SQL

ΣΗΜΕΙΩΣΕΙΣ ΜΑΘΗΜΑΤΟΣ ΠΛΣ60

2022

SQL

SQL – Structured Query Language

SELECT

```
SELECT * FROM T1 LIMIT 10;
```

```
SELECT TOP 10 * FROM T1;
```

```
SELECT TOP 50 PERCENT * FROM T1;
```

```
SELECT DISTINCT c1, c2 FROM T1;
```

το οποίο επιστρέφει τα διακριτά ζεύγη c1,c2

```
SELECT COUNT(*) FROM T1;
```

```
SELECT COUNT(c1) FROM T1;
```

```
SELECT COUNT(DISTINCT c1) FROM T1;
```

OPERATORS

< > not equal

In some SQL versions, != is used to represent not equal as well.

AND, OR, NOT, IN, BETWEEN, LIKE

DATE

2022-02-15 12:24:56

COMMENTS

--comment

/* comment */

IN

```
SELECT * FROM T1
```

```
WHERE c1 IN (v1, v2, …, vn)
```

or

```
SELECT * FROM T1
```

```
WHERE c1 IN (subquery)
```

Επίσης, αντί για IN μπορεί να έχουμε και NOT IN

BETWEEN

```
SELECT * FROM T1
```

```
WHERE c1 BETWEEN value1 AND value2
```

LIKE

```
WHERE c1 LIKE 'DATA%'
```

```
WHERE c1 NOT LIKE 'DATA%'
```

Το % είναι μπαλαντέρ για πολλούς χαρακτήρες.

Το _ είναι μπαλαντέρ για έναν χαρακτήρα.

[oa] επιλέγει μόνον ένα γράμμα κάθε φορά. Παράδειγμα, h[oa]t βρίσκει τις λέξεις hot και hat.

[^oa] το αντίθετο του παραπάνω

[a-c] όπως το παραπάνω για όλους τους χαρακτήρες από a ως και c.

AGGREGATE FUNCTIONS

AVG MAX MIN SUM COUNT

```
SELECT SUM(c1) FROM T1;
```

ORDER BY

```
ORDER BY c1 ASC, c2 DESC;
```

INSERT

```
INSERT INTO T1(c1, c2, …, cn)

VALUES(v1, v2, …, vn)
```

Η παρένθεση με τα attributes μπορεί να παραληφθεί όταν πρόκειται για όλα.

```
INSERT INTO T1

SELECT c1 FROM oldtable

WHERE condition;
```

όπου οι πίνακες newtable, oldtable είναι συμβατοί και προϋπάρχουν.

```
SELECT c1

INTO newtable

FROM oldtable

WHERE condition;
```

όπου οι πίνακες newtable, oldtable είναι συμβατοί. Λαμβάνει δεδομένα από τον oldtable, δημιουργεί τον newtable και τα εναποθέτει.

UPDATE

```
UPDATE T1

SET c1 = value1, c2 = value2

WHERE condition;
```

DELETE

```
DELETE FROM T1

WHERE condition;
```

ALIAS

```
SELECT c1 AS C

FROM tableR AS W
```

ή

```
SELECT c1 AS C

FROM tableR W
```

JOIN

```
SELECT c1

FROM T1

INNER JOIN S

ON condition
```

Επίσης, αντί για INNER JOIN έχουμε:

```
LEFT OUTER JOIN

RIGHT OUTER JOIN

FULL OUTER JOIN

NATURAL JOIN (χωρίς ON)

CROSS JOIN
```

Το παρακάτω είναι το SELF JOIN:

```
SELECT A.c1, B.c2

FROM tableR A, tableR B

WHERE condition;
```

Αντί για το CROSS JOIN μπορούμε να έχουμε το select πολλαπλών πινάκων:

```
SELECT c1, c2,…

FROM T1, T2

WHERE condition;
```

Επίσης, το INNER JOIN δεν είναι τίποτε άλλο παρά ένα CROSS JOIN με συνθήκη.

GROUP BY

```
SELECT aggregate function(c1)

FROM T1

WHERE condition

GROUP BY c2

HAVING condition1;
```

Κάνει ομαδοποίηση των δεδομένων κατά c2, κρατάει μόνον τις ομάδες στις οποίες ισχύει η συνθήκη του HAVING και σε κάθε ομάδα εφαρμόζει την aggregate function και επιστρέφει το αποτέλεσμα.

Γενικότερα, μπορούμε να κάνουμε ομαδοποίηση ως προς περισσότερα του ενός χαρακτηριστικά.

Επίσης, μπορούμε να εφαρμόσουμε το GROUP BY σε ένα cross join, αρκεί τα χαρακτηριστικά που επιστρέφουμε να είναι αυτά ως προς τα οποία κάνουμε ομαδοποίηση ή να ανήκουν σε aggregate function.

UNION INTERSECT MINUS

Οι πίνακες T1, T2 πρέπει να είναι συμβατοί.

```
SELECT c1 FROM T1

UNION/INTERSECT/MINUS

SELECT c2 FROM T2
```

Στην MS SQL αντί για MINUS γράφουμε EXCEPT.

