



ΠΕΡΙΕΧΟΜΕΝΑ:

1. Ένωση Ερωτημάτων (UNION)
2. Εισαγωγή στις Συναρτήσεις
 1. IF και IFNULL
 2. CASE και COALESCE
3. Περίπλοκα Ερωτήματα

Αλέξανδρος Τ.

Σμαραγδένιος Χορηγός Μαθήματος

Νίκος Π.

Ασημένιος Χορηγός Μαθήματος

ΜΑΘΗΜΑ 1.8: ΕΝΩΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

Σύνταξη της UNION:

```
QUERY-1
UNION [ALL]
QUERY-2
UNION [ALL]
...
```

- Η **UNION** επιστρέφει την ένωση των αποτελεσμάτων των ερωτημάτων.
 - Δηλαδή γραμμές που προκύπτουν είτε από το 1ο ερώτημα, είτε από το 2ο ερώτημα, είτε και από τα δύο.
 - Εξ'ορισμού στην ένωση, αν μια εγγραφή υπάρχει στα αποτελέσματα και των δύο ερωτημάτων, τότε θα συμπεριληφθεί στο τελικό αποτέλεσμα μία φορά.
 - Η **UNION ALL**:
 - συμπεριλαμβάνει τις κοινές εγγραφές δύο φορές στο τελικό αποτέλεσμα.
 - (και οι παραπάνω επισημάνσεις ισχύουν αντίστοιχα αν κάνουμε ένωση σε παραπάνω από δύο αποτελέσματα ερωτημάτων)
- Πρέπει να ισχύουν:
- Κάθε ερώτημα πρέπει να επιστρέφει ίσο πλήθος στηλών.
 - και συνηθίζεται οι επιλεγόμενες στήλες να είναι ίδιες ή τουλάχιστον παρόμοιες.

Παρατήρηση:

- Ως τελικά ονόματα στηλών, επιλέγονται αυτά που έχουμε θέσει στον πρώτο πίνακα.

1.1. Ένωση

Παράδειγμα 1.1: DB: world, union.sql

```
SELECT Code, Name, Continent
FROM country
WHERE name like 'Ar%'
UNION ALL
SELECT Code, Name, Continent
FROM country
WHERE name like '%nia';
```

Παρατήρηση:

- Μπορούμε να έχουμε ORDER BY μόνο στο τέλος (όχι στα επιμέρους υποερωτήματα).
- Σε άλλα ΣΔΒΔ υπάρχουν και άλλες συνολοθεωρητικές πράξεις, όπως η τομή και η διαφορά, αλλά όχι στη MySQL.

Παράδειγμα 1.2: DB: classic, union.sql

```
SELECT contactLastName AS last_name,
       contactFirstName AS first_name,
       addressLine1 AS address, city, country
FROM customers
UNION
SELECT e.lastName, e.firstName,
       o.addressLine1, o.city, o.country
FROM employees e JOIN offices o
       ON e.officeCode = o.officeCode
ORDER BY 1, 2
```

ΜΑΘΗΜΑ 1.8: ΕΝΩΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

Άσκηση 1: DB: world

Κατασκευάστε έναν πίνακα ο οποίος θα περιέχει τις στήλες:

- country: Όνομα χώρας
- city: Όνομα πόλης
- population: πληθυσμός
- description: Περιγραφή: παίρνει δύο τιμές: 'Capital' (πρωτεύουσα) και 'Largest city' (μεγαλύτερη σε πληθυσμό πόλη)

Οραματιστείτε τη λύση ως την ένωση δύο ερωτημάτων που

- το ένα να υπολογίζει την πρωτεύουσα του κράτους
- το άλλο να υπολογίζει την μεγαλύτερη (σε πληθυσμό) πόλη

Αν για μία χώρα δεν υπάρχουν πόλεις στον πίνακα city, τότε οι χώρες να εμφανίζονται κανονικά, με NULL στη 2η και τη 3η στήλη. Τελικά ο πίνακας να εμφανίζεται ταξινομημένος πρώτα κατά τη 1η και έπειτα κατά τη 4η στήλη.

