

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή	9
Τι είναι η MySQL;	10
Εκδόσεις της MySQL	12
Βασικοί πόροι της MySQL	13
Τεχνικές απαιτήσεις	14
Λίγα λόγια για το βιβλίο	15
Είναι αυτό το βιβλίο κατάλληλο για εσάς;	15
Συνοδευτική τοποθεσία Ιστού	16
Ερωτήσεις, σχόλια, ή προτάσεις	16
Κεφάλαιο 1: Εγκατάσταση της MySQL	17
Γενικά βήματα εγκατάστασης	18
Εγκατάσταση της MySQL σε Windows	21
Διευθέτηση της MySQL σε Windows	24
Εγκατάσταση της MySQL σε Macintosh	26
Εγκατάσταση της MySQL σε Linux	29
Βασικές επιλογές διευθέτησης	34
Αναβάθμιση της MySQL	37
Κεφάλαιο 2: Εκτέλεση της MySQL	39
Εκτέλεση της MySQL στα Windows και τα Windows NT	40
Εκτέλεση της MySQL σε Mac OS X	49
Εκτέλεση της MySQL σε Linux και Unix	52
Χρήση του mysqladmin	55
Διευθέτηση του κωδικού πρόσβασης χρήστη ρίζας	58
Χρήση του πελάτη mysql	60
Χρήστες και προνόμια	64
Κεφάλαιο 3: Σχεδιασμός βάσεων δεδομένων	73
Κανονικοποίηση	74
Κατανόηση των κλειδιών	75
Κατανόηση των σχέσεων	77
Πρώτη κανονική μορφή	79

	Δεύτερη κανονική μορφή	81
	Τρίτη κανονική μορφή	84
Κεφάλαιο 4:	Δημιουργία βάσης δεδομένων της MySQL	87
	Τύποι δεδομένων της MySQL.....	88
	Πρόσθετα χαρακτηριστικά στηλών.....	93
	Εισαγωγή στα ευρετήρια	96
	Ολοκλήρωση της σχεδίασης ενός πίνακα	98
	Επιλογή μηχανής αποθήκευσης.....	101
	Σύνολα χαρακτήρων και συγκριτικές αλληλουχίες.....	104
	Δημιουργία βάσεων δεδομένων	107
	Δημιουργία πινάκων.....	109
	Τροποποίηση πινάκων.....	114
Κεφάλαιο 5:	Βασική SQL	117
	Χρήση τιμών σε ερωτήματα.....	118
	Εισαγωγή δεδομένων.....	120
	Επιλογή δεδομένων	124
	Χρήση συνθηκών	127
	Χρήση των συνθηκών LIKE και NOT LIKE.....	130
	Εκτέλεση ενώσεων	133
	Ταξινόμηση αποτελεσμάτων ερωτήματος	139
	Περιορισμός αποτελεσμάτων ερωτήματος.....	141
	Ενημέρωση δεδομένων.....	143
	Διαγραφή δεδομένων.....	145
Κεφάλαιο 6:	Συναρτήσεις της MySQL.....	149
	Συναρτήσεις κειμένου	150
	Συνένωση και ψευδώνυμα.....	153
	Αριθμητικές συναρτήσεις.....	158
	Συναρτήσεις ημερομηνίας και ώρας	161
	Μορφοποίηση ημερομηνίας και ώρας.....	165
	Συναρτήσεις κρυπτογράφησης.....	167
	Συναρτήσεις ομαδοποίησης.....	170
	Άλλες συναρτήσεις.....	173
Κεφάλαιο 7:	MySQL και PHP	177
	Σύνδεση στη MySQL και επιλογή βάσης δεδομένων	178
	Εκτέλεση απλών ερωτημάτων.....	182
	Ανάκτηση αποτελεσμάτων ερωτήματος	190
	Χρήση της συνάρτησης mysqli_insert_id()	198
	Χειρισμός σφαλμάτων.....	205
	Ζητήματα ασφάλειας.....	208
Κεφάλαιο 8:	MySQL και Perl.....	221
	Εγκατάσταση της Perl με υποστήριξη για MySQL σε Windows	222
	Εγκατάσταση υποστήριξης Perl για MySQL σε Unix και Mac OS X.....	225

Έλεγχος της Perl και της MySQL.....	227
Σύνδεση στη MySQL.....	231
Ανάκτηση αποτελεσμάτων ερωτήματος.....	234
Εκτέλεση απλών ερωτημάτων.....	238
Ανάκτηση του αναγνωριστικού εισαγωγής.....	243
Χειρισμός σφαλμάτων.....	246
Ζητήματα ασφάλειας.....	250
Χρήση έτοιμων εντολών.....	253
Κεφάλαιο 9: MySQL και Java.....	259
Εγκατάσταση υποστήριξης Java για MySQL.....	260
Σύνδεση στη βάση δεδομένων.....	263
Εκτέλεση απλών ερωτημάτων.....	270
Ανάκτηση αποτελεσμάτων ερωτήματος.....	277
Ανάκτηση του αναγνωριστικού εισαγωγής.....	284
Χρήση έτοιμων εντολών.....	290
Κεφάλαιο 10: Σύνθετη SQL και MySQL.....	295
Εκτέλεση συναλλαγών.....	296
Αναζήτηση πλήρους κειμένου.....	303
Κανονικές παραστάσεις.....	313
Μεταβλητές οριζόμενες από το χρήστη.....	316
Εισαγωγή στις ενώσεις.....	319
Κεφάλαιο 11: Χαρακτηριστικά της MySQL 5.....	325
Αποθηκευμένες ρουτίνες.....	326
Χρήση παραμέτρων OUT.....	346
Ενεργοποιητές.....	349
Προβολές.....	354
Κεφάλαιο 12: Τεχνικές προγραμματισμού.....	361
Αποθήκευση και ανάκτηση δυαδικών δεδομένων.....	362
Δημιουργία σελίδων αποτελεσμάτων ερωτήματος.....	380
Χρήση συναλλαγών με την Perl.....	394
Κεφάλαιο 13: Διαχείριση της MySQL.....	403
Η Διαχείριση της MySQL.....	404
Δημιουργία αντιγράφων ασφαλείας βάσεων δεδομένων.....	408
Εισαγωγή δεδομένων.....	412
Καταγραφή συμβάντων στη MySQL.....	414
Συντήρηση βάσεων δεδομένων.....	417
Βελτίωση της απόδοσης.....	421
Χρήση αρχείων δέσμης.....	425
Παράρτημα Α: Αντιμετώπιση προβλημάτων.....	429
Εγκατάσταση.....	430
Εκκίνηση της MySQL.....	431

