

Βάσεις Δεδομένων – 6

SQL – A Relational Database Language – Μέρος Γ

Δρ. Χαράλαμπος Καρανίκας

Επεξεργασία δεδομένων με SQL

SQL Data Manipulation Language

Self JOIN, UNION, GROUP BY, HAVING, EXISTS,
ANY and ALL Operators, AUTO INCREMENT, VIEW

Self JOIN

- Μια self JOIN είναι μια κανονική σύνδεση, αλλά ο πίνακας συνδέεται με τον εαυτό του.

```
SELECT column_name(s)  
FROM table1 T1, table1 T2  
WHERE condition;
```

```
SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
FROM Customers AS A, Customers AS B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City  
ORDER BY A.City;
```

UNION Operator 1/4

- Ο τελεστής UNION χρησιμοποιείται για να ενώσει το σύνολο αποτελεσμάτων δύο ή περισσότερων εντολών SELECT.
 - Κάθε εντολή SELECT στο πλαίσιο της UNION πρέπει να έχει τον ίδιο αριθμό στηλών
 - Οι στήλες πρέπει επίσης να έχουν παρόμοιους τύπους δεδομένων
 - Οι στήλες σε κάθε δήλωση SELECT πρέπει επίσης να είναι στην ίδια σειρά

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

UNION Operator 2/4

- Ο τελεστής UNION επιλέγει από προεπιλογή μόνο μοναδικές τιμές. Για να επιτρέψετε διπλές τιμές, χρησιμοποιήστε το UNION ALL

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```

- Τα ονόματα στηλών στο σύνολο αποτελεσμάτων είναι συνήθως ίδια με τα ονόματα στηλών στην πρώτη εντολή SELECT της JOIN.

UNION Operator 3/4

```
SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;
```

```
SELECT City FROM Customers  
UNION ALL  
SELECT City FROM Suppliers  
ORDER BY City;
```

UNION Operator 4/4

```
SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;
```

```
SELECT 'Customer' As Type, ContactName, City, Country  
FROM Customers  
UNION  
SELECT 'Supplier', ContactName, City, Country  
FROM Suppliers;
```

GROUP BY 1/3

- Η εντολή GROUP BY χρησιμοποιείται συχνά με τις συναρτήσεις COUNT, MAX, MIN, SUM, AVG για την ομαδοποίηση των αποτελεσμάτων από μία ή περισσότερες στήλες.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

GROUP BY 2/3

- Ο αριθμός των πελατών σε κάθε χώρα

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country;
```

- Ο αριθμός των πελατών σε κάθε χώρα, ταξινομημένος υψηλός σε χαμηλός

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

GROUP BY 3/3

- Τον αριθμό των παραγγελιών που αποστέλλονται από κάθε αποστολέα

```
SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM  
Orders  
LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID  
GROUP BY ShipperName;
```

SQL HAVING 1/3

- Το HAVING προστέθηκε στην SQL επειδή η λέξη WHERE δεν μπορούσε να χρησιμοποιηθεί με συναρτήσεις.
- Οι συνθήκες στο HAVING εφαρμόζονται μετά τον σχηματισμό ομάδων, ενώ οι συνθήκες στο WHERE εφαρμόζονται πριν από τη διαμόρφωση των ομάδων

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
HAVING condition  
ORDER BY column_name(s);
```

SQL HAVING 2/3

- Αριθμός των πελατών σε κάθε χώρα, με μόνο χώρες με περισσότερους από 5 πελάτες

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5;
```

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;
```

SQL HAVING 3/3

- Οι υπάλληλοι που έχουν καταγράψει περισσότερες από 10 παραγγελίες

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders
FROM (Orders
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID)
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 10;
```

- Παρουσιάζει εάν οι εργαζόμενοι "Davolio" ή "Fuller" έχουν καταγράψει περισσότερες από 25 παραγγελίες

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders
FROM Orders
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
WHERE LastName = 'Davolio' OR LastName = 'Fuller'
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 25;
```

SQL EXISTS Operator 1/2

- Ο τελεστής EXISTS χρησιμοποιείται για να ελέγξει την ύπαρξη οποιασδήποτε εγγραφής σε υποερώτημα. Επιστρέφει true αν το υποερώτημα επιστρέφει μία ή περισσότερες εγγραφές.

