

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

9^ο ΕΞΑΜΗΝΟ

HMMYStat

2^η ΕΡΓΑΣΙΑ: ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Γέραλη Μαρίνα 7414 - mngerali@auth.gr

Δημανίδης Αναστάσιος 7422 - dhmtasos@gmail.com

Ζαφειρίου Ιωάννης 7429 - ioanniza@gmail.com



Θεσσαλονίκη, 2/3/2015

Περιεχόμενα

Εισαγωγή.....	5
Περιγραφή της βάσης.....	5
Υλοποίηση.....	5
Τυπογραφικές Παραδοχές.....	7
Διάγραμμα Οντοτήτων-Συσχετίσεων	7
Περιγραφή Πινάκων	10
Ρόλοι Χρηστών.....	31
Όψεις	35
Όψη Year_Statistics	35
Όψη Exams_Subjects	42
Παραδείγματα Ερωτημάτων	49
Triggers.....	57
Εισαγωγή δεδομένων στην βάση	61
Τρόποι εισαγωγής.....	61
Αυτοματοποίηση	62
Εισαγωγή Δεδομένων	62
Παράρτημα	77

Λίστα Σχημάτων

Σχήμα 1: Διάγραμμα Οντοτήτων Συσχετίσεων.....	8
Σχήμα 2: Διάγραμμα Συσχετίσεων από το Microsoft SQL Server	9
Σχήμα 3: Πίνακας ΦΟΙΤΗΤΗΣ.....	10
Σχήμα 4: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ.....	11
Σχήμα 5: Πίνακας ΑΠΟΦΟΙΤΟΣ.....	11
Σχήμα 6: Παράδειγμα του πίνακα ΑΠΟΦΟΙΤΟΣ.....	12
Σχήμα 7: Πίνακας ΤΟΜΕΑΣ.....	12
Σχήμα 8: Παράδειγμα του πίνακα ΤΟΜΕΑΣ.....	13
Σχήμα 9: Πίνακας ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ.....	13
Σχήμα 10: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ.....	14
Σχήμα 11: Πίνακας ΠΡΑΚΤΙΚΗ.....	14
Σχήμα 12: Παράδειγμα του πίνακα ΠΡΑΚΤΙΚΗ.....	15
Σχήμα 13: Πίνακας ΦΟΙΤΗΤΗΣ_ΠΡΑΚΤΙΚΗ	15
Σχήμα 14: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ_ΠΡΑΚΤΙΚΗ	16
Σχήμα 15: Πίνακας ERASMUS	16
Σχήμα 16: Παράδειγμα του πίνακα ERASMUS	16
Σχήμα 17: Πίνακας ΦΟΙΤΗΤΗΣ_ERASMUS.....	17
Σχήμα 18: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ_ERASMUS.....	17
Σχήμα 19: Πίνακας ΚΑΘΗΓΗΤΗΣ	18
Σχήμα 20: Παράδειγμα του πίνακα ΚΑΘΗΓΗΤΗΣ	18
Σχήμα 21: Πίνακας ΜΑΘΗΜΑ.....	19
Σχήμα 22: Παράδειγμα του πίνακα ΜΑΘΗΜΑ.....	19
Σχήμα 23: Σχέση ΜΑΘΗΜΑ_ΤΟΜΕΑΣ	20
Σχήμα 24: Παράδειγμα της σχέσης ΜΑΘΗΜΑ_ΤΟΜΕΑΣ	20
Σχήμα 25: Πίνακας ΕΞΕΤΑΣΤΙΚΗ.....	21
Σχήμα 26: Παράδειγμα του πίνακα ΕΞΕΤΑΣΤΙΚΗ.....	21
Σχήμα 27: Πίνακας ΔΙΔΑΣΚΑΛΙΑ.....	22
Σχήμα 28: Παράδειγμα του πίνακα ΔΙΔΑΣΚΑΛΙΑ.....	22
Σχήμα 29: Σχέση ΚΑΘΗΓΗΤΗΣ_ΔΙΔΑΣΚΑΛΙΑ	23
Σχήμα 30: Παράδειγμα της σχέσης ΚΑΘΗΓΗΤΗΣ_ΔΙΔΑΣΚΑΛΙΑ	23
Σχήμα 31: Σχέση ΕΞΕΤΑΣΗ_ΜΑΘΗΜΑΤΟΣ.....	24
Σχήμα 32: Παράδειγμα της σχέσης ΕΞΕΤΑΣΗ_ΜΑΘΗΜΑΤΟΣ	25
Σχήμα 33: Πίνακας ΑΙΘΟΥΣΑ.....	25
Σχήμα 34: Παράδειγμα του πίνακα ΑΙΘΟΥΣΑ.....	26
Σχήμα 35: Πίνακας ΣΥΓΓΡΑΜΜΑ.....	26
Σχήμα 36: Παράδειγμα του πίνακα ΣΥΓΓΡΑΜΜΑ.....	27
Σχήμα 37: Σχέση ΔΗΛΩΣΗ_ΣΥΓΓΡΑΜΜΑΤΟΣ	27
Σχήμα 38: Παράδειγμα της σχέσης ΔΗΛΩΣΗ_ΣΥΓΓΡΑΜΜΑΤΟΣ.....	28
Σχήμα 39: Σχέση ΒΑΘΜΟΛΟΓΙΑ	28
Σχήμα 40: Παράδειγμα της σχέσης ΒΑΘΜΟΛΟΓΙΑ.....	29
Σχήμα 41: Σχέση ΔΙΑΛΕΞΗ.....	29
Σχήμα 42: Παράδειγμα της σχέσης ΔΙΑΛΕΞΗ	30
Σχήμα 43: Ρόλοι χρηστών	32