Προσπέλαση της MySQL.....	432
Προβλήματα mysql.sock	434
Ερωτήματα που επιστρέφουν παράξενα αποτελέσματα	436
Προβλήματα πρωτοκόλλων πιστοποίησης αυθεντικότητας.....	437
Επαναφορά του κωδικού πρόσβασης ρίζας	438
Παράρτημα Β: Αναφορές για την SQL και τη MySQL	441
Βασική SQL	442
Διαταγές ALTER.....	444
Όροι SQL.....	445
Προνόμια της MySQL.....	446
Τύποι δεδομένων της MySQL.....	447
Συναρτήσεις της MySQL	449
Άλλες αναφορές	453
Παράρτημα Γ: Πόροι	455
Πόροι συγκεκριμένοι για τη MySQL.....	456
Εφαρμογές της MySQL από άλλους κατασκευαστές	458
SQL.....	459
PHP.....	460
Perl.....	461
Java	462
Άλλα	463
Ευρετήριο	465

ΣΥΝΑΡΤΗΣΕΙΣ ΤΗΣ MYSQL

6

Η MySQL διαθέτει δεκάδες ενσωματωμένες συναρτήσεις που έχουν σχεδιαστεί να απλοποιούν συνηθισμένες εργασίες. Σε αυτό το κεφάλαιο καλύπτονται οι πιο χρήσιμες, αλλά μην εκπλαγείτε αν διαπιστώσετε ότι στις πιο πρόσφατες εκδόσεις του λογισμικού της MySQL έχουν προστεθεί και άλλες. Κατά την άποψή μου, οι συναρτήσεις είναι αυτές που κάνουν την εκτέλεση ερωτημάτων σε μια βάση δεδομένων τόσο χρήσιμη (η SQL από μόνη της είναι πολύ απλή).

Οι περισσότερες από τις συναρτήσεις που θα μάθετε σε αυτό το κεφάλαιο χρησιμοποιούνται σε συνδυασμό με ερωτήματα SQL για τη μορφοποίηση και την τροποποίηση των επιστρεφόμενων δεδομένων. Όπως και στο προηγούμενο κεφάλαιο, κάθε παράδειγμα παρουσιάζεται μέσα από τον πελάτη `mysql`. Κάθε παράδειγμα βασίζεται στην υπάρχουσα βάση δεδομένων *accounting* και θα χρησιμοποιηθεί για την επεξήγηση καθιερωμένων τεχνικών βάσεων δεδομένων.

Συναρτήσεις κειμένου

Η πρώτη ομάδα συναρτήσεων που θα παρουσιάσω αφορά τις συναρτήσεις που προορίζονται για το χειρισμό διαφόρων στηλών κειμένου και χαρακτήρων. Οι περισσότερες από τις συναρτήσεις αυτής της κατηγορίας παρουσιάζονται στον Πίνακα 6.1.

Για να χρησιμοποιήσετε οποιαδήποτε συνάρτηση πρέπει να τροποποιήσετε το ερώτημά σας ώστε να καθορίσετε σε ποιες στήλες θα εφαρμοστεί η συνάρτηση.

```
SELECT FUNCTION(στήλη) FROM όνομα_πίνακα
```

Για να καθορίσετε πολλές στήλες, μπορείτε να γράψετε ένα ερώτημα όπως τα παρακάτω:

```
◆ SELECT *, ΣΥΝΑΡΤΗΣΗ(στήλη) FROM
  → όνομα_πίνακα
```

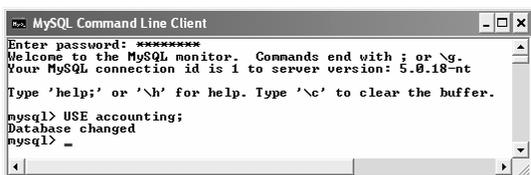
```
◆ SELECT στήλη1, ΣΥΝΑΡΤΗΣΗ(στήλη2),
  → στήλη3 FROM όνομα_πίνακα
```

Αν και στα ονόματα συναρτήσεων δεν γίνεται διάκριση πεζών-κεφαλαίων, θα συνεχίσω να τα γράφω με κεφαλαία ώστε να ξεχωρίζουν από τα ονόματα πινάκων και στηλών (τακτική που ακολουθώ και με όρους SQL). Ένας σημαντικός κανόνας με τις συναρτήσεις είναι ότι δεν επιτρέπεται να αφήνετε κενά διαστήματα μεταξύ του ονόματος της συνάρτησης και της παρένθεσης ανοίγματος της συνάρτησης, παρόλο που επιτρέπονται τα κενά μέσα στις παρενθέσεις.

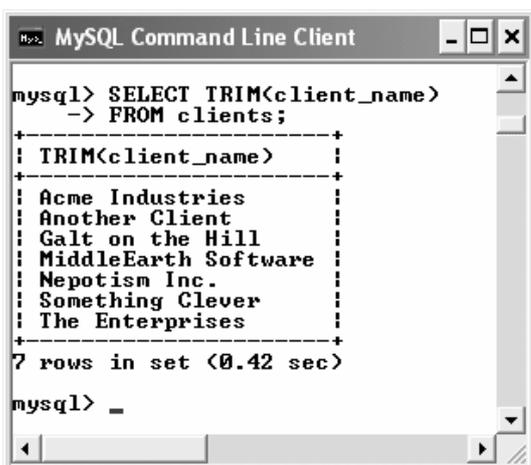
Θα επαναλάβω αυτή την τελευταία επισήμανση επειδή αποτελεί συνηθισμένη αιτία προβλημάτων: *μην αφήνετε κενό διάστημα μεταξύ του ονόματος μιας συνάρτησης και της παρένθεσης ανοίγματος!*

Πίνακας 6.1 Αυτές είναι μερικές, αλλά όχι όλες, από τις συναρτήσεις που μπορείτε να χρησιμοποιήσετε σε στήλες κειμένου στη MySQL.

