



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΜΕΡΟΣ ΤΡΙΤΟ

Foreign key, Index, DML Ερωτήματα

Γιώργος Μαρκομανώλης

Περιεχόμενα

Foreign Key	1
Index	4
DML Ερωτήματα	6
INSERT	6
SELECT	6
Συναρτήσεις	9

Foreign Key

Ένα foreign key είναι μια στήλη ή ένα σύνολο στηλών σε ένα πίνακα το οποίο αντιστοιχεί με την ίδια σειρά σε μια στήλη ή σε ένα σύνολο στηλών αντίστοιχα που είναι ορισμένες σαν primary key ή unique.

Για να οριστεί τέτοιου είδους περιορισμός πρέπει να έχετε δημιουργήσει δύο πίνακες που στον έναν θα έχετε δηλωμένες τις στήλες που θέλετε σαν primary key ή unique και στον άλλον πίνακα που θα είναι το foreign key θα υπάρχουν στήλες που θα αντιστοιχούν στις στήλες του πρώτου πίνακα.

Αν θεωρήσουμε λοιπόν το αυτό το μοντέλο, δηλαδή ότι έχουμε δύο πίνακες που τον έναν τον ονομάζουμε parent και τον άλλον child, τότε οι γραμμές που είναι foreign key στον πίνακα child εξαρτώνται από τις γραμμές του parent. Το foreign key έχει τους ακόλουθους περιορισμούς:

- Απαιτεί ότι οι τιμές που είναι στις στήλες του πίνακα child που αποτελούν το foreign key, πρέπει να υπάρχουν στις στήλες του primary key που αποτελούν το primary key ή unique.
- Απαγορεύεται να διαγράψετε μια στήλη από τον πίνακα parent ή να αλλάξετε την τιμή από μια εγγραφή της οποίας το γνώρισμα είναι συνδεδεμένο με foreign key και υπάρχει και η αντίστοιχη τιμή στον πίνακα child.

Για να δημιουργήσετε έναν περιορισμό πρέπει να δημιουργήσετε πρώτα τον πίνακα parent. Σε αυτό τον πίνακα θα ορίσετε το primary key και αν υπάρχει και περιορισμός unique για την σύνδεση των πινάκων. Εμείς θα κάνουμε ένα παράδειγμα με τους πίνακες υπάλληλοι και τμήματα.

```
CREATE TABLE EMPLOYEE (
EMPLOYEE_ID BIGINT NOT NULL,
NAME VARCHAR(20),
SURNAME VARCHAR(40),
CONSTRAINT PK_EMPLOYEE PRIMARY KEY(EMPLOYEE_ID));
```

```
CREATE TABLE DEPARTMENTS (
DEPARTMENT_ID BIGINT NOT NULL,
NAME CHAR(30) NOT NULL,
DIRECTOR_ID BIGINT NOT NULL,
CONSTRAINT PK_DEPARTMENTS PRIMARY KEY(DEPARTMENT_ID));
```

Για να ορίσετε ένα Foreign Key στον πίνακα DEPARTMENTS πρέπει να πληκτρολογήσετε την ακόλουθη εντολή:

```
ALTER TABLE DEPARTMENTS
  ADD CONSTRAINT FK_DEPARTMENTS_EMPLOYEE
  FOREIGN KEY (DIRECTOR_ID)
  REFERENCES EMPLOYEE(EMPLOYEE_ID);
```

Η σύνταξη της εντολής για foreign key είναι η εξής:

```
FOREIGN KEY(Όνομα_στήλης)
REFERENCES όνομα_πίνακα_parent(Όνομα_στήλης)
[ON DELETE {NO ACTION | CASCADE | SET NULL | DEFAULT}]
[ON UPDATE {NO ACTION | CASCADE | SET NULL | DEFAULT}];
```

Με την παραπάνω εντολή δημιουργείται ένα foreign key με την στήλη που είναι μέσα στην παρένθεση και αναφέρετε στην στήλη που είναι στον πίνακα μετά την λέξη REFERENCES. Επίσης αν θέλετε ορίζεται και τις λειτουργίες που θα γίνονται όταν διαγράψετε ή ενημερώνετε κάποια τιμή της στήλης του πίνακα στον οποίο αναφέρετε ο περιορισμός.