Σχήμα 45: Αποτελέσματα όψης Students_YearIn	36
Σχήμα 46: Αποτελέσματα όψης Year_Entrants	37
Σχήμα 47: Αποτελέσματα όψης Year_Graduates	37
Σχήμα 48: Αποτελέσματα όψης Year_Internships.....	38
Σχήμα 49: Αποτελέσματα όψης Year_Transferred.....	38
Σχήμα 50: Αποτελέσματα όψης Year_Erasmus	39
Σχήμα 51: Αποτελέσματα όψης Year_Picked_Electronics.....	39
Σχήμα 52: Αποτελέσματα όψης Year_Picked_Energy	40
Σχήμα 53: Αποτελέσματα όψης Year_Picked_Telecommunications	40
Σχήμα 54: Αποτελέσματα όψης Year_Average_Grade.....	41
Σχήμα 55: Αποτελέσματα όψης Year_Statistics	42
Σχήμα 57: Αποτελέσματα όψης Exams-Basic_Info.....	43
Σχήμα 58: Αποτελέσματα όψης Exams_Taken_Results	44
Σχήμα 59: Αποτελέσματα όψης Exams_Passed	45
Σχήμα 60: Αποτελέσματα όψης Exams_Failed	45
Σχήμα 61: Αποτελέσματα όψης Exams_Participation	46
Σχήμα 62: Αποτελέσματα όψης Exams_Average_Grade	47
Σχήμα 63: Αποτελέσματα όψης Exams_Subjects	48
Σχήμα 64: Αποτελέσματα πρώτου query	49
Σχήμα 65: Αποτελέσματα δεύτερου query.....	50
Σχήμα 66: Αποτελέσματα τρίτου query.....	50
Σχήμα 67: Αποτελέσματα τέταρτου query	51
Σχήμα 68: Αποτελέσματα πέμπτου query	51
Σχήμα 69: Αποτελέσματα έκτου query.....	52
Σχήμα 70: Αποτέλεσμα έβδομου query	52
Σχήμα 71: Αποτέλεσμα όγδοου query.....	52
Σχήμα 72: Αποτελέσματα ένατου query.....	53
Σχήμα 73: Αποτέλεσμα δέκατου query	53
Σχήμα 74: Αποτελέσματα ενδέκατου query.....	54
Σχήμα 75: Αποτελέσματα δωδέκατου query.....	55
Σχήμα 76: Αποτελέσματα δέκατου-τρίτου query.....	56

