



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΜΕΡΟΣ ΔΕΥΤΕΡΟ

Δημιουργία Βάσης, Πίνακα, Domain

Γιώργος Μαρκομανώλης

Περιεχόμενα

Δημιουργία Βάσης.....	1
Δημιουργία μιας shadow βάσης.....	2
Διαγραφή της shadow βάσης.....	2
Διαγραφή βάσης.....	2
Δημιουργία Πινάκων.....	3
Προκαθορισμένη τιμή (Default).....	4
Character set.....	5
Η εντολή Collate.....	5
Στήλες που υπολογίζονται από άλλες (Computed).....	5
Not Null.....	6
Primary Key.....	6
Unique Constraint.....	7
Αλλαγή ονόματος στήλης.....	8
Αλλαγή τύπου δεδομένων στήλης.....	8
Διαγραφή στήλης.....	8
Διαγραφή περιορισμού.....	9
Διαγραφή περιορισμών CHECK.....	9
Προσθήκη στήλης.....	9
Προσθήκη περιορισμού σε πίνακα.....	9
Διαγραφή πίνακα.....	9
Domain.....	10
Δημιουργία Domain.....	11
Τύπος Δεδομένων για Domain.....	11
Η ιδιότητα Default.....	12
Ιδιότητα NOT NULL.....	13
Εντολή CHECK.....	13
Αλλαγή του Domain.....	16
Διαγραφή Domain.....	17
Δημιουργία script (sql αρχείου).....	18
Δημιουργία βάσης από sql αρχείο.....	18
Δημιουργία backup.....	18
Επαναφορά από backup.....	18

Δημιουργία Βάσης

Αρχικά συνδεθείτε στην Firebird.

Για να συνδεθείτε στην Firebird πληκτρολογήστε την εντολή:

```
isql -u sysdba -p "ο κωδικός που έχετε" χωρίς τα "".
```

Η σύνταξη της εντολής για δημιουργία μιας βάσης είναι η παρακάτω:

```
CREATE DATABASE 'πλήρης_διαδρομή της βάσης'  
[USER 'όνομα_χρήστη' PASSWORD 'κωδικός_χρήστη']  
[DEFAULT CHARACTER SET charset];
```

Για Ελληνικά το charset είναι το ISO88597

Η εντολή δημιουργίας μιας βάσης έχει και άλλες συνθήκες αλλά ξεφεύγουν από τον σκοπό των εργαστηρίων.

Παρατηρήσεις: Αν δεν χρησιμοποιήσετε καθόλου όνομα χρήστη και κωδικό στην παραπάνω εντολή, τότε ο χρήστης με τον οποίο συνδεθήκατε στην βάση, θεωρείται ο δημιουργός της βάσης, ενώ αν χρησιμοποιήσετε όνομα χρήστη και κωδικό, τότε θα είναι αυτός ο που έχει πλήρη δικαιώματα στην βάση. Επίσης αν δεν χρησιμοποιήσετε charset τότε θα χρησιμοποιηθεί η Αγγλική γλώσσα.

Παράδειγμα

Για να δημιουργήσετε μια βάση στον φάκελο d:\databases\ με όνομα mathimatiko, πληκτρολογήστε την εντολή:

```
CREATE DATABASE 'd:\databases\mathimatiko.fdb';
```

Παρατήρηση: Ο δημιουργός της βάσης είναι ο χρήστης με τον οποίο έχετε συνδεθεί στην βάση. Ο χρήστης sysdba έχει πρόσβαση σε όλες τις βάσεις. Αν έχετε συνδεθεί στην Firebird σαν sysdba και δώστε την εντολή CREATE DATABASE 'd:\databases\mathimatiko.fdb' user 'όνομα_χρήστη' password 'κωδικός'; Τότε ο ιδιοκτήτης της βάσης θα είναι ο νέος χρήστης.

Διαδικασία shadowing

Αν συμβεί κάποιο πρόβλημα στον δίσκο ή στο δίκτυο (αν επικοινωνεί ο χρήστης με την βάση μέσω δικτύου) ή γίνει κάποιο άλλο λάθος του λειτουργικού, υπάρχει η δυνατότητα για επανάκτηση της βάσης με την διαδικασία shadowing που δεν είναι τίποτα άλλο παρά

μόνο μια διαδικασία με την οποία δημιουργείται ένα πιστό αντίγραφο της βάσης κατά την δημιουργία της και την αλλαγή των περιεχομένων της.

Δεν χρησιμεύει σαν backup της βάσης και αν την διαγράψετε δεν θα λειτουργεί πλέον η διαδικασία για αυτή την βάση. Δεν μπορείτε να συνδεθείτε σε μια βάση που έχει δημιουργηθεί από shadowing. Μια βάση shadowing πρέπει να δημιουργηθεί σε άλλον δίσκο από αυτόν που υπάρχει η αρχική βάση ώστε άμα χαλάσει ο δίσκος να μπορέσετε να επαναφέρετε την βάση.

Δημιουργία μιας shadow βάσης

- 1) Συνδεθείτε με την βάση που θέλετε να χρησιμοποιήσετε την διαδικασία shadowing.
- 2) Πληκτρολογήστε την εντολή:
CREATE SHADOW έναν_ακέραιο CONDITIONAL 'πλήρης διαδρομή της βάσης με επέκταση της βάσης shd';

Παράδειγμα

Για να δημιουργήσετε μια βάση στον φάκελο d:\databases\ της βάσης employee ακολουθήστε τα βήματα:

- 1) Συνδεθείτε στην βάση employee.
- 2) Πληκτρολογήστε την εντολή:
CREATE SHADOW 1 CONDITIONAL 'd:\databases\employee.shd';

Προσοχή: Το όνομα της shadow βάσης πρέπει να είναι ίδιο με το όνομα της αρχικής βάσης και η επέκταση του αρχείου είναι "shd".

Διαγραφή της shadow βάσης

Η εντολή για την διαγραφή της shadow βάσης είναι η ακόλουθη:

DROP SHADOW αριθμός;

Όπου αριθμός, είναι ο ακέραιος που χρησιμοποιήσατε όταν δημιουργήσατε την shadow βάση. Αν δεν θυμάστε τον αριθμό, συνδεθείτε με την βάση που θέλετε να διαγράψετε την shadow, δώστε την εντολή show database; και θα δείτε τον αριθμό της shadow.

Διαγραφή βάσης

Συνδεθείτε με την βάση που θέλετε να διαγράψετε και πληκτρολογήστε τις εντολές:
DROP DATABASE;

COMMIT;

Πίνακες

Δημιουργία Πινάκων

Αρχικά έχετε δημιουργήσει την βάση για την οποία θα φτιάξετε τους πίνακες. Επίσης έχετε συνδεθεί στην βάση και έχετε δημιουργήσει τα domain που θέλετε να χρησιμοποιήσετε.

Η γενική σύνταξη της εντολής για την δημιουργία πινάκων είναι:

CREATE TABLE όνομα_πίνακα (ορισμός γνωρισμάτων του πίνακα).

Πρέπει το όνομα του πίνακα να είναι μοναδικό σε κάθε βάση, δηλαδή δεν γίνεται να δημιουργήσετε σε μια βάση δύο πίνακες με το ίδιο όνομα.

Για τον ορισμό των γνωρισμάτων του πίνακα πρέπει να ορίσουμε:

- Τον όνομα του πεδίου που είναι μοναδικό στον πίνακα.
- Ένα από τα ακόλουθα
 - 1) Τύπος δεδομένων
 - 2) Μια έκφραση για μια στήλη που υπολογίζεται από άλλες
 - 3) Το όνομα ενός domain
- Προκαθορισμένη τιμή αν έχει
- Αν δέχεται το NULL ή όχι
- Αν υπάρχει κάποιος έλεγχος
- Αλλαγή της διάταξης των γραμμμάτων με collate

Παράδειγμα

Θα δημιουργήσουμε έναν πίνακα με όνομα student και θα έχει σαν γνωρίσματα: student_id (αριθμός μητρώου), First_name, Last_name, Full_name και Phone_number το κινητό τηλέφωνο των φοιτητών.

Πριν προχωρήσετε πρέπει να αποφασίσουμε το κάθε γνώρισμα τι τύπος δεδομένων είναι. Το `student_id` επειδή μπορεί να είναι μεγάλος αριθμός θα το δηλώσουμε σαν `BIGINT`. Το όνομα δεν θα είναι πάνω από 20 χαρακτήρες. Το επώνυμο θα είναι μέχρι 50 χαρακτήρες. Το `full_name` στηρίζεται στο `first_name` και `last_name`. Τέλος το `phone_number` θα είναι ένα domain που θα δηλώσουμε σε λίγο.

Ορίζουμε το Domain:

```
Create domain tel_number as varchar(10)
check (value like '(69%)%');
```

Βάλαμε την τιμή like '(69%)%' γιατί όλα τα νούμερα κινητών αρχίζουμε με 69.

Έπειτα δημιουργούμε τον πίνακα.

```
CREATE TABLE student (
  STUDENT_ID BIGINT NOT NULL,
  FIRST_NAME VARCHAR(20),
  LAST_NAME VARCHAR(50),
  FULL_NAME COMPUTED BY (FIRST_NAME ||' '|| LAST_NAME),
  PHONE_NUMBER TEL_NUMBER);
```

Παρατήρηση: Ο χρήστης που δημιουργεί τον πίνακα, είναι και αυτός που έχει δικαιοδοσία στον πίνακα. Επίσης θα μπορούσατε να προσθέσετε το check κατά την δημιουργία του πίνακα δίπλα από την στήλη που θέλετε να έχει το check.

Γνώρισμα που έχει οριστεί με domain

Αν ένα γνώρισμα έχει οριστεί με domain, μπορείτε να ορίσετε κατά την δημιουργία του πίνακα νέα προκαθορισμένη τιμή, ένα νέο έλεγχο και νέα διάταξη με το collate και να λαμβάνονται αυτές επιλογές και όχι του domain.

Προσοχή: Ένα domain που έχει οριστεί ώστε να μην παίρνει την κενή τιμή το γνώρισμα δεν γίνεται να ορίσετε τιμή NULL κατά την δημιουργία του πίνακα.

Προκαθορισμένη τιμή (Default)

Η προκαθορισμένη τιμή αποτρέπει την δημιουργία λαθών κατά την εισαγωγή δεδομένων. Αν δεν υπάρχει το όνομα της στήλης κατά την εισαγωγή δεδομένων, τότε σε αυτή την στήλη θα καταχωρηθεί η προκαθορισμένη τιμή.

Η προκαθορισμένη τιμή μπορεί να είναι:

- Μια σταθερά.
- Μεταβλητή όπως CURRENT_TIMESTAMP, CURRENT_USER και άλλα.
- Ημερομηνία όπως 'NOW', 'TOMORROW' και άλλα/
- NULL.

Character set

Το Character set το χρησιμοποιούμε για χαρακτήρες ή blob στήλες για το περιεχόμενο που θα αποθηκεύεται.

Παράδειγμα

Αν θέλετε να δημιουργήσετε έναν πίνακα με το όνομα description και να έχετε μια στήλη που να αποθηκεύεται Ελληνικό κείμενο, τότε δώστε την εντολή:

```
CREATE TABLE description (
  TITLE VARCHAR(100) CHARACTER SET ISO88597);
```

Η εντολή Collate

Χρησιμοποιείτε σε στήλες char ή varchar για να αλλάξετε την σειρά ταξινόμησης.

Παράδειγμα

Για να δημιουργήσετε ένα πίνακα όπως πιο πάνω αλλά με collation Ελληνικά στον παραπάνω πίνακα δώστε την εντολή:

```
CREATE TABLE description (
  TITLE VARCHAR(100) CHARACTER SET ISO88597 COLLATE ISO8859_7);
```

Στήλες που υπολογίζονται από άλλες (Computed)

Στήλες στις οποίες το περιεχόμενο είναι το σύνολο κάποιων άλλων στηλών είναι computed.

Η σύνταξη της εντολής είναι:

όνομα_στήλης COMPUTED BY (στήλες)

Όπως είδαμε παραπάνω για την δημιουργία του πίνακα χρησιμοποιήσαμε την εντολή:

```
CREATE TABLE student (
  STUDENT_ID BIGINT NOT NULL,
  FIRST_NAME VARCHAR(20),
  LAST_NAME VARCHAR(50),
  FULL_NAME COMPUTED BY (FIRST_NAME ||' '|| LAST_NAME),
  PHONE_NUMBER TEL_NUMBER);
```

Οπότε η στήλη FULL_NAME περιέχει το FIRST_NAME και το LAST_NAME

Παρατήρηση: Οι περιορισμοί που μπαίνουν σε computed στήλες αγνοούνται. Επίσης δεν μπορείτε να κάνετε index μια computed στήλη. (Θα μιλήσουμε αργότερα για το index).

Περιορισμοί

NOT NULL

Με αυτό τον περιορισμό πρέπει η στήλη να περιέχει οπωσδήποτε τιμές.

Πρέπει να χρησιμοποιείτε σε στήλες που είναι primary key ή unique. Δεν μπορείτε να αλλάξετε αυτό τον περιορισμό χωρίς να διαγράψετε την στήλη.

PRIMARY KEY

Μια στήλη ή το μικρότερο σύνολο στηλών που έχει τιμή διαφορετική για κάθε γραμμή είναι primary key. Δηλαδή αν ξέρουμε το primary key και μπορούμε να βρούμε μοναδική την γραμμή

Το primary key καλύτερα να είναι μια στήλη αν γίνεται. Το πως θα το βρίσκεται είναι θέμα λογικής.

Παράδειγμα

Αν θέλετε σε έναν πίνακα με το όνομα student, ο αριθμός μητρώου (student_id) να είναι primary key, τότε πληκτρολογήστε την εντολή:

```
CREATE TABLE student(
  STUDENT_ID BIGINT NOT NULL PRIMARY KEY);

COMMIT;
```


Προσοχή: Για να δηλώσετε μια στήλη σαν primary key, πρέπει να την δηλώσετε και σαν NOT NULL.

Αν θέλετε να δημιουργήσετε έναν δικό σας περιορισμό με όνομα PRI για το primary key, και όχι του συστήματος που δημιουργήθηκε πριν, τότε δώστε την εντολή:

```
CREATE TABLE student(
STUDENT_ID BIGINT NOT NULL,
CONSTRAINT PRI PRIMARY KEY(STUDENT_ID));
```

Προσθήκη primary key σε ήδη δημιουργημένο πίνακα

Έστω δημιουργούμε τον πίνακα:

```
CREATE TABLE example(
NUM INTEGER NOT NULL);
```

```
COMMIT;
```

Για να κάνετε την στήλη NUM primary key, δώστε την εντολή:

```
ALTER TABLE example
ADD CONSTRAINT PK
PRIMARY KEY(NUM);
```

UNIQUE CONSTRAINT

Όπως το Primary key έτσι και με το unique μπορείτε να ορίσετε ότι δεν θα υπάρχουν δύο σειρές που θα έχουν ίδιες τιμές σε μια στήλη ή σε ένα σύνολο στηλών. Ο unique περιορισμός δημιουργεί ένα unique key χρήσιμο για περιπτώσεις που θέλετε να ψάξετε στην βάση αλλά και για δημιουργία foreign keys που θα μιλήσουμε αργότερα. Στην πράξη ορίζετε unique και στην περίπτωση που έχετε δύο ή παραπάνω υποψήφια primary key και ορίζετε το ένα primary key και τα άλλα unique.

Παράδειγμα

Θα δημιουργήσετε ένα πίνακα που αποθηκεύεται ο αριθμός μητρώου του φοιτητή και το τηλέφωνό του. Ο αριθμός μητρώου του φοιτητή είναι primary key. Όμως αν θεωρήσουμε ότι δεν υπάρχουν φοιτητές με ίδιο νούμερο τηλεφώνου, τότε και αυτό θα μπορούσε να είναι κλειδί. Διαλέγουμε σαν κύριο κλειδί αυτό που πιστεύουμε ότι προσδιορίζει τον φοιτητή, τον αριθμό μητρώου γιατί τηλέφωνο μπορεί να αλλάξει, αριθμό μητρώου όμως όχι.

Δώστε την εντολή:

```
CREATE DATABASE students(
STUDENT_ID BIGINT NOT NULL PRIMARY KEY,
ADDRESS VARCHAR(40) NOT NULL UNIQUE);
```

Επεξεργασία πίνακα

Με την εντολή ALTER TABLE μπορείτε να ξεκινήσετε την διαδικασία για να αλλάξετε, να διαγράψετε και να προσθέσετε στήλες ή περιορισμούς σε έναν πίνακα.

Προσοχή: Όταν αλλάξετε κάτι σε μια στήλη προσέξτε μην υπάρχουν εξαρτήσεις από περιορισμούς, όψεις ή κάτι άλλο και δημιουργηθούν προβλήματα. Διαγράψτε πρώτα κάθε εξάρτηση.

Αλλαγή ονόματος στήλης

Για να αλλάξετε το όνομα STUDENT_ID σε STUDENTS_ID, πληκτρολογήστε την εντολή:

```
ALTER TABLE students
ALTER COLUMN STUDENT_ID TO STUDENTS_ID;
```

Αλλαγή τύπου δεδομένων στήλης

Για να αλλάξετε τύπο δεδομένων από BIGINT σε VARCHAR(10) δώστε την εντολή:

```
ALTER TABLE STUDENTS
ALTER COLUMN STUDENTS_ID TYPE VARCHAR(10);
```

Διαγραφή στήλης

Για να μπορέσετε να διαγράψετε στήλη πρέπει να μην είναι unique, primary key ή foreign key, να μην έχει έλεγχο (check) και να μην χρησιμοποιείτε σε view, trigger, stored procedure.

Για να διαγράψετε την στήλη ADDRESS από τον πίνακα STUDENTS δώστε την εντολή:

```
ALTER TABLE STUDENTS DROP ADDRESS;
```

Για να διαγράψετε πολλές στήλες μαζί, δώστε την εντολή:

```
ALTER TABLE NAME
```

```
DROP COLUMN1_NAME,  
DROP COLUMN2_NAME;
```

Διαγραφή περιορισμού

Για να διαγράψετε έναν περιορισμό με όνομα TER, δώστε την εντολή:

```
ALTER TABLE NAME  
DROP CONSTRAINT TER;
```

Διαγραφή περιορισμών CHECK

Η διαγραφή ενός check που ορίστηκε κατά την δημιουργία ενός πίνακα διαγράφεται με την εντολή:

```
ALTER TABLE NAME  
DROP CONSTRAINT NAME_OF_CHECK;
```

Για να διαγράψετε check από domain πρέπει να αλλάξετε τύπο δεδομένων ή domain.

Προσθήκη στήλης

Για να προσθέσετε δύο στήλες στον πίνακα students με όνομα f_name και l_name όπου καταχωρούν όνομα και επώνυμο, δώστε την εντολή:

```
ALTER TABLE STUDENTS  
ADD F_NAME VARCHAR(30) NOT NULL,  
ADD L_NAME VARCHAR(50) NOT NULL;
```

Για να προσθέσετε στήλη m_name με περιορισμό unique δώστε την εντολή:

```
ALTER TABLE STUDENTS  
ADD M_NAME VARCHAR(30) NOT NULL UNIQUE;
```

Προσθήκη περιορισμού σε πίνακα

```
ALTER TABLE STUDENTS  
ADD CONSTRAINT CON1 UNIQUE(L_NAME);
```

Διαγραφή πίνακα

Για να διαγράψετε τον πίνακα students, δώστε την εντολή
DROP TABLE STUDENTS;

Domains

Το Domain είναι ένας τύπος δεδομένων που καθορίζεται από τον χρήστη. Δεν είναι δυνατόν να φτιάξετε έναν νέο τύπο δεδομένων με τα εργαλεία που έχει η firebird, αλλά με ένα domain μπορείτε να δημιουργήσετε ένα πακέτο με ιδιότητες των τύπων δεδομένων που υπάρχουν ήδη, να του δώσετε ένα όνομα και να το χρησιμοποιείτε σαν τύπο δεδομένων για την δήλωση στηλών σε πίνακες.

Ένα Domain είναι διαθέσιμο για κάθε πίνακα που υπάρχει στην βάση. Όλες οι στήλες, σε κάθε πίνακα που είναι ορισμένες με το ίδιο Domain θα έχουν ολόιδιες ιδιότητες, οι οποίες κληρονομούν τις ιδιότητες του Domain οι οποίες μπορεί να είναι:

- Τύπος δεδομένων (είναι υποχρεωτικό να υπάρχει σε κάθε Domain)
- Default value για εισαγωγή δεδομένων. Δηλαδή όταν εισάγετε δεδομένα στον πίνακα, οι στήλες που είναι ορισμένες με το Domain να έχουν μια αρχική τιμή.
- Αν γίνεται αυτή η στήλη να παίρνει NULL τιμές ή όχι.
- CHECK περιορισμοί.
- Character set (χρησιμοποιείτε μόνο σε τύπους δεδομένων συμβολοσειρών και blob)
- Collation order (για συμβολοσειρές μόνο).

Επεξήγηση για το Character set και Collation order:

Το Character_set ρυθμίζει τους χαρακτήρες που μπορούν να χρησιμοποιηθούν για την αποθήκευση κειμένου σε στήλες με τύπο δεδομένων CHAR, VARCHAR και BLOB SUB_TYPE 1, καθώς και τα byte που χρειάζεται κάθε χαρακτήρας.

Το Collation order ρυθμίζει την διάταξη των γραμμμάτων για σύγκριση συμβολοσειρών.

Σημείωση: Δεν μπορείτε να χρησιμοποιήσετε ξένα κλειδιά σε στήλες που έχουν οριστεί με Domain. Τα ξένα κλειδιά θα τα δούμε αργότερα.

Παράδειγμα

Έστω ότι σε ένα πίνακα αποθηκεύουμε το επώνυμο των υπαλλήλων που δουλεύουν σε μια εταιρεία και θέλουμε να είναι όλα τα γράμματα με κεφαλαία και έστω ότι το επώνυμο θα έχει μήκος το πολύ 50 χαρακτήρες και φυσικά κάθε υπάλληλος θα έχει επώνυμο, δεν γίνεται να υπάρχει κενό πεδίο. Τότε θα γράψουμε:

```
CREATE DOMAIN SURNAME AS CHAR(50) NOT NULL
CHECK (VALUE=UPPER(VALUE));
```

Παρατήρηση: Μετά από εντολές σχετικά με δημιουργία και εισαγωγή δεδομένων θα πληκτρολογείτε την εντολή COMMIT;

Με τον παραπάνω τρόπο δημιουργούμε Domain με το όνομα SURNAME, έχει τύπο δεδομένων CHAR μέχρι 50 χαρακτήρες και δεν είναι ποτέ κενό (NULL) . Επίσης κάνει έλεγχο αν το δεδομένο που θέλετε να εισάγετε (VALUE), έχει όλα τα γράμματα κεφαλαία (UPPER).

Δημιουργία Domain

Η σύνταξη για την δημιουργία Domain είναι η ακόλουθη:

```
CREATE DOMAIN όνομα_domain [AS] <τύπος δεδομένων>
[DEFAULT {literal | NULL | USER}]
[NOT NULL] [CHECK (<συνθήκες>)]
[CHARSET charset | NONE]
[COLLATE collation];
```

Ό,τι υπάρχει μέσα σε [] είναι προαιρετικό.

Πρέπει το όνομα_domain να είναι μοναδικό σε κάθε βάση.

Τύπος Δεδομένων για Domain

Ο τύπος δεδομένων είναι υποχρεωτικό να υπάρχει και μπορεί να είναι όποιος θέλετε αρκεί να τον υποστηρίζει η Firebird. Δεν είναι δυνατόν να βάλετε σαν τύπο δεδομένων, ένα άλλο domain.

Παράδειγμα

Με την εντολή:

```
CREATE DOMAIN DESCRIPTION AS BLOB SUB_TYPE TEXT
CHARACTER SET ISO-8859-7;
```

Δημιουργείται ένα blob domain με όνομα DESCRIPTION, με το οποίο μπορείτε σε μια στήλη να αποθηκεύσετε Ελληνικό κείμενο. Το προκαθορισμένο CHARACTER SET είναι τα αγγλικά αλλά με το domain το αλλάζετε για κάθε στήλη που θα δηλωθεί με το domain.

Η ιδιότητα Default

Μπορείτε να ορίσετε μια default τιμή την οποία θα χρησιμοποιεί η Firebird για την εισαγωγή δεδομένων, όταν δεν έχετε κάποια τιμή στην εντολή insert για την συγκεκριμένη στήλη.

Οι προκαθορισμένες τιμές μπορεί να είναι:

- Ένας σταθερός. Μπορείτε να ορίσετε μια συμβολοσειρά, ένα νούμερο ή μια ημερομηνία ακόμα και το μηδέν για στήλες που δεν παίρνουν τιμή NULL.
- CURRENT_TIMESTAMP (την ημερομηνία και την ώρα την στιγμή που γίνεται η εισαγωγή δεδομένων), CURRENT_DATE (την ημερομηνία την στιγμή που γίνεται η εισαγωγή δεδομένων), CURRENT_TIME (την ώρα την στιγμή που γίνεται εισαγωγή των δεδομένων) ή ημερομηνία.
- USER (όνομα χρήστη), CURRENT_USER (ο χρήστης που είναι συνδεδεμένος) ή CURRENT_ROLE (ο role που χρησιμοποιείτε εκείνη την στιγμή).
- CURRENT_CONNECTION (η σύνδεση με την οποία γίνεται η σύνδεση με την βάση κατά την διάρκεια εισαγωγής δεδομένων) ή CURRENT_TRANSACTION (η συναλλαγή με την οποία γίνεται η εισαγωγή δεδομένων).

Μπορείτε να δηλώσετε και ένα domain σαν NULL, αλλά αυτό γίνεται αυτόματα όταν δεν είναι δηλωμένο σαν NOT NULL.

Παράδειγμα

Θα δημιουργήσουμε ένα domain με το όνομα payment, η τιμή του μισθού πρέπει να είναι το πολύ 2000 ευρώ και η προκαθορισμένη τιμή μισθού είναι 1000 ευρώ. Άρα θα έχουμε:

```
CREATE DOMAIN payment AS INTEGER
DEFAULT 1000
CHECK (VALUE<2000);
```

Προσοχή: Η default τιμή θα δουλέψει κατά την εισαγωγή νέας γραμμής σε έναν πίνακα και μόνο αν η εντολή insert δεν περιέχει την στήλη που είναι δηλωμένη με το domain με την default τιμή.

Ιδιότητα NOT NULL

Με αυτή την ιδιότητα πρέπει όλα τα πεδία της στήλης να έχουν κάποια τιμή και να μην είναι κενά.

Εντολή CHECK

Η εντολή CHECK συντάσσεται ως CHECK(συνθήκη) και έχει σκοπό να περιορίζει τα δεδομένα που θέλουμε να αποθηκεύουμε στον πίνακα, όπου συνθήκη μπορεί να είναι μία ή περισσότερες από τις παρακάτω:

Συνθήκη=

1) VALUE (αριθμητικός τελεστής) <val>

Δηλαδή αν θέλουμε η τιμή που θα καταχωρείτε στην βάση να είναι πάνω από 1000, τότε έχουμε:

VALUE>1000

Ο αριθμητικός τελεστής μπορεί να είναι: =,<,>,<=,>=,!<,!>,<>,! =

Το !< σημαίνει όχι μικρότερο από

Το !> σημαίνει όχι μεγαλύτερο από

Το !=, <> σημαίνει όχι ίσο με

2) VALUE [NOT] BETWEEN <val> AND <val>

Αυτή η συνθήκη είναι για να ελέγχουμε αν τα δεδομένα εισαγωγής είναι μεταξύ δύο τιμών. Αν θέλετε να αποθηκεύονται μόνο οι τιμές που είναι μεταξύ του 1000 και του 5000, τότε έχουμε:

VALUE BETWEEN 1000 AND 5000

Αν θέλετε τις τιμές που είναι έξω από το παραπάνω διάστημα, τότε έχουμε:

VALUE NOT BETWEEN 1000 AND 5000

3) VALUE [NOT] LIKE <val>

Αυτή η συνθήκη ελέγχει αν η VALUE είναι ίση με μια συμβολοσειρά, της οποίας ορίζουμε ένα μέρος της. Αν θέλουμε να καταχωρείται το νούμερο ενός κινητού τηλεφώνου το οποίο ξέρουμε ότι θα ξεκινάει με τα ψηφία 69, τότε θα έχουμε:

VALUE LIKE '69%'

Όταν θέλουμε να ψάξουμε για μια συμβολοσειρά της οποίας ξέρουμε ένα μέρος της, τότε γράφουμε:

VALUE LIKE '%name%'

Με την παραπάνω εντολή επιτρέπεται τα δεδομένα να είναι της μορφής: μια_λέξη_name_μια_λέξη ή μια_λέξη_name, ή name_μια_λέξη ή name.

Αντίστοιχα με την εντολή VALUE NOT LIKE '%name%' θέλουμε τα δεδομένα να μην είναι αυτά που δέχεται η εντολή χωρίς το NOT.

Με το σύμβολο % έχουμε ή κανέναν χαρακτήρα ή έναν ή πολλούς. Με το σύμβολο _ έχουμε έναν χαρακτήρα.

4) VALUE [NOT] IN (<val> [,<val>...])

Αυτή η συνθήκη ελέγχει αν το δεδομένο εισαγωγής είναι μια από τις τιμές <val>. Δηλαδή αν ξέρουμε τις πιθανές τιμές μιας στήλης μπορούμε να χρησιμοποιήσουμε αυτή την συνθήκη. Για παράδειγμα αν ξέρουμε ότι σε μια στήλη τις πιθανές απαντήσεις, έστω ότι είναι 4 απαντήσεις, τότε θα έχουμε:

VALUE IN ('A1', 'A2', 'A3', 'A4')

Οπότε θα γίνεται έλεγχος αν το δεδομένο εισαγωγής είναι μια από αυτές τις λέξεις.

Με το NOT γίνεται έλεγχος αν τα δεδομένα εισαγωγής δεν είναι κάποιο από τις τιμές στην συνθήκη.

5) VALUE IS [NOT] NULL

Με την συνθήκη VALUE IS NULL επιτρέπουμε στην Furebird να μην καταχωρεί καμιά τιμή. Με το NOT πρέπει οπωσδήποτε να υπάρχει τιμή.

6) VALUE [NOT] CONTAINING <val>

Με την συνθήκη VALUE CONTAINING <val> επιτρέπουμε την εισαγωγή δεδομένων που περιέχουν την τιμή val. Για να καταχωρούμε τα δεδομένα που έχουν την λέξη ma, δίνουμε την συνθήκη:

VALUE CONTAINING 'ma'

Με το NOT πρέπει τα δεδομένα να μην περιέχουν αυτή την λέξη.

7) VALUE [NOT] STARTING WITH <val>

Με την συνθήκη VALUE STARTING WITH <val> επιτρέπουμε την εισαγωγή δεδομένων που ξεκινάνε με την τιμή val. Αν θέλουμε να καταχωρούμε νούμερο κινητού, τότε μπορούμε να δώσουμε την συνθήκη:

VALUE STARTING WITH '69'

Με το NOT πρέπει τα δεδομένα να μην ξεκινάνε με την τιμή val.

8) Μια συνθήκη μπορεί να είναι ίση με μια άλλη συνθήκη ή όχι ίση.

9) Μια συνθήκη μπορεί να είναι ίση με μία συνθήκη από πολλές, δηλαδή

Συνθήκη = Συνθήκη OR Συνθήκη

10) Μια συνθήκη μπορεί να είναι η τομή δύο ή περισσότερων συνθηκών, δηλαδή

Συνθήκη = Συνθήκη AND Συνθήκη

Με την εντολή CHECK γίνεται έλεγχος αν μπορεί να καταχωρηθεί κάποια τιμή που θέλετε να εισάγετε και αν δεν ικανοποιεί τον έλεγχο, τότε δημιουργείτε μια εξαίρεση. Αν επιτρέπετε σε ένα πεδίο τιμών ενός πεδίου να παίρνει τιμή NULL, πρέπει να δηλώνεται στον έλεγχο.

Ένα Domain μπορεί να έχει μόνο έναν έλεγχο, όμως ο έλεγχος μπορεί να έχει πολλές συνθήκες. Καλό είναι όταν έχετε πολλαπλές συνθήκες να τις βάζετε παρενθέσεις για να μην γίνουν λάθη.

Αν θέλετε να φτιάξετε ένα domain με το όνομα test και ότι το πεδίο τιμών ενός πεδίου δεν θα είναι μηδέν και θα ξεκινάει από το γράμμα τότε δίνεται την εντολή:

```
create domain test as varchar(10)
check ((value is not null) and (value starting with 'G'));
```

Η Firebird δεν έχει τύπο δεδομένων Boolean, δηλαδή τύπο δεδομένων για την αποθήκευση μιας τιμή για true or false. Αυτό μπορεί να λυθεί με την δημιουργία ενός Domain. Αν θέλετε να αποθηκεύετε τιμές T,F ή Y,N θα χρησιμοποιείτε CHAR τύπο δεδομένων, αλλιώς αν θέλετε να χρησιμοποιείτε 1,0 θα χρησιμοποιείτε SMALLINT.

Boolean domain με True, False και default False

```
CREATE DOMAIN T_BOOLEAN AS CHAR
DEAFULT 'F' NOT NULL
CHECK(VALUE IN ('T','F'));
```

Παρατήρηση: η συνθήκη NOT NULL μπορεί να μπει και πριν το check.