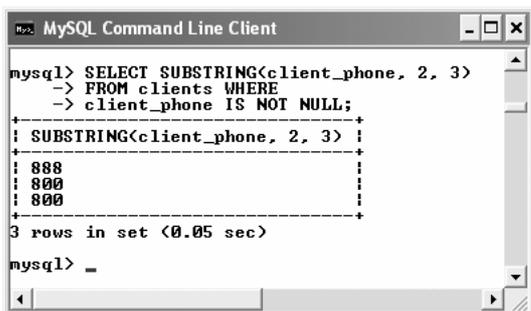
Συναρτήσεις κειμένου	
ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΗΣΗ	ΣΚΟΠΟΣ
LENGTH(<i>text</i>)	Επιστρέφει το μήκος του αλφαριθμητικού που είναι αποθηκευμένο στη στήλη.
LEFT(<i>text</i> , <i>x</i>)	Επιστρέφει τους αριστερότερους <i>x</i> χαρακτήρες από μια στήλη.
RIGHT(<i>text</i> , <i>x</i>)	Επιστρέφει τους δεξιότερους <i>x</i> χαρακτήρες από μια στήλη.
TRIM(<i>text</i>)	Περικόπτει τα περιττά διαστήματα από την αρχή και το τέλος του αποθηκευμένου αλφαριθμητικού.
UPPER(<i>text</i>)	Μετατρέπει σε κεφαλαία ολόκληρο το αποθηκευμένο αλφαριθμητικό.
LOWER(<i>text</i>)	Μετατρέπει το αποθηκευμένο αλφαριθμητικό σε πεζά.
SUBSTRING(<i>text</i> , <i>start</i> , <i>length</i>)	Επιστρέφει χαρακτήρες καθορισμένου μήκους (<i>length</i>) από κείμενο (<i>text</i>) ξεκινώντας από συγκεκριμένο αρχικό σημείο (<i>start</i> — με αφετηρία το 1).
FIND_IN_SET(<i>str</i> , <i>set</i>)	Επιστρέφει ένα θετικό αριθμό αν το αλφαριθμητικό (<i>str</i>) βρίσκεται στο σύνολο (<i>set</i>), διαφορετικά επιστρέφει 0.



Εικόνα 6.1 Και πάλι, ο πελάτης mysql και η βάση δεδομένων *accounting* θα αποτελούν τη βάση όλων των παραδειγμάτων σε αυτό το κεφάλαιο.



Εικόνα 6.2 Η συνάρτηση TRIM() περικόπτει τα περιττά κενά από την αρχή και το τέλος των επιστρεφόμενων τιμών.



Εικόνα 6.3 Αυτή η εφαρμογή της συνάρτησης SUBSTRING() επιστρέφει μόνο το δεύτερο από τους πέντε χαρακτήρες μιας αποθηκευμένης τιμής.

Για να μορφοποιήσετε κείμενο:

1. Ανοίξτε τον πελάτη mysql και επιλέξτε τη βάση δεδομένων *accounting* (**Εικόνα 6.1**).

```
USE accounting;
```

Όπως και στο προηγούμενο κεφάλαιο, από εδώ και στο εξής θα υποθέσω ότι χρησιμοποιείτε ήδη τη βάση δεδομένων *accounting* μέσα στον πελάτη mysql.

2. Αφαιρέστε όλα τα επιπλέον κενά ("λευκό χώρο") από τα ονόματα πελατών (**Εικόνα 6.2**).

```
SELECT TRIM(client_name) FROM
-> clients;
```

Η συνάρτηση TRIM() διαγράφει αυτόματα τον κενό χώρο (διαστήματα, στηλοθέτες, και επαναφορές κεφαλής) από την αρχή και από το τέλος ενός αλφαριθμητικού.

3. Εμφανίστε μόνο τους κωδικούς περιοχής για όλους τους τηλεφωνικούς αριθμούς πελατών (**Εικόνα 6.3**).

```
SELECT SUBSTRING(client_phone, 2, 3)
-> FROM clients WHERE client_phone IS
-> NOT NULL;
```

Η συνάρτηση SUBSTRING() επιστρέφει ένα τμήμα ενός αλφαριθμητικού. Το πρώτο όρισμά της είναι το κυριολεκτικό αλφαριθμητικό ή το όνομα στήλης. Το δεύτερο όρισμα καθορίζει τη θέση από την οποία θα εξαχθεί το αλφαριθμητικό, με αφετηρία το 1. Επειδή ο πρώτος χαρακτήρας σε έναν τηλεφωνικό αριθμό πρέπει να είναι η παρένθεση ανοίγματος, ξεκινώ από το δεύτερο χαρακτήρα. Το τρίτο όρισμα, που είναι προαιρετικό, υπαγορεύει πόσοι χαρακτήρες πρέπει να επιστραφούν (αν παραλειφθεί αυτό το όρισμα επιστρέφονται όλοι οι υπόλοιποι χαρακτήρες).

συνεχίζεται

4. Εντοπίστε το πιο μακροσκελές όνομα κατηγορίας δαπάνης (**Εικόνα 6.4**).

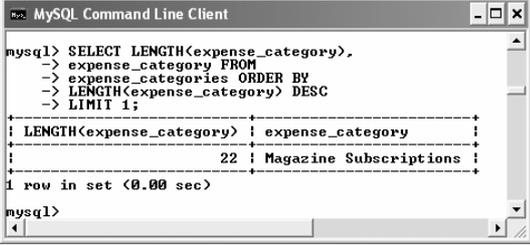
```
SELECT LENGTH(expense_category),
→ expense_category FROM
expense_categories ORDER BY
LENGTH(expense_category) DESC
LIMIT 1;
```

Αυτό το ερώτημα καταρχήν συγκεντρώνει όλες τις κατηγορίες δαπανών, μαζί με το μήκος τους. Κατόπιν ταξινομεί αυτά τα δεδομένα σύμφωνα με την πιο μακροσκελή κατηγορία. Τέλος, επιστρέφεται μόνο η πρώτη (δηλαδή η μεγαλύτερη) εγγραφή.