Πρότυπο της απάντησης:

country	city	population	description
...
Antarctica	NULL	NULL	Capital
Antarctica	NULL	NULL	Largest City
...
Belgium	Bruxelles	133859	Capital
Belgium	Antwerpen	446525	Largest City

Ασκήσεις

Άσκηση 2: DB: classic

Κατασκευάστε πίνακα που να περιέχει για κάθε παραγγελία του πίνακα orders τις εξής πληροφορίες:

order_number	customer_name	total_amount	description
10165	Dragon Souvenirs, Ltd.	67392.85	Shipped on 2003-12-26
...
10407	The Sharp Gifts Warehouse	52229.55	Pending..
...
10391	Anna's Decorations, Ltd	29848.52	Shipped on 2005-03-15
10422	Diecast Classics Inc.	5849.44	Pending..

Καθοδηγηθείτε από τη στήλη description, που πρέπει τα δεδομένα της να εμφανίζονται ακριβώς με τον ίδιο τρόπο, είτε για διεκπεραιωμένες παραγγελίες (Shipped on XXXX-XX-XX), είτε για οποιαδήποτε άλλη κατηγορία (Pending...)
Η ταξινόμηση των εγγραφών να γίνει με βάση το συνολικό ποσό της παραγγελίας σε φθίνουσα σειρά.

ΜΑΘΗΜΑ 1.8: ΕΝΩΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

Η συνάρτηση IF

- Συντακτικό:

```
IF(test-expr, true-expr, false-expr)
```

- Ελέγχεται η λογική έκφραση test-expr:
 - Αν είναι αληθής, επιστρέφεται η true-expr
 - Αν είναι ψευδής, επιστρέφεται η false-expr

Σημείωση:

- Η IF μπορεί να χρησιμοποιηθεί σε οποιοδήποτε clause
 - αλλά το συνηθέστερο είναι να χρησιμοποιείται στη SELECT, για την τροποποίηση των τελικών τιμών ενός ερωτήματος.

Η συνάρτηση IFNULL

- Συντακτικό:

```
IFNULL(test-expr, null-expr)
```

- Ελέγχεται αν η έκφραση test-expr είναι NULL:
 - Αν όχι, επιστρέφεται η test-expr,
 - Αν ναι, επιστρέφεται η null-expr

Σημείωση:

- Η IFNULL είναι απλά μια συντακτική ευκολία
 - αφού μπορεί να γραφεί ισοδύναμα με χρήση μόνο της IF:

```
IF(test-expr IS NULL, null-expr, test-expr)
```

2.1. IF και IFNULL

Παράδειγμα 2.1: DB: classic, if.sql

```
SELECT IF(status='Shipped', 'Shipped', 'Not Shipped') AS status,
       COUNT(*) AS total
FROM orders
GROUP BY 1;
```

Παράδειγμα 2.2:

```
SELECT c.customerName AS customer,
       IFNULL(CONCAT(e.firstName, ' ', e.lastName),
              'None') AS company_employee
FROM customers c LEFT JOIN employees e
  ON c.salesRepEmployeeNumber = e.employeeNumber
ORDER BY 2;
```

Η συνάρτηση NULLIF

- Συντακτικό:

```
NULLIF(expr1, expr2)
```

- Ελέγχεται αν expr1=expr2:
 - Αν ναι, επιστρέφεται NULL, αλλιώς η expr1
- και η NULLIF μπορεί να αντικατασταθεί με χρήση της IF:

```
IF(expr1=expr2, NULL, expr1)
```

ΜΑΘΗΜΑ 1.8: ΕΝΩΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

Η συνάρτηση CASE

- ΣΥΝΤΑΚΤΙΚΟ:

```
CASE expression
  WHEN exp1 THEN res1
  [WHEN exp2 THEN res2]
  ...
  [ELSE resElse]
END AS alias
```

- Διαδοχικά:
 - Αν expression=exp1, επιστρέφεται το res1
 - Αν expression=exp2, επιστρέφεται το res2
 - ...
 - Αλλιώς επιστρέφεται το resElse