Boolean domain με 1,0 και NULL

```
CREATE DOMAIN L_BOOLEAN AS SMALLINT
CHECK ((VALUE IS NULL) OR (VALUE IN (1,0)));
```

Μην χρησιμοποιείτε σαν ονόματα domain λέξεις όπως Boolean, true, false γιατί είναι καταχωρημένες από την Firebird.

Αλλαγή του Domain

Σε ένα domain μπορείτε να αλλάξετε:

- Το όνομα
- Τον τύπο δεδομένων
- Να διαγράψετε μια προκαθορισμένη τιμή
- Να θέσετε νέα προκαθορισμένη τιμή
- Να διαγράψετε έναν έλεγχο
- Να δημιουργήσετε έναν έλεγχο

Δεν μπορείτε να αλλάξετε την συνθήκη NOT NULL παρά μόνο αν διαγράψετε το domain και δημιουργήσετε νέο.

Η σύνταξη για αλλαγή του domain είναι η εξής:

```
ALTER DOMAIN {όνομα_υπάρχοντος_domain | όνομα_υπάρχοντος_domain TO
νέο_όνομα_domain} {
[SET DEFAULT {σταθερά | NULL | USER }]
|[DROP DEFAULT]
|[ADD [CONSTRAINT] CHECK (συνθήκη)]
|[DROP CONSTRAINT]
|TYPE τύπος_δεδομένων
```

```
};
```

Παραδείγματα

Για να αλλάξετε το όνομα ενός domain που έχει το όνομα month σε year, πληκτρολογήστε την εντολή:

```
ALTER DOMAIN month TO year;
```

Για να δώσετε νέα default τιμή την 20 σε ένα domain που λέγετε week, πληκτρολογήστε την εντολή:

```
ALTER DOMAIN week SET DEFAULT 20;
```

Για να αλλάξετε τον τύπο δεδομένων του domain week σε integer, πληκτρολογήστε την εντολή:

```
ALTER DOMAIN week TYPE INTEGER;
```

Προσοχή: Η αλλαγή του τύπου δεδομένων σε ένα domain δεν θα επιτραπεί αν πρόκειται να γίνει απώλεια δεδομένων.

Για να αλλάξετε τον περιορισμό στο domain week και να βάλετε έναν νέο ο οποίος κάνει δεχτές τις τιμές άνω του 4, τότε πληκτρολογήστε την εντολή:

```
ALTER DOMAIN week add check (value>4);
```

Προσοχή: Αν το domain περιέχει ήδη έναν έλεγχο, πρέπει να τον διαγράψετε πριν βάλετε νέο έλεγχο. Επομένως στην περίπτωση που υπάρχει έλεγχος, για να τον διαγράψετε πληκτρολογήστε την εντολή:

```
ALTER DOMAIN week DROP CONSTRAINT;
```

Διαγραφή Domain

Για να διαγράψετε το domain week πληκτρολογήστε την εντολή:

```
DROP DOMAIN week;
```

Προσοχή: Προσέξτε μην διαγράψετε κάποιο domain που χρησιμοποιείτε από πίνακα γιατί θα δημιουργηθούν προβλήματα.

Scripts - Backup

Δημιουργία script (sql αρχείου)

Για να δημιουργήσετε ένα sql αρχείο με όνομα 'name' στον φάκελο firebird_1_5, για την βάση employee, δώστε την παρακάτω εντολή:

```
isql -u sysdba -p ο_κωδικός_σας -extract -output c:\firebird_1_5\name.sql
c:\firebird_1_5\examples\employee.fdb
```

Δημιουργία βάσης από sql αρχείο

Για να τρέξετε το αρχείο name.sql, πληκτρολογήστε την εντολή:

```
INPUT c:\firebird_1_5\name.sql;
```

Την παραπάνω εντολή την πληκτρολογείτε αφού έχετε μπει στην γραμμή εντολών SQL.

Δημιουργία backup

Για να κάνετε backup την βάση employee που είναι στον φάκελο c:\firebird_1_5\examples\ και να δημιουργήσετε ένα αρχείο employee στον φάκελο d:\databases\ που θα είναι το backup πληκτρολογήστε την εντολή:

```
gbak -b -user sysdba -pas masterkey c:\firebird_1_5\examples\employee.fdb
d:\databases\employee.fbk
```

Όπου masterkey βάλτε τον κωδικό σας.

Την παραπάνω εντολή θα την πληκτρολογήσετε όταν είστε στον φάκελο c:\firebird_1_5\bin\

Επαναφορά από backup

Για να επαναφέρετε την βάση από ένα backup αρχείο, πληκτρολογήστε την εντολή:

```
gbak -c -user sysdba -pas masterkey d:\databases\employee.fbk
c:\firebird_1_5\examples\employee.fdb
```

Όπου masterkey βάλτε τον κωδικό σας.

Αν θέλετε η επαναφορά της βάσης να διαγράψει μια βάση που ήδη υπάρχει, τότε στην παραπάνω εντολή αντί για gbak -c πληκτρολογήστε gbak -r .