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name W
HERE condition);
```

SQL EXISTS Operator 2/2

- Επιστρέφει TRUE και απαριθμεί τους προμηθευτές με τιμή προϊόντος μικρότερη από 20

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products WHERE SupplierId =
Suppliers.supplierId AND Price < 20);
```

- Επιστρέφει TRUE και απαριθμεί τους προμηθευτές με τιμή προϊόντος ίση με 22

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products WHERE SupplierId =
Suppliers.supplierId AND Price = 22);
```

ANY and ALL Operators 1/2

- Οι τελεστές ANY και ALL χρησιμοποιούνται με το WHERE ή το HAVING.
 - Ο ANY επιστρέφει true, αν κάποια από τις τιμές των υποερωτήσεων πληρεί τη συνθήκη.
 - Ο ALL επιστρέφει true αν όλες οι τιμές subquery πληρούν την συνθήκη.

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
(SELECT column_name FROM table_name WHERE
condition);
```

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
(SELECT column_name FROM table_name WHERE
condition);
```

ANY and ALL Operators 2/2

- Επιστρέφει TRUE και παραθέτει τα ονόματα των προϊόντων αν εντοπίσει ΟΠΟΙΕΣΔΗΠΟΤΕ εγγραφές στον πίνακα OrderDetails ότι η ποσότητα = 10

```
SELECT ProductName  
FROM Products  
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE  
Quantity = 10);
```

- Επιστρέφει TRUE και παραθέτει τα ονόματα προϊόντων αν όλες οι εγγραφές στον πίνακα OrderDetails έχουν ποσότητα = 10

```
SELECT ProductName  
FROM Products  
WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE  
Quantity = 10);
```

AUTO INCREMENT

- AUTO INCREMENT (αυτόματη αύξηση) επιτρέπει την αυτόματη δημιουργία ενός μοναδικού αριθμού όταν μια νέα εγγραφή εισάγεται σε έναν πίνακα.
- Συχνά αυτό είναι το πρωτεύον πεδίο κλειδιού που θα θέλαμε να δημιουργούμε αυτόματα κάθε φορά που εισάγεται μια νέα εγγραφή.

AUTO INCREMENT - MySQL

- Η ακόλουθη πρόταση SQL ορίζει τη στήλη "ID" ως πεδίο πρωτεύοντος κλειδιού αυτόματης προσαύξησης στον πίνακα Persons.
- Από προεπιλογή, η τιμή εκκίνησης για το AUTO_INCREMENT είναι 1 και θα αυξηθεί κατά 1 για κάθε νέα εγγραφή.

```
CREATE TABLE Persons (  
    ID int NOT NULL AUTO_INCREMENT,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);
```

AUTO INCREMENT - MySQL

- Για να αφήσετε την ακολουθία AUTO INCREMENT να ξεκινήσει με μια άλλη τιμή, χρησιμοποιήστε την ακόλουθη πρόταση SQL:

```
ALTER TABLE Persons AUTO_INCREMENT=100;
```

- Για να εισαγάγετε μια νέα εγγραφή στον πίνακα Person, ΔΕΝ θα πρέπει να καθορίσετε μια τιμή για τη στήλη "ID" (μια μοναδική τιμή θα προστεθεί αυτόματα)

CREATE VIEW

- Στην SQL, ένα **View** είναι ένας εικονικός πίνακας με βάση το σύνολο αποτελεσμάτων μιας εντολής SQL.
- Τα πεδία σε ένα **View** είναι πεδία από έναν ή περισσότερους πραγματικούς πίνακες στη βάση δεδομένων.
- Παρέχει έναν μηχανισμό για την απόκρυψη ορισμένων δεδομένων από κάποιους χρήστες.
- Μπορείτε να προσθέσετε εντολές SQL, WHERE και JOIN σε ένα **View** και να παρουσιάσετε τα δεδομένα σαν να προέρχονταν από ένα μόνο πίνακα.

CREATE VIEW

- Ένα **View** εμφανίζει πάντα ενημερωμένα δεδομένα. Η βάση δεδομένων επαναδημιουργεί τα δεδομένα, χρησιμοποιώντας τη δήλωση SQL του **View**, κάθε φορά που ένας χρήστης ερωτά μια προβολή.
- Οποιαδήποτε relation (σχέση) δεν είναι του εννοιολογικού μοντέλου αλλά γίνεται ορατή σε ένα χρήστη ως "εικονική σχέση" ονομάζεται view.

CREATE VIEW

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
CREATE VIEW [Current Product List] AS  
SELECT ProductID, ProductName  
FROM Products  
WHERE Discontinued = No;
```

```
SELECT * FROM [Current Product List];
```

```
DROP VIEW view_name;
```