✓ Συμβουλές

- Ένα ερώτημα όπως αυτό στο Βήμα 4 (επίσης και στην Εικόνα 6.4) μπορεί να είναι χρήσιμο για τον ακριβή καθορισμό του μήκους των στηλών σας όταν προστεθούν μερικές εγγραφές στη βάση δεδομένων σας.
- Μπορείτε να χρησιμοποιείτε τις περισσότερες συναρτήσεις της MySQL σε διάφορα ερωτήματα εκτός από ερωτήματα επιλογής (SELECT). Πολύ συχνά χρησιμοποιείτε μια συνάρτηση για να μορφοποιείτε τα δεδομένα σε ένα ερώτημα INSERT.
- Υπάρχουν δύο άλλες χρήσιμες συναρτήσεις που δεν αναφέρονται εδώ λόγω της πολυπλοκότητας στη σύνταξη και τη χρήση τους. Η πρώτη είναι η LOCATE(), η οποία επιστρέφει την αρχική θέση ενός χαρακτήρα ή αλφαριθμητικού που βρίσκεται μέσα σε ένα άλλο. Η δεύτερη είναι η REPLACE(), που επιστρέφει ένα αλφαριθμητικό αφού αντικαταστήσει κάποιους χαρακτήρες ή κείμενο με άλλους χαρακτήρες και κείμενο. Και οι δύο, φυσικά, περιγράφονται λεπτομερώς στο εγχειρίδιο.
- Οι συναρτήσεις μπορούν να εφαρμόζονται με τον ίδιο τρόπο και σε στήλες και σε κυριολεκτικά αλφαριθμητικά. Για παράδειγμα, το επόμενο είναι απολύτως αποδεκτό:


```
SELECT UPPER('μετατροπήσεκεφαλαία')
```
- Νέα στη MySQL 4.1 είναι η δυνατότητα μετατροπής κειμένου από ένα σύνολο χαρακτήρων σε κάποιο άλλο. Για να το κάνετε αυτό, χρησιμοποιήστε τη συνάρτηση CONVERT(). Έχετε υπόψη ότι δεν μεταφράζεται κείμενο από μια γλώσσα σε μια άλλη· απλώς αλλάζουν οι χαρακτήρες που χρησιμοποιούνται.



```
mysql> SELECT LENGTH(expense_category),
→ expense_category FROM
→ expense_categories ORDER BY
→ LENGTH(expense_category) DESC
→ LIMIT 1;
+-----+-----+
| LENGTH(expense_category) | expense_category |
+-----+-----+
| 22 | Magazine Subscriptions |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Εικόνα 6.4 Μπορείτε ακόμη και να χρησιμοποιήσετε μια συνάρτηση μέσα σε έναν όρο ORDER BY.

Συνένωση και ψευδώνυμο

Η CONCAT(), ίσως η πιο χρήσιμη συνάρτηση κειμένου, αξίζει ξεχωριστή αναφορά, μαζί με το συχνό συνοδευτικό της στην SQL, το ψευδώνυμο (alias). Η συνάρτηση CONCAT() πραγματοποιεί *συνένωση* (concatenation) πολλών τιμών μαζί. Η σύνταξη για συνένωση απαιτεί την τοποθέτηση, μέσα σε παρενθέσεις, διάφορων τιμών που θέλετε να συνθροιστούν, στη σειρά, και διαχωρισμένες με κόμματα:

```
CONCAT(στήλη1, στήλη2)
```

Ενώ μπορείτε — και συνήθως θα το κάνετε — να εφαρμόζετε την CONCAT() σε στήλες, έχετε επίσης τη δυνατότητα να ενσωματώνετε αλφαριθμητικά τοποθετώντας τα μέσα σε απλά εισαγωγικά. Για να μορφοποιήσετε το όνομα ενός ατόμου ως *Επώνυμο, Όνομα* από δύο στήλες, μπορείτε να χρησιμοποιήσετε τη σύνταξη

```
CONCAT(επώνυμο, ' ', ' ', όνομα)
```

Επειδή η συνένωση χρησιμοποιείται για τη δημιουργία μιας νέας τιμής, χρειάζεστε ένα νέο τρόπο για να αναφέρεστε στο επιστρεφόμενο αποτέλεσμα. Εδώ λοιπόν βρίσκει εφαρμογή η έννοια των *ψευδωνύμων* της SQL. Ένα ψευδώνυμο (alias) είναι απλώς μια συμβολική μετονομασία. Εφαρμόζεται με χρήση του όρου AS:

```
SELECT CONCAT(επώνυμο, ' ', ' ',  
→ όνομα) AS όνομα FROM χρήστες
```

Το αποτέλεσμα αυτού του ερωτήματος θα είναι η μορφοποίηση των ονομάτων των χρηστών με τον τρόπο που θέλετε, και η επιστρεφόμενη στήλη θα ονομάζεται *όνομα*.

Επίσης, στα ερωτήματά σας μπορείτε να δημιουργήσετε ένα ψευδώνυμο από οποιαδήποτε στήλη ή πίνακα. Η γενική σύνταξη είναι η εξής:

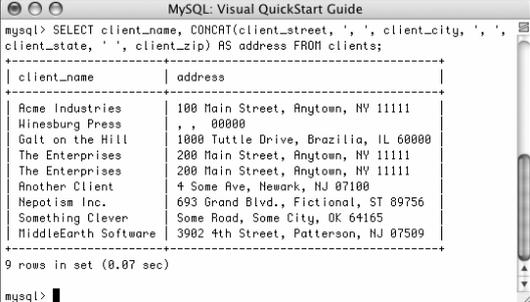
```
SELECT στήλη AS ψευδώνυμο FROM πίνακας  
→ AS ψεύδ_πίνακας
```

Για να χρησιμοποιήσετε συνένωση και ψευδώνυμα:

1. Εμφανίστε όλες τις πληροφορίες διευθύνσεων πελατών ως μία τιμή (Εικόνα 6.5).