Δηλαδή:

Με την εντολή:

ON DELETE CASCADE

ορίζουμε ότι αν διαγράψουμε μια τιμή από την στήλη EMPLOYEE_ID του πίνακα EMPLOYEE και υπάρχει η ίδια τιμή σε στήλη του DIRECTOR στον πίνακα DEPARTMENTS, τότε θα διαγραφούν όλες οι εγγραφές από τον πίνακα DEPARTMENTS που έχουν την συγκεκριμένη τιμή.

Αν αντί για **ON DELETE CASCADE** βάλετε **ON DELETE NO ACTION** τότε δηλώνετε ότι όταν διαγράψετε εγγραφές από τον πίνακα EMPLOYEE, δεν γίνονται αλλαγές στον πίνακα DEPARTMENTS. Αν δηλώσετε **ON DELETE SET NULL** τότε όταν διαγράψετε εγγραφές από τον πίνακα EMPLOYEE, καταχωρείται η τιμή NULL στις αντίστοιχες τιμές της στήλης DIRECTOR. Αν δηλώσετε **ON DELETE DEFAULT** τότε όταν διαγράψετε εγγραφές από τον πίνακα EMPLOYEE, καταχωρείται η DEFAULT τιμή στις αντίστοιχες τιμές της στήλης DIRECTOR.

Ομοίως για το **ON UPDATE** απλώς αυτή η εντολή αφορά την ενημέρωση και όχι την διαγραφή εγγραφών.

Προσοχή: Το αν θα επιλέξετε CASCADE ή κάποια άλλη εντολή εξαρτάται πάντα από το νόημα που έχουν οι στήλες, δηλαδή στο συγκεκριμένο παράδειγμα αν δηλώσουμε την εντολή ON DELETE CASCADE τότε αν διαγράψουμε έναν υπάλληλο από τον πίνακα

EMPLOYEE που είναι διευθυντής σε ένα τμήμα, τότε θα διαγραφεί και ο διευθυντής αλλά και ολόκληρο το τμήμα με αποτέλεσμα να δημιουργηθεί σφάλμα στην βάση. Παρατηρήστε ότι ο τρόπος σύνδεσης των πίνακα των υπαλλήλων με τον πίνακα των τμημάτων είναι ένα προς πολλά γιατί μπορείτε να βάλετε τον ίδιο διευθυντή σε πολλά τμήματα. Αν όμως θέλουμε να έχουμε έναν διευθυντή ανά τμήμα (το πιο λογικό) θα πρέπει να δηλώσετε την στήλη DIRECTOR σαν unique constraint.

Δώστε την εντολή:

```
ALTER TABLE DEPARTMENTS  
ADD CONSTRAINT DIRECTOR_UNIQ UNIQUE(DIRECTOR_ID);
```

Τώρα πλέον γίνεται να βάλετε μόνο έναν διευθυντή ανά τμήμα, άρα η σχέση των υπαλλήλων με τα τμήματα είναι ένα προς ένα.

Index

Το index είναι μια ιδιότητα του πίνακα η οποία μπορεί να δηλωθεί σε μια στήλη ή σε ένα σύνολο στηλών ώστε να επιταχυνθεί η εύρεση σειρών από τον πίνακα. Το index είναι σαν ένας σελιδοδείκτης όπου βρίσκει κατευθείαν την σελίδα που ψάχνετε χωρίς να πρέπει να διαβάσετε όλο το βιβλίο.

Μπορείτε να φτιάξετε μέχρι 256 index για κάθε πίνακα. Όταν δηλώνετε έναν περιορισμό όπως primary key, foreign key, unique, τότε αυτόματα δημιουργείται ένα index για τις στήλες που δηλώνεται τους περιορισμούς. Επίσης στο index έχετε την δυνατότητα να ορίσετε Ascending ώστε τα δεδομένα του index να μπαίνουν σε σειρά από την μικρότερη τιμή προς την μεγαλύτερη και Descending για να μπαίνουν οι τιμές σε σειρά από το μεγαλύτερο προς το μικρότερο.

Ποιες στήλες να κάνετε index

Συνήθως θα κάνετε index μια στήλη στις ακόλουθες περιπτώσεις:

- Όταν υπάρχουν συνθήκες αναζήτησης που αναφέρονται στην στήλη, όπως BETWEEN, STARTING WITH, CONTAINING. Δεν είναι χρήσιμο το index σε στήλη που αναζητάτε τιμή με το LIKE.
- Είναι χρήσιμο όταν έχουμε πολλά ερωτήματα με την εντολή ORDER BY (θα την δούμε αργότερα) αφού με αυτή την εντολή κάνουμε ταξινόμηση των γραμμών που γίνεται και με το index.
- Μπορείτε να προσθέσετε την δυνατότητα index στις στήλες που πιστεύετε ότι απαιτείται (συνήθως σε αυτές που γίνονται πολλά ερωτήματα και κάνετε συχνή αναζήτηση).

Δεν πρέπει να βάζετε index όταν:

- Ανανεώνονται συχνά οι τιμές της στήλης.
- Είναι προκαθορισμένες οι πιθανές απαντήσεις.

Προσοχή: Δεν είναι σωστό να δημιουργείτε πολλά index σε έναν πίνακα γιατί αντί να βοηθάνε για την πιο γρήγορη εύρεση δεδομένων θα κάνουν την βάση πιο αργή. Να γίνετε χρήση των index με προσοχή.

Δημιουργία Index

Η σύνταξη για την δημιουργία index είναι η ακόλουθη:

```
CREATE [UNIQUE] [ASC | DESC]  
INDEX όνομα_του_index ON όνομα_πίνακα (όνομα_στήλης ή ονόματα_στηλών)
```

Παράδειγμα

Για να δημιουργήσετε ένα Index με το όνομα EMP για την στήλη SURNAME του πίνακα EMPLOYEE δώστε την εντολή:

CREATE INDEX EMP ON EMPLOYEE(SURNAME);

Για να δημιουργήσετε Index για το NAME και SURNAME δώστε την εντολή:

CREATE INDEX EMP ON EMPLOYEE(NAME,SURNAME);

Αν θέλετε να δημιουργήσετε ένα unique index, τότε θα δώσετε την εντολή:

CREATE UNIQUE INDEX EMP ON EMPLOYEE(SURNAME);

Απενεργοποίηση και ενεργοποίηση index

Για να απενεργοποιήσετε το index EMP δώστε την εντολή:

ALTER INDEX EMP INACTIVE;

Για να το ενεργοποιήσετε:

ALTER INDEX EMP ACTIVE;

Διαγραφή Index

Για να διαγράψετε ένα index δώστε την εντολή:

DROP INDEX EMP;

DML Ερωτήματα

Τα αρχικά DML σημαίνουν data manipulation language και εννοούμε την γλώσσα που διαχειρίζεται τα δεδομένα σε μια βάση (εύρεση, εισαγωγή, ενημέρωση και διαγραφή).

INSERT

Για να μπορέσετε να καταχωρήσετε μερικά δεδομένα για να δείτε καλύτερα τα παραδείγματα με την εντολή select θα πούμε για την εντολή Insert.

Σύνταξη εντολής:

*INSERT INTO <όνομα πίνακα ή όνομα όψης> (ονόματα στηλών για τα οποία θα εισάγετε δεδομένα)
VALUES (δεδομένα που θα εισάγετε για κάθε στήλη χωρισμένα με κόμμα)*

Παράδειγμα

Για να καταχωρήσετε έναν υπάλληλο στον πίνακα EMPLOYEE δώστε την εντολή:

*INSERT INTO EMPLOYEE (EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
FATHER_NAME, POLICE_ID, BIRTH_DATE, TELEPHONE, GENDER,
DEPARTMENT_ID, ADDRESS, EMAIL)
VALUES
(1, 'JOHN', 'JELIOS', 'PETER', 'Σ063748', '5.3.1975', '2651012345', 'M', '4', 'ΠΑΝΕΠΙΣΤ
ΗΜΙΟΥ 12', 'john@company.gr');*

SELECT

Με την εντολή SELECT επιλέγουμε κάποια δεδομένα από πίνακες ανάλογα με τα κριτήρια που δηλώνεται μέσα στην εντολή SELECT.

Η γενική σύνταξη της εντολής SELECT είναι η ακόλουθη:

```
SELECT
  [FIRST(m)][SKIP(n)][[ALL]]DISTINCT]
  <λίστα στηλών> [, [όνομα στήλης] | expression | constant] AS όνομα
FROM <πίνακας ή stored procedure ή view>
[ { [[INNER] | [LEFT | RIGHT | FULL] [OUTER]] JOIN } ] <πίνακας ή stored procedure
ή view>
ON <συνθήκες ένωσης> [ { JOIN.. } ]
[WHERE <συνθήκη>]
[GROUP BY <ονόματα στηλών για ομαδοποίηση>]
```


[HAVING <συνθήκη αναζήτησης>]
 [UNION <εντολή select> [ALL]]
 [PLAN <έκφραση plan>]
 [ORDER BY <λίστα στηλών>]
 [FOR UPDATE [OF <ονόματα στηλών>] [WITH LOCK]]

Όλες οι εντολές του select που χρειάζονται άμεσα θα εξηγηθούν στα ακόλουθα μαθήματα.

Παραδείγματα

Για να δείτε όλα τα δεδομένα του πίνακα των υπαλλήλων, αφού έχετε συνδεθεί φυσικά με την βάση, πληκτρολογήστε την εντολή:

SELECT * FROM EMPLOYEE;

Για να δείτε όλους τους αριθμούς μητρώων των υπαλλήλων δώστε την εντολή:

SELECT EMPLOYEE_ID FROM EMPLOYEE;

Αν θέλουμε να δούμε τους αριθμούς μητρώων μαζί με το όνομα και το επώνυμο των υπαλλήλων δώστε την εντολή:

SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME FROM EMPLOYEE;

Αν θέλετε κατά την εμφάνιση των αποτελεσμάτων να έχει διαφορετικό όνομα κάποια στήλη τότε χρησιμοποιείτε την εντολή AS. Δηλαδή αν θέλετε το FIRST_NAME να το βλέπετε σαν NAME και το LAST_NAME σαν SURNAME τότε δώστε την εντολή:

***SELECT EMPLOYEE_ID, FIRST_NAME AS NAME, LAST_NAME AS SURNAME
FROM EMPLOYEE;***

Προσοχή: Μετά την εντολή AS πρέπει να υπάρχει μια συνεχόμενη πρόταση, δηλαδή αν θέλετε μετά το AS να βάλετε τις λέξεις FULL NAME τότε πρέπει να την γράψετε FULL_NAME, δηλαδή να μην υπάρχουν κενά ανάμεσα στις λέξεις.

Επίσης μπορείτε να κάνετε select τις στήλες με όποια σειρά θέλετε.

Αν θέλετε να βρείτε κάποιες εγγραφές αλλά να μην εμφανιστούν δύο ίδιες αλλά κάθε εγγραφή να είναι μοναδική, τότε πληκτρολογήστε την εντολή:

SELECT DISTINCT BIRTH_DATE FROM EMPLOYEE;

Για να βρείτε τα στοιχεία ενός υπαλλήλου που ξέρετε τον αριθμό μητρώου του (έστω ο αριθμός μητρώου είναι 4), πληκτρολογήστε την εντολή:

SELECT * FROM EMPLOYEE WHERE EMPLOYEE_ID=4;

Δηλαδή αν ψάχνετε συγκεκριμένες εγγραφές γράφετε μετά το SELECT τις στήλες που θέλετε να βρείτε ή * για να εμφανιστούν όλες οι στήλες, έπειτα γράφετε from τον πίνακα που θα βρει τις στήλες, γράφεται where και μετά το όνομα της στήλης και κάποιον τελεστή. Θα μπορούσαμε παραπάνω να είχαμε EMPLOYEE_ID>4, !=4... ή θα μπορούσαμε να είχαμε BIRTH_DATE='11/18/2004' AND NAME='JOHN'

Προσοχή με τις ημερομηνίες: Στο παραπάνω ερώτημα με την στήλη BIRTH_DATE η ημερομηνία που έχει καταγραφεί στην βάση είναι 18/11/2004 αλλά όταν θέλετε να βρείτε εγγραφές και έχετε ως κριτήριο την ημερομηνία θα βάζετε πρώτα τον μήνα και μετά την μέρα. Όμως για λόγους ευκολίας μπορείτε να γράφετε την ημερομηνία με . αντί για /, δηλαδή 18.11.2004 όπου με αυτή την μορφή η ημερομηνία γράφεται κανονικά όπως γνωρίζουμε.

Αν θέλετε μετά το where να βάλετε πολλές συνθήκες μπορείτε αναλόγως αν θέλετε να ισχύουν όλες να χωρίζονται με AND ή αν θέλετε να ισχύουν κάποιες από αυτές να χωρίζονται με OR.

