



Εθνικό Μετσόβιο Πολυτεχνείο

Αναφορά Εξαμηνιαίου Project

Βάσεις Δεδομένων

Σχεδιασμοί Βάσεων Δεδομένων

Βασίλης Γερακάρης (08092)
Διονύσης Ζήνδρος (06601)
Γρηγόρης Λύρας (09687)

Διδάσκοντες:
Γιάννης Βασιλείου
Τίμος Σελλής

18 Μαρτίου 2012

1 Λεπτομέρειες Υλοποίησης

Το project κατασκευάστηκε ως μια εφαρμογή ιστού, με έμφαση στην χρηστικότητα των ιστοσελίδων και την εύκολη πρόσβαση στα δεδομένα.

Για την υλοποίηση του χρησιμοποιήθηκε το *LAMP stack* δηλαδή:

- GNU-Linux (διανομή Debian squeeze ως λειτουργικού συστήματος)
- Apache (Web Server)
- MySQL (DataBase Management System)
- PHP (Scripting Language).

Ως database (storage) engine επιλέχθηκε η InnoDB. Επιπλέον, έγινε χρήση CSS για επεξεργασία της μορφοποίησης των ιστοσελίδων, και JavaScript.

Όλο το προαναφερθέν λογισμικό είναι ανοιχτού κώδικα (open source).

Κατά τη φάση σχεδίασης, αποφασίστηκε η εφαρμογή της αρχιτεκτονικής MVC (Model/View/Controller), η οποία ορίζει σαφή διαχωρισμό μεταξύ των διαφόρων συνιστωσών της εφαρμογής.

Έτσι το project αναλύθηκε σε models, όπου εμπεριέχεται όλη η λογική επικοινωνίας με τη βάση, views, όπου γίνεται η παραγωγή της διεπαφής με το χρήστη, και controllers, όπου αποφασίζεται ποιο view και model θα κληθεί, και υλοποιεί τη σωστή επικοινωνία μεταξύ τους.

Με τον τρόπο αυτό, καθίσταται ευκολότερη και ταχύτερη η ανάπτυξη και η αποσφαλμάτωση του λογισμικού, ενώ ο κώδικας είναι πιο ευέλικτος, αφού αποκρύπτονται οι λεπτομέρειες υλοποίησης μεταξύ των συνιστωσών (αγνωστικό μοντέλο - αποσύνδεση).

1.1 Πλεονεκτήματα

Η επιλογή του παραπάνω τρόπου υλοποίησης επιφέρει αρκετά πλεονεκτήματα, τα σημαντικότερα των οποίων είναι τα παρακάτω:

- Η κατασκευή σε μορφή ιστοσελίδων επιτρέπει πρόσβαση στη βάση από οποιονδήποτε υπολογιστή, χωρίς να εμφανίζονται διαφορές από όσους χρησιμοποιούν διαφορετικά λειτουργικά συστήματα.
- Η διανομή Debian squeeze (stable) διακρίνεται για τη σταθερότητα και την ασφάλειά που προσφέρει, επομένως είναι ιδανική για μια εφαρμογή web server.
- Εύκολη σύνταξη κώδικα, λόγω της ομαλής καμπύλης εκμάθησης της PHP και της MySQL.
- Εύκολη ανάπτυξη λογισμικού, αφού τα αρχεία της PHP αποθηκεύονται και επεξεργάζονται απευθείας σε remote Web server, κάτι που μας επέτρεψε απομακρυσμένη και ταυτόχρονη πρόσβαση και εργασία.
- Εύκολη κατασκευή της διεπαφής με το χρήστη με χρήση ιστοσελίδων.
- Αξιόπιστες συναλλαγές με τη βάση, αφού το InnoDB ικανοποιεί τις ιδιότητες ACID (Atomicity, Consistency, Isolation, Durability).

- Γρήγορη (σχεδόν άμεση) ανάκαμψη από crashes ή μη-αναμενόμενα shutdowns, αφού το InnoDB απλώς επαναλαμβάνει της ενέργειες στα log files της, αντίθετα με το MyISAM, που ελέγχει και επισκευάζει όλους τους δείκτες και πίνακες.

1.2 Μειονεκτήματα

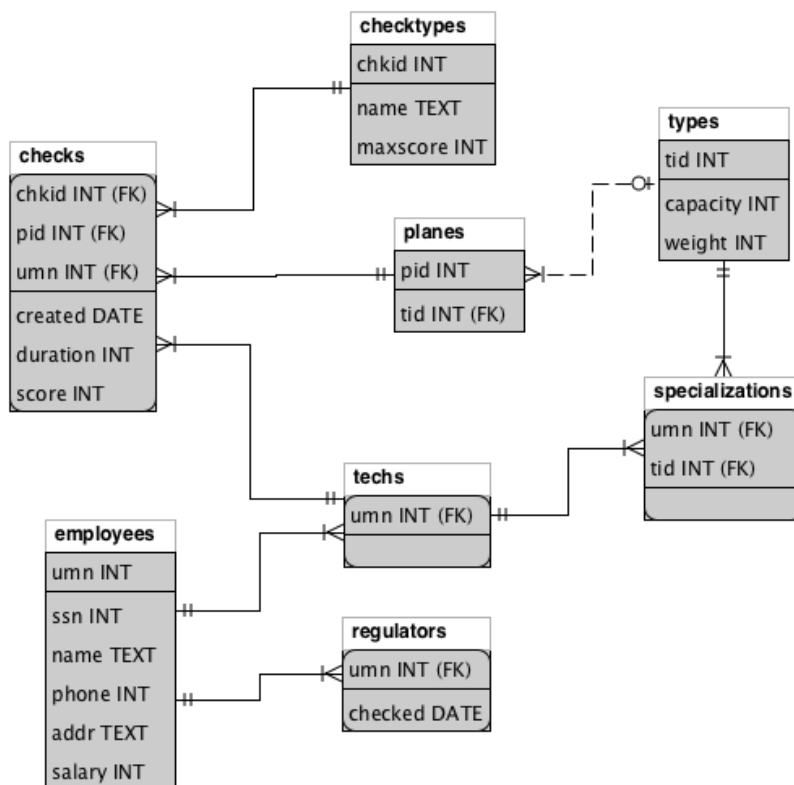
Οι επιλογές μας είχαν και τα μειονεκτήματά τους, όπως αναφέρονται παρακάτω:

- Η MySQL δεν υποστηρίζει constraint checking, οπότε οι περιορισμοί που θέλαμε να υπάρχουν εφαρμόστηκαν στην host γλώσσα, και απλώς αναφέρονται στα DDL κατασκευής της βάσης για λόγους πληρότητας.

2 Σχεδιασμός Βάσης

Για την υλοποίηση της βάσης μας, χρησιμοποιήσαμε το σχεσιακό μοντέλο, όπως το δημιουργήσαμε για την 1^η άσκηση του μαθήματος (όπου αναλύονται και σε μεγαλύτερο βάθος οι επιλογές και οι παραδοχές μας).

Το σχήμα του relationship model, παρατίθεται παρακάτω:



Σχήμα 1: Διάγραμμα σχεσιακού μοντέλου

Στη σχεδίαση αυτή, προσθέσαμε ορισμένους περιορισμούς, οι οποίοι προκύπτουν λογικά και υλοποιήθηκαν με check constraints στη MySQL. Αυτοί είναι:

- Σε όσες σχέσεις χρησιμοποιούν ξένα κλειδιά, υπάρχουν συνθήκες 'ON DELETE' και 'ON UPDATE' που πραγματοποιούν μια ενέργεια (συνήθως CASCADE).
- Στην τιμή score του πίνακα checks υλοποιήθηκε check ώστε να έχει τιμή όχι μεγαλύτερη από το maxscore του αντίστοιχου checktype.

Εκτός αυτών, για πληθώρα πεδίων που θα πρέπει να είναι μη αρνητικά, χρησιμοποιήθηκε ως datatype το unsigned int, ώστε να μη συνταχθούν αχρείαστα constraints.

Όπως αναφέραμε, η MySQL δεν υποστηρίζει constraint checking, επομένως οι περιορισμοί υλοποιήθηκαν στην host γλώσσα (php) με χρήση φίλτρων, όπου χρειάστηκαν, έως ότου υπάρξει υποστήριξη από το DBMS.

3 DDL Script κατασκευής βάσης

Παρατίθεται το DDL script που χρησιμοποιήθηκε για να κατασκευαστεί η βάση:

```
CREATE TABLE images (  
    imageid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    size INT UNSIGNED DEFAULT NULL,  
    width INT UNSIGNED DEFAULT NULL,  
    height INT UNSIGNED DEFAULT NULL  
) ENGINE=InnoDB;  
CREATE TABLE employees (  
    umn INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ssn INT UNSIGNED UNIQUE,  
    name TEXT NOT NULL,  
    imageid INT DEFAULT NULL,  
    phone VARCHAR( 32 ),  
    addr TEXT,  
    salary INT UNSIGNED  
    FOREIGN KEY ( imageid ) REFERENCES images( imageid )  
        ON DELETE SET NULL ON UPDATE CASCADE  
) ENGINE=InnoDB;  
CREATE TABLE regulators (  
    umn INT UNSIGNED NOT NULL PRIMARY KEY,  
    checked DATE NOT NULL,  
    FOREIGN KEY ( umn ) REFERENCES employees( umn )  
        ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;  
CREATE TABLE techs (  
    umn INT UNSIGNED NOT NULL PRIMARY KEY,  
    FOREIGN KEY ( umn ) REFERENCES employees( umn )  
        ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;  
CREATE TABLE checktypes (  
    checktypeid INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name TEXT,  
    maxscore INT UNSIGNED NOT NULL  
) ENGINE=InnoDB;  
CREATE TABLE planetypes (
```

```

        tid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR( 32 ) NOT NULL,
        capacity INT UNSIGNED DEFAULT NULL,
        weight INT UNSIGNED DEFAULT NULL,
        imageid INT DEFAULT NULL,
        FOREIGN KEY ( imageid ) REFERENCES images( imageid )
            ON DELETE SET NULL ON UPDATE CASCADE
        CONSTRAINT chk_planetypes CHECK
            ( capacity > 0 AND weight > 0 )
    ) ENGINE=InnoDB;
CREATE TABLE planes (
    pid INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tid INT UNSIGNED NOT NULL,
    FOREIGN KEY ( tid ) REFERENCES planetypes( tid )
        ON DELETE CASCADE ON UPDATE CASCADE
    ) ENGINE=InnoDB;
CREATE TABLE specializations (
    umn INT UNSIGNED NOT NULL,
    tid INT UNSIGNED NOT NULL,
    PRIMARY KEY ( umn, tid ),
    FOREIGN KEY ( umn ) REFERENCES techs( umn )
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY ( tid ) REFERENCES planetypes( tid )
        ON DELETE CASCADE ON UPDATE CASCADE
    ) ENGINE=InnoDB;
CREATE TABLE checks (
    checkid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    checktypeid INT UNSIGNED NOT NULL,
    pid INT UNSIGNED NOT NULL,
    umn INT UNSIGNED,
    created DATETIME,
    duration INT UNSIGNED NOT NULL,
    score INT UNSIGNED NOT NULL,
    FOREIGN KEY ( checktypeid ) REFERENCES checktypes( checktypeid )
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY ( pid ) REFERENCES planes( pid )
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY ( umn ) REFERENCES techs( umn )
        ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT chk_checks CHECK
        ( score > ( SELECT maxscore FROM checktypes t
            WHERE checktypeid = t.checktypeid )
            AND duration > 0 )
    ) ENGINE=InnoDB;
CREATE TABLE users (
    uid INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(32),
    password CHAR(64),
    passwordsalt CHAR(8),
    email VARCHAR(64)
    ) ENGINE=InnoDB;
CREATE VIEW workers AS
    SELECT
        e.*,

```

```

        ( CASE WHEN t.umn IS NOT NULL THEN 'tech'
              WHEN r.umn IS NOT NULL THEN 'regulator'
              ELSE '' END ) AS occ
FROM
    employees e
  LEFT JOIN
    techs t ON t.umn = e.umn
  LEFT JOIN
    regulators r ON r.umn = e.umn;

```

Παραπάνω, φαίνεται και η κατασκευή του view "workers", το οποίο αντιστοιχίζει σε κάθε εργαζόμενο την ιδιότητά του (και χρησιμοποιήθηκε αργότερα για την εξαγωγή στατιστικών των εργαζομένων). Επίσης, φαίνεται και η χρήση nested query, προκειμένου να οριστεί ο περιορισμός του πεδίου score από το maxscore του αντίστοιχου ελέγχου.

4 SQL queries

Σε όλες τις σχέσεις της βάσεις μας έχουν δημιουργηθεί queries για εισαγωγή, ενημέρωση, διαγραφή και επιλογή μίας ή όλων των εγγραφών, τα οποία δε χρειάζονται καμία εξήγηση.

4.1 Εργαζόμενοι

Στους εργαζόμενους έχουμε το 1ο ερώτημα με συναθροιστική συνάρτηση καθώς και ομαδοποίηση, που μας επιστρέφει τον ελάχιστο, μέσο και μέγιστο μισθό για κάθε απασχόληση.

```

SELECT
    occ, MIN( salary ) AS minimum, MAX( salary ) AS maximum, AVG( salary ) AS average
FROM
    workers
GROUP BY
    occ

```

```

INSERT INTO
    employees
SET
    umn = :umn,
    ssn = :ssn,
    name = :name,
    phone = :phone,
    addr = :addr,
    salary = :salary

```

```

DELETE FROM
    employees
WHERE
    umn = :umn

```

```

SELECT
    'tech' as occ, e.*, i.width, i.height
FROM
    employees e
LEFT JOIN
    techs t ON t.umn = e.umn
LEFT JOIN
    images i USING ( imageid )
WHERE
    ( :umn = e.umn AND t.umn IS NOT NULL )
LIMIT 1

```

```

SELECT
    'regulator' as occ, e.*, i.width, i.height, r.checked
FROM
    employees e
LEFT JOIN
    regulators r ON r.umn = e.umn
LEFT JOIN
    images i USING ( imageid )
WHERE
    ( :umn = e.umn AND r.umn IS NOT NULL )
LIMIT 1

```

```

SELECT
    'other' as occ, e.*, i.width, i.height
FROM
    employees e
LEFT JOIN
    images i USING ( imageid )
WHERE
    :umn = e.umn
LIMIT 1

```

```

SELECT
    e.*, i.width, i.height, r.checked
FROM
    employees e
LEFT JOIN
    images i USING ( imageid )
LEFT JOIN
    techs t ON t.umn = e.umn
LEFT JOIN
    regulators r ON r.umn = e.umn
WHERE
    ( '' = 'tech' AND t.umn IS NOT NULL )
    OR ( '' = 'regulator' AND r.umn IS NOT NULL )
    OR ( '' = '' )
ORDER BY umn

```

4.2 Τύποι Αεροσκαφών

INSERT INTO

planetypes

SET

name = :name,
weight = :weight,
capacity = :capacity

UPDATE

planetypes

SET

name = :name,
weight = :weight,
capacity = :capacity

WHERE

tid = :tid

SELECT

*

FROM

planetypes

WHERE

tid = :tid

DELETE FROM

planetypes

WHERE

tid = :tid

SELECT

*

FROM

planetypes

4.3 Αεροσκάφη

Η επιλογή των planes, χρησιμοποιεί INNER JOIN με τα planetypes, προκειμένου να εμφανίσουμε πληροφορίες σχετικά με το αεροπλάνο.

Υπάρχει και ένα aggregate query όπου καταμετράται το πλήθος των αεροσκαφών που υπάρχουν για κάθε δεδομένο planetype.

INSERT INTO

planes

SET

pid = :pid,
tid = :tid


```
DELETE FROM
    planes
WHERE
    pid = :pid
```

```
UPDATE
    planes
SET
    pid = :pid ,
    tid = :tid
WHERE
    pid = :pid
```

```
SELECT
    t.* , p.*
FROM
    planes p
INNER JOIN
    planetypes t
ON
    p.tid = t.tid
WHERE
    p.pid = :pid
LIMIT 1
```

```
SELECT
    t.* , p.*
FROM
    planes p
CROSS JOIN
    planetypes t ON t.tid = p.tid
```

```
SELECT
    name, p.tid , count( p.tid ) AS planeCount
FROM
    planes p
CROSS JOIN
    planetypes t ON t.tid = p.tid
GROUP BY
    name
```

4.4 Ελεγκτές Εναέριας Κυκλοφορίας

```
INSERT INTO
    regulators
SET
    umn = :umn
    checked = :checked
```

```

DELETE FROM
    regulators
WHERE
    umn = :umn

```

```

SELECT
    *
FROM
    regulators
WHERE
    umn = :umn

```

```

SELECT
    e.*, i.width, i.height, r.checked
FROM
    employees e
LEFT JOIN
    images i USING ( imageid )
LEFT JOIN
    techs t ON t.umn = e.umn
LEFT JOIN
    regulators r ON r.umn = e.umn
WHERE
    ( 'regulator' = 'tech' AND t.umn IS NOT NULL )
    OR ( 'regulator' = 'regulator' AND r.umn IS NOT NULL )
    OR ( 'regulator' = '' )
ORDER BY umn

```

4.5 Τεχνικοί

```

INSERT INTO
    techs
SET
    umn = :umn

```

```

DELETE FROM
    techs
WHERE
    umn = :umn

```

```

SELECT
    *
FROM
    techs
WHERE
    umn = :umn

```

```

SELECT
    e.*, i.width, i.height, r.checked
FROM
    employees e
LEFT JOIN
    images i USING ( imageid )
LEFT JOIN
    techs t ON t.umn = e.umn
LEFT JOIN
    regulators r ON r.umn = e.umn
WHERE
    ( 'tech' = 'tech' AND t.umn IS NOT NULL )
    OR ( 'tech' = 'regulator' AND r.umn IS NOT NULL )
    OR ( 'tech' = '' )
ORDER BY umn

```

4.6 Ειδικότητες

Στις επιλογές των specializations πραγματοποιούνται inner joins με τους πίνακες employees και planetypes, προκειμένου να εμφανίσουμε χρήσιμες πληροφορίες.

```

INSERT INTO
    specializations
SET
    umn = :umn,
    tid = :tid

```

```

DELETE FROM
    specializations
WHERE
    umn = :umn,
    tid = :tid

```

```

SELECT
    s.*, e.name AS eName, t.name AS tName
FROM
    specializations s
INNER JOIN
    planetypes t
ON
    t.tid = s.tid
INNER JOIN
    employees e
ON
    s.umn = e.umn
WHERE
    s.tid = :tid
    s.umn = :umn
LIMIT 1

```

```

SELECT
    t.name AS tName, s.* , e.name AS eName
FROM
    specializations s
INNER JOIN
    planetypes t
ON
    t.tid = s.tid
INNER JOIN
    employees e
ON
    s.umn = e.umn
ORDER BY eName

```

4.7 Έλεγχοι

Η επιλογή στους ελέγχους πραγματοποιεί CROSS JOIN με 4 διαφορετικές σχέσεις, προκειμένου να αντλήσουμε όλες τις πληροφορίες που θεωρήσαμε χρήσιμες για το χρήστη.

Επιπλέον, υπάρχει και ένα query ομαδοποίησης με περιορισμό (GROUP BY / HAVING) όπου μας εμφανίζονται τα ελαττωματικά αεροσκάφη, δηλαδή αυτά που έχουν συνολική επίδοση κάτω από 50% στους ελέγχους που έχουν υποβληθεί.

4.8 Αρχεία Εικόνων

```

INSERT INTO
    images
SET
    size = :size ,
    width = :width ,
    height = :height

```