Εισαγωγή

Περιγραφή της βάσης

Στα πλαίσια της διδασκαλίας του μαθήματος “Βάσεις Δεδομένων”, κληθήκαμε να σχεδιάσουμε και να υλοποιήσουμε μία βάση δεδομένων της επιλογής μας. Το ζήτημα το οποίο αποτέλεσε ερέθισμα για την επιλογή του θέματος της βάσης είναι η ολοκληρωμένη ενημέρωση των φοιτητών κατά τη διάρκεια των σπουδών τους στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης (ΑΠΘ). Αυτό που μας οδήγησε στην επιλογή ενός τέτοιου θέματος για βάση δεδομένων, είναι η απουσία κάποιου τέτοιου μηχανισμού παροχής πληροφορίας προς τους φοιτητές. Έτσι τα χαρακτηριστικά μιας τέτοιας βάσης θα πρέπει να είναι:

- Η συλλογή όλων των πληροφοριών που απασχολούν το τμήμα και τους φοιτητές
- Η οργάνωση της πληροφορίας με τέτοιο τρόπο, ώστε τα ερωτήματα εισαγωγής, διαγραφής, επιλογής και ενημέρωσης να είναι απλά.
- Η δυνατότητα να αντληθούν πάσης φύσεως πληροφορίες, «με μία εντολή».
- Η ικανότητα να αξιοποιηθεί από μία εφαρμογή, η οποία θα προβάλλει τις πληροφορίες, με φιλικό προς το χρήστη τρόπο.

Για τη λειτουργία της βάσης θα πρέπει να συνεργαστούν τόσο οι διδάσκοντες όσο και η γραμματεία του τμήματος.

Έτσι, δημιουργήσαμε τη βάση δεδομένων «HMMYStat» που λειτουργεί, καταρχάς, ως άλλη μία ηλεκτρονική γραμματεία συλλέγοντας ανώνυμα τις επιδόσεις των φοιτητών του εν λόγω τμήματος στα διάφορα μαθήματα, καθώς και τις προτιμήσεις τους σε βιβλία, τομείς, συμμετοχή στο πρόγραμμα Erasmus, στην εκπόνηση ή όχι πρακτικής και ούτω καθεξής. Η διαφοροποίησή της, ωστόσο, από τις μέχρι τώρα ηλεκτρονικές γραμματείες είναι ότι τα δεδομένα που συλλέγονται χρησιμοποιούνται στη δημιουργία και παρουσίαση στατιστικών μεγεθών τα οποία θα βοηθήσουν τόσο τους φοιτητές στην ακαδημαϊκή τους πορεία όσο και το πανεπιστήμιο, γενικότερα, στα πλαίσια της εύρυθμης λειτουργίας του (για παράδειγμα καλύτερη κατανομή των διαθέσιμων αιθουσών, ιδίως σε μαθήματα όπου παρατηρούνται προβλήματα συνωστισμού).

Υλοποίηση

Σε αυτό το παραδοτέο υλοποιήθηκε η βάση HMMYStat, στο σύστημα διαχείρισης βάσεων δεδομένων της Microsoft, τον SQL Server (ο οποίος παρέχεται δωρεάν για τους φοιτητές από το πρόγραμμα Dreamspark της Microsoft για πανεπιστήμια). Κατά την υλοποίηση βασιστήκαμε ολοκληρωτικά στην λογική σχεδίαση του πρώτου παραδοτέου. Η βάση γράφτηκε σε έναν υπολογιστή και η ομάδα συνδεόταν σε αυτόν είτε τοπικά είτε απομακρυσμένα για να πραγματοποιήσει αλλαγές. Κατά την

δημιουργία της χρησιμοποιήθηκε το περιβάλλον SQL Server 2014 Management Studio. Η υλοποίηση περιλαμβάνει.

