο τμήμα 10.

SELECT ENAME, JOB, SAL, DEPTNO

FROM EMP

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

AND DEPTNO <> 10;

***Παράδειγμα*** Βρες 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;

***Παράδειγμα*** Βρες πωλητές οποιουδήποτε τμήματος και αναλυτές του Τμήματος 10.

SELECT \*

FROM EMP

WHERE JOB = 'SALESMAN'

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

Ποιό είναι το αποτέλεσμα της επόμενης αναζήτησης;

SELECT \*

FROM EMP

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

AND DEPTNO = 20;

***Παράδειγμα*** Βρες υπάλληλους της Τμήματος 10 που δεν είναι πωλητές ή αναλυτές.

SELECT \*

FROM EMP

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

AND DEPTNO = 10;

***Παράδειγμα*** Βρες ποιοί έχουν μισθό μεταξύ 1,500 και 1,800.

SELECT ENAME,JOB,SAL

FROM EMP

WHERE SAL BETWEEN 1500 AND 1800;

***Παράδειγμα*** Χρησιμοποίησε το τελεστή BETWEEN με τον τελεστή NOT για να βρείς τους υπόλοιπους υπάλληλους.

SELECT ENAME,JOB,SAL

FROM EMP

WHERE SAL NOT BETWEEN 1500 AND 1800;

***Παράδειγμα*** Βρες υπάλληλους που είναι αναλυτές ή πωλητές.

SELECT ENAME,JOB,DEPTNO

FROM EMP

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

***Παράδειγμα*** Βρες τους υπόλοιπους υπάλληλους.

SELECT ENAME,JOB,DEPTNO

FROM EMP

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

***Παράδειγμα*** Βρες 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%');

***Παράδειγμα*** Βρες υπάλληλους που το όνομα τους τελειώνει σε THE και είναι Πωλητές.

SELECT ENAME,JOB,DEPTNO

FROM EMP

WHERE ENAME LIKE '%THE'

AND JOB = 'SALESMAN';

**Σημειώματα**

**Σημείωμα Αναφοράς**

Copyright Πανεπιστήμιο Δυτικής Αττικής, Χ. Σκουρλάς 2018. Χ. Σκουρλάς, «Βάσεις Δεδομένων Ι. Ενότητα 4: Δημιουργία και διαχείριση βάσης προσωπικού με χρήση απλών και σύνθετων δηλώσεων (statements) SQL στα προϊόντα mySQL και Oracle. Έμφαση σε θέματα αναζήτησης (queries)». Έκδοση: 1.0. Αθήνα 2018. 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».

[](file:///C:\Users\pantelis\Downloads\%5b1%5d%20http:\creativecommons.org\licenses\by-nc-sa\4.0\)

[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: