

Ειδικότητα:
Τεχνικός
Εφαρμογών
Πληροφορικής

Γ' Εξάμηνο
Μάθημα:
Βάσεις Δεδομένων Ι
(2Ε)

3^η-4^η Ενότητα:
Περιορισμοί
Ακεραιότητας &
Αναφοράς – Κλειδιά

Περιορισμοί Ακεραιότητας

- Μια ΒΔ είναι χρήσιμη μόνο όσο καταχωρεί χρήσιμη και σωστή πληροφορία. Λογική απαίτηση αποτελεί το ΣΔΒΔ να αποτρέπει την καταχώρηση λανθασμένης πληροφορίας.
- **Περιορισμός ακεραιότητας (IC-Integrity Constraint)** ονομάζεται η συνθήκη η οποία δηλώνεται να ισχύει επάνω σε ένα σχήμα ΒΔ και η οποία επιβάλλει περιορισμούς επί των δεδομένων τα οποία επιτρέπεται να καταχωρεί ένα (κάθε) στιγμιότυπο της ΒΔ.

Περιορισμοί Ακεραιότητας

- Ένα στιγμιότυπο ΒΔ ονομάζεται **έγκυρο** όταν τα δεδομένα τα οποία καταχωρεί, ικανοποιούν το σύνολο των περιορισμών ακεραιότητας που έχουν δηλωθεί να ισχύουν επάνω στο αντίστοιχο σχήμα ΒΔ.
- Το ΣΔΒΔ λειτουργεί έτσι ώστε να **επιβάλλει** διαρκώς την ισχύ των περιορισμών ακεραιότητας, επιτρέποντας να καταχωρούνται μόνο τα έγκυρα στιγμιότυπα της ΒΔ.

Περιορισμοί Ακεραιότητας

- Η δήλωση και ενεργοποίηση των IC γίνονται σε διαφορετικές χρονικές στιγμές:
 - τη στιγμή που ο τελικός χρήστης ή ο διαχειριστής της ΒΔ δηλώνει το σχήμα ΒΔ, δηλώνει και τους IC
 - καθώς λειτουργεί η εφαρμογή, το ΣΔΒΔ ελέγχει αυτόματα για τυχόν παραβιάσεις των IC, μη επιτρέποντας να συμβαίνουν μεταβολές που τους παραβιάζουν.

Περιορισμοί Ακεραιότητας

- Οι IC αποτελούν ένα μηχανισμό για τον έλεγχο της **συνέπειας** των δεδομένων.
- Χρησιμοποιούνται προκειμένου να εξασφαλιστεί ότι μια ΒΔ δε θα βρεθεί ποτέ σε ασυνεπή κατάσταση.
- Κατηγορίες:
 1. Περιορισμοί πεδίου ορισμού
 2. Περιορισμοί πρωτεύοντος κλειδιού
 3. Περιορισμοί αναφοράς / ξένου κλειδιού

1. Περιορισμοί Πεδίου Ορισμού

Προσθήκη περιορισμού κατά τη δημιουργία πίνακα

Παραδείγματα:

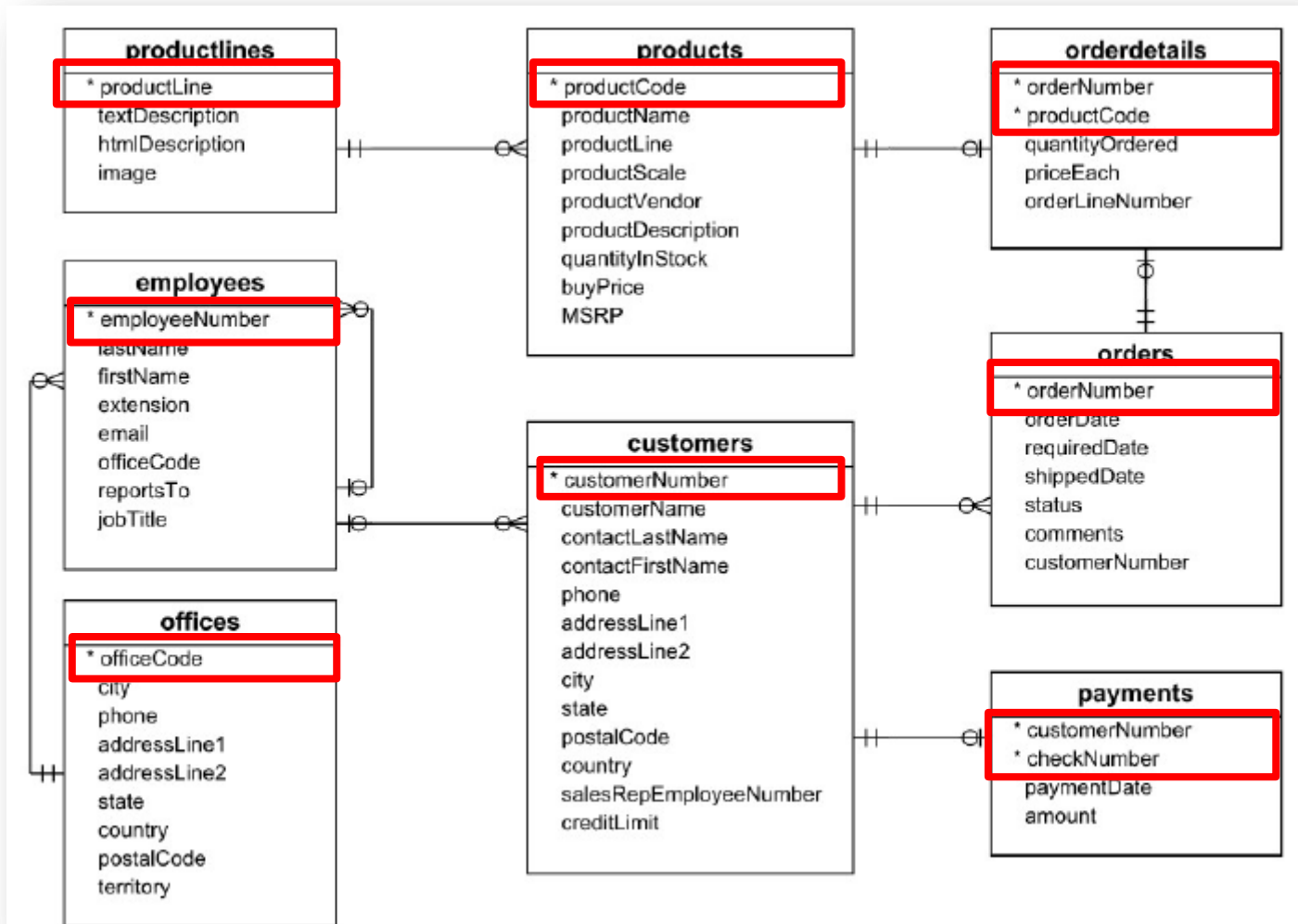
```
CREATE TABLE CUSTOMER (  
    customer_id    INT                CHECK (customer_id>0),  
    afm             CHAR(10)          DEFAULT '0000000000',  
    name           CHAR(20)          NOT NULL,  
    surname        CHAR(20)          NOT NULL,  
    account_type   CHAR(20)          CHECK(account_type in  
    ('Checking', 'Savings')),  
    birth_date     DATE  
)
```

- NOT NULL: το πεδίο δεν μπορεί να παραμείνει κενό κατά την εισαγωγή δεδομένων
- DEFAULT τιμή: δίνεται εξ αρχής μια προκαθορισμένη τιμή στο πεδίο, σε περίπτωση που δεν εισαχθεί κάποια τιμή στο πεδίο κατά την εισαγωγή δεδομένων

2. Περιορισμοί Πρωτεύοντος Κλειδιού

- Υπερκλειδί (Superkey): σύνολο από ένα ή περισσότερα γνωρίσματα (πεδία) που προσδιορίζουν μοναδικά μια οντότητα (πίνακα).
- Υποψήφιο κλειδί (Candidate key): το ελάχιστο (με το μικρότερο αριθμό γνωρισμάτων) υπερκλειδί.
- Πρωτεύον κλειδί (Primary key): το υποψήφιο κλειδί που επιλέγουμε για μια οντότητα.
- Ξένο κλειδί (Foreign key): σύνολο από ένα ή περισσότερα πεδία ενός πίνακα που μας επιτρέπουν να «συνδεθούμε» με μια γραμμή ενός άλλου πίνακα. Πρόκειται για κλειδί πίνακα που είναι πρωτεύον κλειδί στον πίνακα με τον οποίο «συνδέεται».

2. Περιορισμοί Πρωτεύοντος Κλειδιού



2. Περιορισμοί Πρωτεύοντος Κλειδιού

Ορισμός πρωτεύοντος κλειδιού κατά τη δημιουργία πίνακα

Παράδειγμα:

```
CREATE TABLE CUSTOMER (  
    customer_id    INT,  
    afm            CHAR(10),  
    name           CHAR(20),  
    PRIMARY KEY (customer_id)  
) ή CONSTRAINT cust_prim PRIMARY KEY (customer_id)
```

Ορισμός πρωτεύοντος κλειδιού μετά τη δημιουργία πίνακα

Παράδειγμα:

```
ALTER TABLE CUSTOMER MODIFY customer_id INT PRIMARY KEY
```

Διαγραφή πρωτεύοντος κλειδιού

Παράδειγμα:

```
ALTER TABLE CUSTOMER DROP PRIMARY KEY
```

2. Περιορισμοί Πρωτεύοντος Κλειδιού

Ορισμός υποψήφιου κλειδιού κατά τη δημιουργία πίνακα

Παράδειγμα:

```
CREATE TABLE CUSTOMER (  
    customer_id    INT,  
    afm            CHAR(10),  
    name          CHAR(20),  
    UNIQUE (customer_id)  
) ή CONSTRAINT cust_uniq UNIQUE (customer_id)
```

Ορισμός υποψήφιου κλειδιού μετά τη δημιουργία πίνακα

Παράδειγμα:

```
ALTER TABLE CUSTOMER ADD CONSTRAINT cust_uniq  
UNIQUE(customer_id)
```

Διαγραφή υποψήφιου κλειδιού

Παράδειγμα:

```
ALTER TABLE CUSTOMER DROP INDEX cust_uniq
```

3. Περιορισμοί Αναφοράς / Ξένου Κλειδιού

Ορισμός ξένου κλειδιού κατά τη δημιουργία πίνακα

Παράδειγμα:

```
CREATE TABLE ORDERR (  
    order_id          INT,  
    order_date       DATE,  
    customer_id      INT,  
    PRIMARY KEY (order_id),  
    FOREIGN KEY (customer_id) REFERENCES CUSTOMER (customer_id)  
) ή CONSTRAINT ord_for FOREIGN KEY (customer_id) REFERENCES  
    CUSTOMER (customer_id)
```

Ορισμός ξένου κλειδιού μετά τη δημιουργία πίνακα

Παράδειγμα:

```
ALTER TABLE ORDERR ADD FOREIGN KEY (customer_id) REFERENCES  
CUSTOMER (customer_id)
```

Διαγραφή ξένου κλειδιού

Παράδειγμα:

```
ALTER TABLE ORDERR DROP FOREIGN KEY ord_for
```

3. Περιορισμοί Αναφοράς / Ξένου Κλειδιού

- Για να καθορίσουμε τι επιθυμούμε να συμβεί σε περίπτωση αλλαγής ή διαγραφής μιας εγγραφής χρησιμοποιούμε στη δήλωση ξένου κλειδιού τις δηλώσεις **ON UPDATE** και **ON DELETE**. Η σύνταξη της δήλωσης ξένου κλειδιού είναι τώρα η εξής:

FOREIGN KEY (πεδία) **REFERENCES** όνομα_πίνακα (πεδία αναφοράς)

ON DELETE ενέργεια

ON UPDATE ενέργεια

3. Περιορισμοί Αναφοράς / Ξένου Κλειδιού

- Ενέργειες κατά τη διαγραφή (**ON DELETE**):
 1. **CASCADE**: σε περίπτωση διαγραφής του κλειδιού στον πίνακα-γονέα, διαγράφονται επίσης οι εγγραφές στον πίνακα-παιδί.
 2. **RESTRICT**: επιτρέπεται η διαγραφή με βάση τιμές της στήλης του πίνακα-γονέα μόνο αν δεν υπάρχουν αντίστοιχες τιμές στον πίνακα-παιδί.
 3. **SET NULL**: σε περίπτωση διαγραφής του κλειδιού στον πίνακα-γονέα, οι τιμές στον πίνακα-παιδί που αναφέρονται σ' αυτόν παίρνουν τιμή NULL.
 4. **SET DEFAULT**: σε περίπτωση διαγραφής του κλειδιού στον πίνακα-γονέα, οι αντίστοιχες τιμές στον πίνακα-παιδί παίρνουν την default τιμή του. Πρέπει να έχει ορισθεί όμως μια default τιμή, αλλιώς γίνεται NULL.

3. Περιορισμοί Αναφοράς / Ξένου Κλειδιού

- Ενέργειες κατά την ενημέρωση (**ON UPDATE**):
 1. **CASCADE**: σε περίπτωση αλλαγής του κλειδιού στον πίνακα-γονέα, αλλάζει επίσης και η αντίστοιχη αναφορά του στον πίνακα-παιδί.
 2. **RESTRICT**: σε περίπτωση αλλαγής του κλειδιού στον πίνακα-γονέα, η αλλαγή επιτρέπεται μόνο αν δεν υπάρχουν αναφορές του στον πίνακα-παιδιού.
 3. **SET NULL**: όταν αλλάζει η τιμή στον πίνακα-γονέα, η αντίστοιχη αναφορά στον πίνακα-παιδί γίνεται NULL.
 4. **SET DEFAULT**: όταν αλλάζει η τιμή στον πίνακα-γονέα, η αντίστοιχη αναφορά του στον πίνακα-παιδί παίρνει την default τιμή. Πρέπει να έχει ορισθεί όμως μία default τιμή, αλλιώς γίνεται NULL.