- Δημιουργία πινάκων και περιορισμών ακεραιότητας αυτών
- Δημιουργία Σχεσιακού Διαγράμματος
- Υλοποίηση trigger σε sql, και δημιουργία νέων trigger
- Υλοποίηση ερωτημάτων σε sql και δημιουργία νέων ερωτημάτων
- Εισαγωγή δεδομένων στην βάση

Τυπογραφικές Παραδοχές

Για την καλύτερη κατανόηση αυτού του εγγράφου, θεωρήθηκε σωστό να δημιουργήσουμε κάποιες συμβάσεις σχετικά με τον τρόπο γραφής του παρόντος κειμένου. Συγκεκριμένα, το κείμενο γράφεται σε γραμματοσειρά Calibri 11, ενώ οι επικεφαλίδες είναι Calibri 16 και οι ύπο-κεφαλίδες Calibri 12. Οι σχέσεις στο διάγραμμα Οντοτήτων – Συσχετίσεων (ρόμβοι) αναφέρονται ως «σχέσεις». Αντίθετα, οι οντότητες αναφέρονται ως «πίνακες». Οι τίτλοι των πινάκων επισημαίνονται με ΚΕΦΑΛΑΙΑ γράμματα. Τα γνωρίσματα μέσα στο κείμενο αναγράφονται με *πλάγια γραφή*. Το πρωτεύον κλειδί κάθε πίνακα κάθε φορά εμφανίζεται σε υπογραμμισμένη γραφή.

Συντομογραφίες:

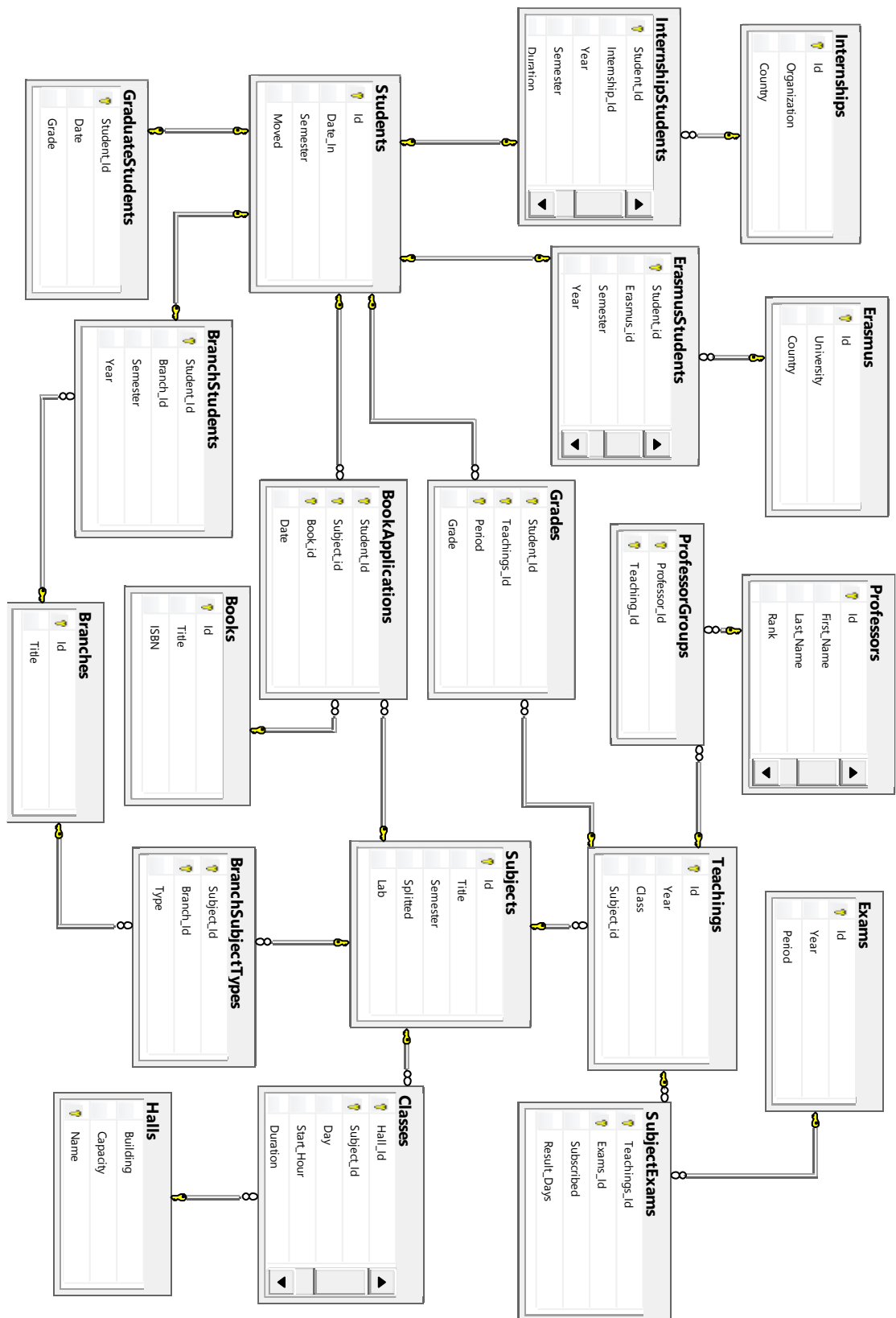
- NN - not null
- Ο/Σ - Οντοτήτων Συσχετίσεων
- ΗΜΜΥ – Ηλεκτρολόγοι Μηχανικοί και Μηχανικοί Ηλεκτρονικών Υπολογιστών
- ΑΕΜ – Αριθμός Ειδικού Μητρώου