Διαφορά μεταξύ INNER JOIN και INTERSECT: Η τομή είναι πράξη συνόλων και δεν επιτρέπει διπλότυπα, αλλά μπορεί να έχει null τιμές. Αντιθέτως, το INNER JOIN μπορεί να έχει διπλότυπα, αλλά όχι NULL τιμές.

EXISTS

```
SELECT c1

FROM T1

WHERE EXISTS (subquery)
```

Αν το subquery επιστρέφει δεδομένα, τότε είναι σαν να επιστρέφει TRUE. Αλλιώς, FALSE.

ANY/ALL

```
SELECT c1

FROM T1

WHERE c2  [op] ANY (subquery)

          [op] ALL (subquery)
```

όπου [op] είναι ένας τελεστής ισότητας ή ανισότητας.

Στο ANY πρέπει να ισχύει η συνθήκη για μία τουλάχιστον πλειάδα στο subquery.

Στο ALL πρέπει να ισχύει για όλες τις πλειάδες.

CASE

```
SELECT c1, c2,

CASE

      WHEN cond1 THEN value1

      WHEN cond2 THE value2

      ELSE value3

END AS c3

FROM T1

WHERE condition;
```

NULL

```
IS NULL

IS NOT NULL
```

Συνάρτηση `ISNULL(c, value)`

Αν το c δεν είναι NULL, τότε η συνάρτηση επιστρέφει την τιμή του. Αλλιώς, η συνάρτηση επιστρέφει την τιμή value.

DATABASE

```
CREATE DATABASE D1;

DROP DATABASE D1;


BACKUP DATABASE D1

TODISK='path'

WITH DIFFERENTIAL;
```

CREATE TABLE

```
CREATE TABLE T1(
      c1 type [keyword],
      c2 type [keyword],
      PRIMARY KEY(c3, c4,…),
      FOREIGN KEY(c5, c6) REFERENCES T3(…),
      CONSTRAINT …
);
```

όπου τα keyword μπορεί να είναι:

- NOT NULL
- UNIQUE
- DEFAULT value
- CHECK (condition)
- AUTO_INCREMENT
- PRIMARY KEY
- FOREIGN KEY REFERENCES T2(attribute)

Για τον ορισμό του CONSTRAINT γράφουμε:

```
CONSTRAINT PK_name PRIMARY KEY(c1, c2)

CONSTRAINT FK_name FOREING KEY(c1, c2) REFERENCES T2(c3, c4)
```

Ειδικά για τα foreign keys ισχύουν τα εξής:

```
ON UPDATE/DELETE SET NULL

ON UPDATE/DELETE SET DEFAULT

ON UPDATE/DELETE CASCADE
```

που σημαίνει πως, όταν αλλάξει ή διαγραφεί το primary key, τότε το αντίστοιχε foreign key:

- αλλάζει/διαγράφεται (CASCADE)
- γίνεται null (SET NULL)
- λαμβάνει την default τιμή του (SET DEFAULT)

DROP TABLE

```
DROP TABLE T1;
```

TRUNCATE

```
TRUNCATE TABLE T1;
```

Διαγράφει όλα τα δεδομένα του πίνακα, αλλά όχι τον ίδιο τον πίνακα.

ALTER TABLE

```
ALTER TABLE T1

ADD COLUMN c1 type;

DROP COLUMN c1 type;

ALTER/MODIFY COLUMN c1 type;

ADD UNIQUE(c1);

ADD/DROP CONSTRAINT…

ADD/DROP FOREING KEY…
```

INDEX

```
CREATE INDEX index_name

ON T1(c1, c2);
```

όπου η σειρά των c1, c2 παίζει ρόλο.

```
DROP INDEX index_name;
```

VIEW

```
CREATE VIEW View_name AS

CREATE OR REPLACE VIEW View_name AS

DROP VIEW View_name
```

TRIGGERS

```
CREATE TRIGGER TriggerName

ON tableName

BEFORE/AFTER INSERT/UPDATE/DELETE

AS


CREATE TRIGGER reminder1

ON Sales.Customer

AFTER INSERT, UPDATE

AS RAISERROR ('Notify Error', 16, 10);

GO
```

DML triggers use the **deleted** and **inserted** logical (conceptual) tables. They're structurally like the table on which the trigger is defined, that is, the table on which the user action is tried. The deleted and inserted tables hold the old values or new values of the rows that may be changed by the user action.

AUTO INCREMENT

`AUTO_INCREMENT` ως keyword δίπλα από τον τύπο του ορίσματος στο CREATE TABLE.

Στην MS SQL χρησιμοποιούμε το `IDENTITY(1,1)`

VARIABLES

```
DECLARE @MYVARIABLE type;

SET @MYVARIABLE = value1;
```

IF/ELSE

```
IF condition
        statement;
ELSE
        statement;
```

LINQ

Method syntax LINQ

Standard Query Operators (tutorialsteacher.com)

New method Include()

If you have the standard SQL query,

```
SELECT * FROM Customers;
```

```
var customers = db.Customers.ToList();
```

then it can be written as shown above. In the same manner, the following SQL query

```
SELECT *
FROM Customers
INNER JOIN Orders
ON Customers.Id = Orders.CustomerId;
```

```
var customersWithOrderDetail =
db.Customers.Include("Orders").ToList();
```

becomes as shown above with the use of the new LINQ method Include().