```
SELECT client_name,
→ CONCAT(client_street, ', ',
→ client_city, ', ', client_state,
→ ' ', client_zip) AS address
→ FROM clients;
```

Με αυτήν την πρώτη εφαρμογή της συνάρτησης CONCAT() συγκεντρώνονται όλες οι πληροφορίες διευθύνσεων σε μια τακτοποιημένη στήλη, που μετονομάζεται σε *address* (δείτε την εικόνα). Αν θέλατε, θα μπορούσατε να προσθέσετε στο ερώτημα τους όρους WHERE *client_street IS NOT NULL and client_city IS NOT NULL and client_state IS NOT NULL* για να αποκλείσετε τις ημιτελείς διευθύνσεις. (Επίσης, θα μπορούσατε απλώς να προσθέσετε τον έναν από τους τρεις αυτούς όρους.)



```
mysql> SELECT client_name, CONCAT(client_street, ', ', client_city, ', ',
client_state, ' ', client_zip) AS address FROM clients;
```

client_name	address
Acme Industries	100 Main Street, Anytown, NY 11111
Hinesburg Press	, , 00000
Galt on the Hill	1000 Tuttle Drive, Brazilia, IL 60000
The Enterprises	200 Main Street, Anytown, NY 11111
The Enterprises	200 Main Street, Anytown, NY 11111
Another Client	4 Some Ave, Newark, NJ 07100
Nepotism Inc.	693 Grand Blvd., Fictional, ST 89756
Something Clever	Some Road, Some City, OK 64165
MiddleEarth Software	3902 4th Street, Patterson, NJ 07509

9 rows in set (0.07 sec)

```
mysql>
```

Εικόνα 6.5 Η συνάρτηση CONCAT() είναι ένα από τα πιο χρήσιμα εργαλεία για τη βελτιστοποίηση των αποτελεσμάτων των ερωτημάτων σας. Να θυμάστε ότι δεν επιτρέπεται να αφήνετε κενά διαστήματα μεταξύ του ονόματος μιας συνάρτησης και της παρένθεσης ανοίγματός της.

2. Επιλέξτε κάθε δαπάνη, μαζί με την περιγραφή και την κατηγορία της (**Εικόνα 6.6**).

```
SELECT expense_amount, expense_date,
CONCAT(expense_category, ': ',
expense_description) FROM expenses,
expense_categories WHERE
expenses.expense_category_id =
expense_categories.expense_
category_id;
```

Σε αυτό το ερώτημα έχω εκτελέσει μια ένωση ώστε να εμφανίσω ταυτόχρονα πληροφορίες δαπανών και κατηγοριών δαπανών. Η συνένωση πραγματοποιείται σε δύο στήλες από δύο διαφορετικούς πίνακες.

Αν εξετάσετε τα ονόματα στηλών στην εικόνα, θα δείτε το αποτέλεσμα της χρήσης συναρτήσεων χωρίς ψευδώνυμα.

συνεχίζεται

expense_amount	expense_date	CONCAT(expense_category, ': ', expense_description)
19.99	2002-04-20 00:00:00	Books: Larry Ullman's "MySQL: Visual QuickStart Guide"
105.50	2006-02-03 14:51:17	Travel: Palmer House Hotel, Chicago
689.00	2006-02-03 14:51:17	Travel: Flight to Chicago
99.99	2006-02-02 23:01:31	Computer Software: Mmm...software
64.99	2006-02-03 14:51:17	Travel: Flight from Chicago
64.50	0000-00-00 00:00:00	Computer Hardware: Apple PowerBook
6464.00	2006-01-26 00:00:00	Plaster of Paris: Sculpting
67.94	2006-04-20 00:00:00	Pencils: Writing implements needed to fill in little bubbles on tests.
1.97	2006-04-20 00:00:00	Drywall: Reconstruction
3216.00	2006-01-26 00:00:00	Sand Paper: Sanding
9712.97	2006-01-26 00:00:00	Sand Paper: There's a lot of sanding to do.
312.64	2006-02-03 14:14:11	Computer Software: Software upgrade.
25.00	2006-05-09 00:00:00	Picture Frames: Frame to display Jess' diploma.
39.99	2006-05-09 00:00:00	Internet Access Fees: Monthly expense for cable modem Internet access.
19.99	2006-05-09 00:00:00	Books: Larry Ullman's "PHP for the World Wide Web: Visual QuickStart Guide".
29.99	2006-05-24 00:00:00	Books: Larry Ullman's "PHP Advanced for the World Wide Web: Visual QuickPro Guide"
129.00	2006-05-24 00:00:00	Computer Software: Software to create visual representations of databases.
29.99	2006-05-24 00:00:00	Books: Larry Ullman's "PHP Advanced for the World Wide Web: Visual QuickPro Guide"

18 rows in set (0.00 sec)

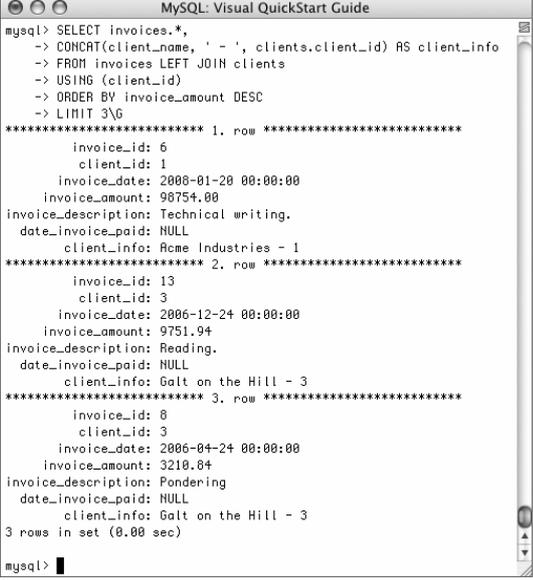
Εικόνα 6.6 Οι συναρτήσεις μπορούν να εφαρμόζονται με διαφορετικούς τρόπους, καθώς και σε πολλούς πίνακες.

3. Εμφανίστε τα τρία ακριβότερα τιμολόγια, μαζί με το όνομα πελάτη και τον αναγνωριστικό αριθμό (Εικόνα 6.7).