Παράδειγμα 3.1: DB: classic, case.sql

```
SELECT CASE productLine
  WHEN 'Classic Cars' THEN 'Cars'
  WHEN 'Vintage Cars' THEN 'Cars'
  WHEN 'Trucks and Buses' THEN 'Cars'
  ELSE 'Other'
END AS productLine, COUNT(*) AS total
FROM productlines
GROUP BY 1;
```

2.2. CASE και COALESCE

Εναλλακτικό συντακτικό της CASE

- Αν παραλείψουμε το expression:

```
CASE
  WHEN exp1 THEN res1
  ...
```

- Ελέγχονται διαδοχικά οι exp1, exp2, ...
 - και όταν βρεθεί η πρώτη που είναι true επιστρέφεται το αντίστοιχο αποτέλεσμα

Παράδειγμα 3.2:

SELECT CASE

```
      WHEN orderDate>='2003-01-01'
        AND orderDate<='2003-12-31' THEN '2003'
      WHEN orderDate>='2004-01-01'
        AND orderDate<='2004-12-31' THEN '2004'
      ELSE 'Other'
    END AS order_date, COUNT(*)
FROM orders
GROUP BY 1;
```

Τέλος η COALESCE:

```
COALESCE (expr1, [expr2, [...]])
```

- Επιστρέφει την πρώτη από τις expr1, expr2,... που δεν είναι NULL.

ΜΑΘΗΜΑ 1.8: ΕΝΩΣΗ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

Για την κατασκευή ενός περίπλοκου ερωτήματος:

- Διασπάμε το πρόβλημα σε επιμέρους προβλήματα
- Βρίσκουμε τους ενδιαμέσους πίνακες που θα πρέπει να κατασκευάσουμε
 - και γράφουμε τα ερωτήματα για τον υπολογισμό τους
- Προχωράμε στην κατασκευή:
 - Πρώτα τα υποερωτήματα,
 - Έπειτα το κυρίως ερώτημα.

Παράδειγμα 4.1: DB: classic, complex.sql

Ο παρακάτω πίνακας:

country	vehicleType	totalAmount
USA	Ground Vehicles	2791592.72
France	Ground Vehicles	2349478.74
USA	Air Vehicles	265777.09
France	Air Vehicles	247058.27
France	Sea Vehicles	215758.94
USA	Sea Vehicles	180746.31

περιέχει το συνολικό ποσό (*totalAmount*) από παραγγελίες που έχουν διεκπεραιωθεί από υπάλληλους στα γραφεία μας στις ΗΠΑ και τη Γαλλία.

Παρατηρήστε ότι οι τύποι των οχημάτων έχουν κατηγοριοποιηθεί με νέο τρόπο σε σχέση με αυτόν στον πίνακα *productLines* (οχήματα στεριάς, ουρανού και θάλασσας)

3. Περίπλοκα Ερωτήματα



(βλ. βίντεο για την κατασκευή της λύσης σε στάδια)

```
SELECT country, vehicleType, sum(price) AS totalAmount
FROM (SELECT o.orderNumber, o.customerNumber,
            od.quantityOrdered * priceEach AS price,
            CASE pl.productLine
              WHEN 'Classic Cars' THEN 'Ground Vehicles'
              WHEN 'Motorcycles' THEN 'Ground Vehicles'
              WHEN 'Planes' THEN 'Air Vehicles'
              WHEN 'Ships' THEN 'Sea Vehicles'
              WHEN 'Trains' THEN 'Ground Vehicles'
              WHEN 'Trucks and Buses' THEN 'Ground Vehicles'
              WHEN 'Vintage Cars' THEN 'Ground Vehicles'
            END AS vehicleType
FROM orderdetails od
JOIN orders o ON o.orderNumber = od.orderNumber
JOIN products p ON od.productCode = p.productCode
JOIN productlines pl ON p.productLine = pl.productLine
WHERE status = 'Shipped') ovt
JOIN (SELECT c.customerNumber, o.country
FROM customers c JOIN employees e
ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
WHERE o.country IN ('USA', 'France')) oc
ON ovt.customerNumber = oc.customerNumber
GROUP BY country, vehicleType
ORDER BY totalAmount DESC;
```