Για να διαλέξετε κάποιες εγγραφές με βάση τον αριθμό των εγγραφών, δώστε την εντολή:

SELECT FIRST 2 SKIP 1 EMPLOYEE_ID FROM EMPLOYEE;

Με την παραπάνω εντολή θα εμφανιστούν δύο αριθμοί μητρώων αλλά θα είναι ο δεύτερος και ο τρίτος στον πίνακα αφού δηλώσαμε να μην εμφανίσει έναν αριθμό μητρώου. Αυτή η εντολή έχει σημασία όταν οι εγγραφές είναι ταξινομημένες. Θα δούμε αργότερα την ταξινόμηση.

Έστω ότι θέλετε να βρείτε τους υπαλλήλους που έχουν γεννηθεί από 10/1/1960 μέχρι 10/1/1970, τότε πληκτρολογήστε την εντολή:

***SELECT * FROM EMPLOYEE WHERE BIRTH DATE BETWEEN '1/10/1960'
AND '1/10/1970';***

Μετά το where μπορείτε να χρησιμοποιήσετε:

- BETWEEN...AND...
- CONTAINING
- IN
- LIKE
- STARTING WITH

Συναρτήσεις

Η SQL έχει συναρτήσεις για την διευκόλυνση κάποιων διαδικασιών.

MAX

Με την MAX μπορείτε να βρείτε την μέγιστη τιμή μιας στήλης αλλά και στηλών με κάποια πράξη.

Παράδειγμα

Για να βρείτε την μέγιστη τιμή από μια στήλη με το όνομα Price που έχει πωλήσεις ενός προϊόντος από τον πίνακα που λέγεται SALES, τότε δώστε την εντολή:

```
SELECT MAX(PRICE) FROM SALES;
```

Αν οι τιμές δεν έχουν ΦΠΑ και θέλετε να βρείτε την μέγιστη τιμή με ΦΠΑ, δώστε την εντολή:

```
SELECT MAX(1.18*PRICE) FROM SALES;
```

MIN

Ομοίως με την MAX απλώς βρίσκει την μικρότερη τιμή.

SUM

Αυτή η συνάρτηση υπολογίζει το άθροισμα των τιμών μιας στήλης.

Αν θέλετε να υπολογίσετε το άθροισμα των τιμών της στήλης PRICE, τότε πληκτρολογήστε την εντολή:

```
SELECT SUM(PRICES) FROM SALES;
```

COUNT

Αυτή η συνάρτηση υπολογίζει τον αριθμό των γραμμών που ικανοποιούν τις συνθήκες.

Αν θέλετε να βρείτε πόσοι υπάλληλοι έχουν γεννηθεί πριν το 1970, τότε δώστε την εντολή:

```
SELECT COUNT(EMPLOYEE_ID) FROM EMPLOYEE WHERE  
BIRTH_DATE<'1.1.1970';
```

AVG

Η συνάρτηση AVG υπολογίζει τον μέσο όρο της στήλης.

Για να βρείτε τον μέσο όρο των τιμών, δώστε την εντολή:

```
SELECT AVG(PRICE) FROM SALES;
```

EXTRACT

Η συνάρτηση EXTRACT είναι πολύ σημαντική γιατί με την χρήση της μπορείτε να πάρετε ένα μέρος από DATE, TIME, TIMESTAMP.

Αν θέλετε να δείτε τις χρονιές που γεννήθηκαν οι υπάλληλοι, δώστε την εντολή:

```
SELECT EXTRACT(YEAR FROM BIRTH_DATE) FROM EMPLOYEE;
```

Ομοίως μπορείτε να αντί για year να βρείτε day, month και άλλα αναλόγως αν είναι DATE, TIME, TIMESTAMP.

SUBSTRING

Με αυτή την συνάρτηση μπορείτε να πάρετε μόνο ένα μέρος μιας συμβολοσειράς.

Για παράδειγμα αν ξέρετε ότι τα τρία πρώτα γράμματα του DEPARTMENT_ID συμβολίζουν κάτι και θέλετε να τα βρείτε, τότε δώστε την εντολή:

```
SELECT SUBSTRING(DEPARTMENT_ID FROM 1 FOR 3) FROM EMPLOYEE;
```

UPPER

Όπως θα γνωρίζετε ήδη αυτή η συνάρτηση μετατρέπει τα γράμματα μιας συμβολοσειράς σε κεφαλαία.

Για να δείτε το επώνυμο των υπαλλήλων με κεφαλαία γράμματα δώστε την εντολή:

```
SELECT UPPER(LAST_NAME) FROM EMPLOYEE;
```