```
SELECT invoices.*,
CONCAT(client_name, ' - ',
→ clients.client_id) AS client_info
FROM invoices LEFT JOIN clients
USING (client_id)
ORDER BY invoice_amount DESC
LIMIT 3\G
```

Για να εκτελέσω αυτό το ερώτημα, χρησιμοποιώ μια αριστερή ένωση, ταξινομώ τα αποτελέσματα σύμφωνα με το πεδίο `invoice_amount`, και τα περιορίζω σε ακριβώς τρεις εγγραφές. Η συνάρτηση `CONCAT()` εφαρμόζεται στο όνομα πελάτη και τον αναγνωριστικό αριθμό (ID).

Για να είναι ευκολότερη η εξέταση των αποτελεσμάτων, χρησιμοποιώ τον τροποποιητή `\G` για να κλείσω το ερώτημα (αυτό είναι μια λειτουργία του `mysql` που περιέγραφα στο Κεφάλαιο 5, "Βασική SQL").



```
mysql> SELECT invoices.*,
→ CONCAT(client_name, ' - ', clients.client_id) AS client_info
→ FROM invoices LEFT JOIN clients
→ USING (client_id)
→ ORDER BY invoice_amount DESC
→ LIMIT 3\G
***** 1. row *****
invoice_id: 6
client_id: 1
invoice_date: 2008-01-20 00:00:00
invoice_amount: 98754.00
invoice_description: Technical writing.
date_invoice_paid: NULL
client_info: Acme Industries - 1
***** 2. row *****
invoice_id: 13
client_id: 3
invoice_date: 2006-12-24 00:00:00
invoice_amount: 9751.94
invoice_description: Reading.
date_invoice_paid: NULL
client_info: Galt on the Hill - 3
***** 3. row *****
invoice_id: 8
client_id: 3
invoice_date: 2006-04-24 00:00:00
invoice_amount: 3218.84
invoice_description: Pondering
date_invoice_paid: NULL
client_info: Galt on the Hill - 3
3 rows in set (0.00 sec)

mysql>
```

Εικόνα 6.7 Οι τεχνικές με τη συνάρτηση `CONCAT()` και τα ψευδώνυμα μπορούν να εφαρμόζονται σε οποιοδήποτε ερώτημα, συμπεριλαμβανομένων και των ενώσεων.

4. Απλοποιήστε το ερώτημα του Βήματος 2 χρησιμοποιώντας ψευδώνυμα για τα ονόματα πινάκων (Εικόνα 6.8).

```
SELECT expense_amount, expense_date,
→ CONCAT(expense_category, ': ',
→ expense_description) FROM
expenses AS e,
expense_categories AS e_c
WHERE e.expense_category_id =
→ e_c.expense_category_id;
```

Το ερώτημα είναι ίδιο με αυτό του Βήματος 2, μόνο που έχω απλοποιήσει τη σύνταξη του χρησιμοποιώντας ψευδώνυμα για τα ονόματα πινάκων. Σε μεγάλα ερωτήματα και ενώσεις η χρήση τέτοιων συντομεύσεων είναι αρκετά βολική.

✓ **Συμβουλές**

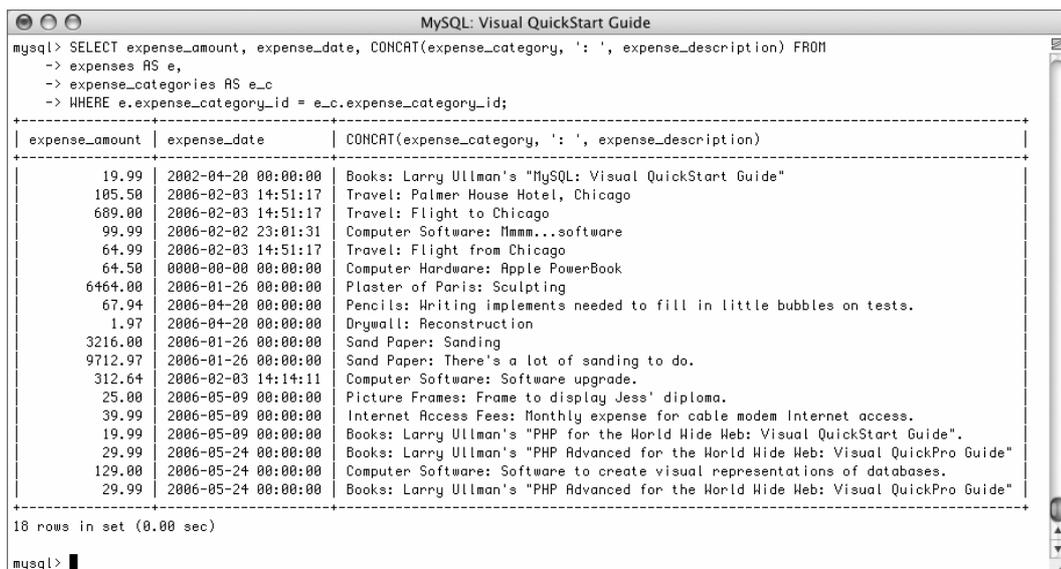
- Η CONCAT() έχει μια επακόλουθη συνάρτηση με όνομα CONCAT_WS(), όπου το _WS είναι τα αρχικά των λέξεων with separator (με διαχωριστικό). Η σύνταξη είναι:

```
CONCAT_WS(διαχωριστικό, στήλη1, 'στήλη2, ...).
```

Το διαχωριστικό παρεμβάλλεται μεταξύ των στηλών που καταχωρίζονται. Μια ωραία δυνατότητα αυτής της συνάρτησης είναι ότι παραλείπει οποιεσδήποτε τιμές NULL.

- Τα ψευδώνυμα μπορούν να έχουν μήκος μέχρι 255 χαρακτήρες και γράφονται πάντα με διάκριση πεζών-κεφαλαίων.
- Ο όρος AS, που χρησιμοποιείται για τη δημιουργία ψευδωνύμων, είναι προαιρετικός. Έχετε τη δυνατότητα να γράψετε ένα ερώτημα με πιο απλό τρόπο, ως εξής:

```
SELECT ψευδώνυμο στήλης FROM πίνακας
```



Εικόνα 6.8 Έχω απλοποιήσει τα ερωτήματά μου, χωρίς να επηρεάζεται το τελικό αποτέλεσμα, με τη χρήση ψευδωνύμων για τα ονόματα των πινάκων μου (συγκρίνετε με την Εικόνα 6.6).

Αριθμητικές συναρτήσεις

Εκτός από τους βασικούς μαθηματικούς τελεστές που χρησιμοποιεί η MySQL (για την πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό, και τη διαίρεση), υπάρχουν περίπου είκοσι συναρτήσεις αποκλειστικές στη μορφοποίηση και την εκτέλεση υπολογισμών σε αριθμητικές στήλες. Στον Πίνακα 6.2 παρατίθενται οι πιο συνηθισμένες από αυτές, μερικές από τις οποίες θα παρουσιάσω σύντομα.

Θα ήθελα να δώσω έμφαση σε τρεις από αυτές τις συναρτήσεις: Τις `FORMAT()`, `ROUND()`, και `RAND()`. Η πρώτη — η οποία από τεχνικής πλευράς δεν σχετίζεται αποκλειστικά με αριθμούς — μετατρέπει οποιονδήποτε αριθμό σε μια πιο συμβατική μορφή. Για παράδειγμα, αν αποθηκεύσατε το κόστος ενός αυτοκινήτου ως 20198.2, η συνάρτηση `FORMAT(car_cost, 2)` θα μετέτρεπε αυτόν τον αριθμό στην πιο συνηθισμένη μορφή 20,198.20.

Η `ROUND()` δέχεται μία τιμή, συνήθως από μια στήλη, και τη στρογγυλοποιεί μέχρι έναν καθορισμένο αριθμό δεκαδικών θέσεων. Αν δεν έχει οριστεί καμία δεκαδική θέση, ο αριθμός στρογγυλοποιείται στον πλησιέστερο ακέραιο. Αν οριστούν περισσότερες δεκαδικές θέσεις από όσες υπάρχουν στον αρχικό αριθμό, οι υπόλοιπες θέσεις συμπληρώνονται με μηδενικά (στα δεξιά της υποδιαστολής).

Η συνάρτηση `RAND()`, όπως θα φαντάζεστε, χρησιμοποιείται για την επιστροφή τυχαίων αριθμών. Πιο συγκεκριμένα, επιστρέφει μια τιμή μεταξύ 0 και 1.0.

```
SELECT RAND()
```

Ένα επιπλέον όφελος της συνάρτησης `RAND()` είναι ότι μπορεί να χρησιμοποιηθεί με τα ερωτήματά σας για να επιστρέφει το αποτέλεσμα σε μια τυχαία σειρά.

```
SELECT * FROM όνομα_πίνακα ORDER BY RAND()
```

Πίνακας 6.2 Εδώ παρουσιάζονται είναι οι πιο χρησιμοποιούμενες αριθμητικές συναρτήσεις, ενώ παραλείπονται οι διάφορες τριγωνομετρικές και εκθετικές συναρτήσεις.

Αριθμητικές συναρτήσεις	
ΣΥΝΑΡΤΗΣΗ ΚΑΙ ΧΡΗΣΗ	ΣΚΟΠΟΣ
<code>ABS(num)</code>	Επιστρέφει την απόλυτη τιμή του αριθμού (<i>num</i>).
<code>CEILING(num)</code>	Επιστρέφει τον επόμενο μεγαλύτερο ακέραιο που βασίζεται στην τιμή του <i>num</i> .
<code>FLOOR(num)</code>	Επιστρέφει την ακέραια τιμή του <i>num</i> .
<code>FORMAT(num, y)</code>	Επιστρέφει τον αριθμό (<i>num</i>) μορφοποιημένο ως ένα αριθμό με <i>y</i> δεκαδικές θέσεις και κόμματα κάθε τρία ψηφία.
<code>MOD(x, y)</code>	Επιστρέφει το υπόλοιπο της διαίρεσης του <i>x</i> με το <i>y</i> (το ένα από αυτά ή και τα δύο μπορεί να είναι μια στήλη).
<code>POW(x, y)</code>	Επιστρέφει την τιμή του <i>x</i> υψωμένη στη δύναμη του <i>y</i> .
<code>RAND()</code>	Επιστρέφει έναν τυχαίο αριθμό μεταξύ 0 και 1.0.
<code>ROUND(x, y)</code>	Επιστρέφει τον αριθμό <i>x</i> στρογγυλοποιημένο σε <i>y</i> δεκαδικές θέσεις.
<code>SIGN(num)</code>	Επιστρέφει μια τιμή που δείχνει αν είναι ο <i>num</i> είναι αρνητικός (-1), μηδέν (0), ή θετικός (+1).
<code>SQRT(num)</code>	Υπολογίζει την τετραγωνική ρίζα του <i>num</i> .

Για να χρησιμοποιήσετε αριθμητικές συναρτήσεις:

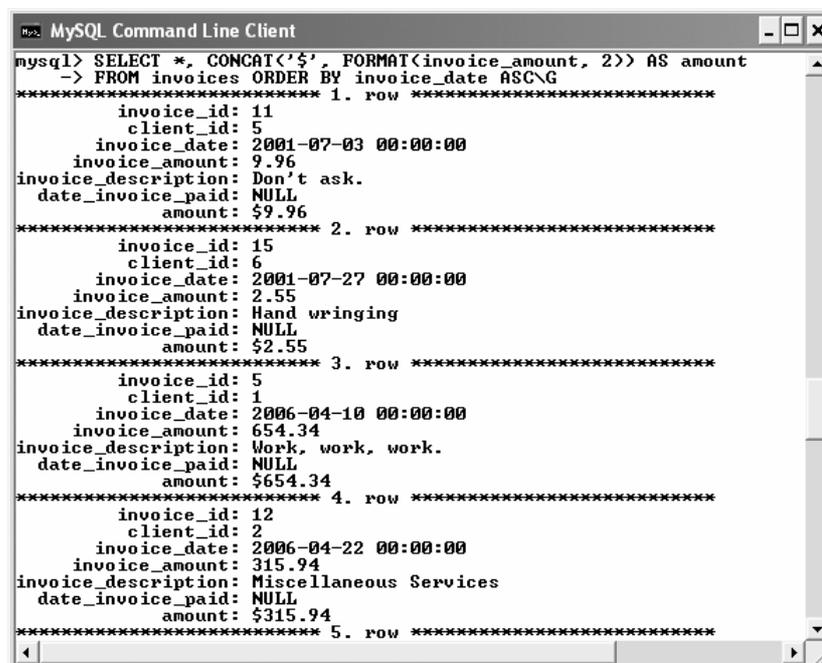
1. Εμφανίστε τα τιμολόγια κατά ημερομηνία, μορφοποιώντας τα ποσά ως δολάρια (Εικόνα 6.9).

```
SELECT *, CONCAT('$', FORMAT
→ (invoice_amount, 2)) AS amount FROM
→ invoices ORDER BY invoice_date ASC\G
```

Χρησιμοποιώντας τη συνάρτηση FORMAT() με τον τρόπο που μόλις περιγράψαμε, σε συνδυασμό με την CONCAT(), μπορείτε να μετατρέψετε οποιονδήποτε αριθμό σε μια νομισματική μορφή που μπορείτε να χρησιμοποιήσετε σε μια ιστοσελίδα ή μια εφαρμογή.

Παρατηρήστε στην εικόνα πώς το ποσό του τιμολογίου στην πραγματικότητα επιστρέφεται δύο φορές: μία φορά ως τμήμα όλων των στηλών (*) και άλλη μία φορά με την καθορισμένη μορφοποίηση.

συνεχίζεται



```
mysql> SELECT *, CONCAT('$', FORMAT
→ (invoice_amount, 2)) AS amount
→ FROM invoices ORDER BY invoice_date ASC\G
***** 1. row *****
invoice_id: 11
client_id: 5
invoice_date: 2001-07-03 00:00:00
invoice_amount: 9.96
invoice_description: Don't ask.
date_invoice_paid: NULL
amount: $9.96
***** 2. row *****
invoice_id: 15
client_id: 6
invoice_date: 2001-07-27 00:00:00
invoice_amount: 2.55
invoice_description: Hand wringing
date_invoice_paid: NULL
amount: $2.55
***** 3. row *****
invoice_id: 5
client_id: 1
invoice_date: 2006-04-10 00:00:00
invoice_amount: 654.34
invoice_description: Work, work, work.
date_invoice_paid: NULL
amount: $654.34
***** 4. row *****
invoice_id: 12
client_id: 2
invoice_date: 2006-04-22 00:00:00
invoice_amount: 315.94
invoice_description: Miscellaneous Services
date_invoice_paid: NULL
amount: $315.94
***** 5. row *****
```

Εικόνα 6.9 Η εφαρμογή δύο συναρτήσεων και της σωστής μορφοποίησης στη στήλη invoice_amount δίνει τις καλύτερες τιμές amount.

2. Στρογγυλοποιήστε κάθε ποσό δαπάνης στο πλησιέστερο ποσό σε δολάρια (Εικόνα 6.10).

```
SELECT ROUND(expense_amount),
→ expense_amount FROM expenses;
```

Η συνάρτηση ROUND(), όταν δεν καθορίσετε ένα δεκαδικό όρισμα, απλώς στρογγυλοποιεί κάθε τιμή στον πλησιέστερο ακέραιο.

3. Ανακτήστε όλα τα ονόματα πελατών σε τυχαία σειρά, δύο φορές (Εικόνα 6.11).

```
SELECT client_id, client_name FROM
→ clients ORDER BY RAND();
```

```
SELECT client_id, client_name FROM
→ clients ORDER BY RAND();
```

Αν και αυτή μπορεί να μην είναι η πρακτικότερη χρήση του όρου ORDER BY RAND(), σας δίνει μια ιδέα για τον τρόπο λειτουργίας του. Ενώ η συνάρτηση RAND() δεν είναι απολύτως τυχαία, είναι αρκετά αποτελεσματική στις περισσότερες περιπτώσεις. Σημειώστε ότι δεν χρειάζεται να καθορίσετε σε ποια στήλη εφαρμόζεται η RAND().

✓ Συμβουλές

- Μαζί με τις μαθηματικές συναρτήσεις που παρουσιάζονται εδώ, υπάρχουν διαθέσιμες και τριγωνομετρικές, εκθετικές, και άλλου είδους συναρτήσεις. Φυσικά, μπορείτε επίσης να χρησιμοποιήσετε οποιονδήποτε από τους μαθηματικούς τελεστές: +, -, *, και /.
- Η συνάρτηση MOD() ισοδυναμεί με το σύμβολο του ποσοστού:


```
SELECT MOD(9,2)
SELECT 9%2
```
- Και πάλι, να θυμάστε ότι οι συναρτήσεις μπορούν να εφαρμόζονται σε στήλες ή σε προκαθορισμένες τιμές. Τα επόμενα ερωτήματα είναι απολύτως αποδεκτά:


```
SELECT ROUND(34.089, 1)
SELECT SQRT(81)
SELECT ABS(-8)
```

```
mysql> SELECT ROUND(expense_amount), expense_amount
→ FROM expenses;
```

ROUND(expense_amount)	expense_amount
2	1.97
20	19.99
20	19.99
25	25.00
30	29.99
40	39.99
65	64.50
65	64.99
68	67.94
100	99.99
105	104.50
129	129.00
313	312.64
689	689.00
3216	3216.00
6464	6464.00
9713	9712.97

17 rows in set (0.19 sec)

```
mysql> _
```

Εικόνα 6.10 Η συνάρτηση ROUND() είναι χρήσιμη σε καταστάσεις όπου δεν έχουν σημασία οι δεκαδικές τιμές.

```
mysql> SELECT client_id, client_name
→ FROM clients ORDER BY RAND();
```

client_id	client_name
5	Nepotism Inc.
3	The Enterprises
4	Another Client
6	Something Clever
1	Acme Industries
7	MiddleEarth Software
2	Galt on the Hill

7 rows in set (0.07 sec)

```
mysql> SELECT client_id, client_name
→ FROM clients ORDER BY RAND();
```

client_id	client_name
2	Galt on the Hill
4	Another Client
5	Nepotism Inc.
7	MiddleEarth Software
3	The Enterprises
1	Acme Industries
6	Something Clever

7 rows in set (0.00 sec)

```
mysql> _
```

Εικόνα 6.11 Η εκτέλεση του ίδιου ερωτήματος δύο φορές με τον όρο ORDER BY RAND() επιστρέφει τα ίδια αποτελέσματα αλλά σε διαφορετική σειρά.