Διάγραμμα Οντοτήτων-Συσχετίσεων

Στην επόμενη σελίδα, παραθέτουμε το διάγραμμα οντοτήτων συσχετίσεων, όπως αυτό προέκυψε από την πρώτη εργασία, καθώς, επίσης, και αυτό που δημιούργησε ο Microsoft SQL Server.

The diagram is an Entity-Relationship (ER) model for a library system. It features several entities (red rectangles) and their attributes (blue ovals). Relationships are indicated by green diamonds.

- Entities and Attributes:**
 - Πρακτική**: Id, Οργανισμός, Χώρα
 - Erasmus**: Id, Πανεπιστήμιο, Χώρα
 - Κωδικητήρας**: Id, Όνομα, Επίθετο, Βαθμίδα, Περιγραφή
 - Διδακτορία**: Id, Έτος, Επιστήμη_Επιστήμη
 - Επιστημονική**: Id, Έτος, Θεματικό, Επιστήμη_Επιστήμη
 - Μαθήματα**: Id, Έτος, Χωρίζεται, Τίτλος, Επιστήμη_Επιστήμη
 - Αντικείμενα**: Id, Τίτλος, ISBN, Έτος
 - Βιβλιοθήκη**: Id, Όνομα, Χωρίζεται, Τίτλος, ISBN
 - Αλβουρα**: Id, Όνομα, Χωρίζεται, Τίτλος, ISBN
 - Τομείς**: Id, Όνομα, Χωρίζεται, Τίτλος, ISBN
- Relationships (Green Diamonds):**
 - Επιστήμη**: Connects Πρακτική (1) to Κωδικητήρας (N).
 - Επιστήμη_Επιστήμη**: Connects Erasmus (1) to Κωδικητήρας (N).
 - Επιστήμη_Επιστήμη**: Connects Κωδικητήρας (M) to Διδακτορία (N).
 - Επιστήμη_Επιστήμη**: Connects Κωδικητήρας (M) to Επιστημονική (N).
 - Επιστήμη_Επιστήμη**: Connects Κωδικητήρας (M) to Μαθήματα (N).
 - Επιστήμη_Επιστήμη**: Connects Διδακτορία (1) to Επιστημονική (N).
 - Επιστήμη_Επιστήμη**: Connects Επιστημονική (1) to Μαθήματα (N).
 - Επιστήμη_Επιστήμη**: Connects Μαθήματα (M) to Αντικείμενα (N).
 - Επιστήμη_Επιστήμη**: Connects Αντικείμενα (M) to Βιβλιοθήκη (N).
 - Επιστήμη_Επιστήμη**: Connects Βιβλιοθήκη (M) to Αλβουρα (N).
 - Επιστήμη_Επιστήμη**: Connects Αλβουρα (M) to Τομείς (N).
 - Επιστήμη_Επιστήμη**: Connects Τομείς (M) to Μαθήματα (N).




Σχήμα 2: Διάγραμμα Συσχετίσεων από το Microsoft SQL Server

Περιγραφή Πινάκων

Σε αυτήν την ενότητα, θα παρουσιαστούν οι πίνακες που απαρτίζουν τη βάση HMMYStat. Για κάθε πίνακα, θα περιγράφεται σύντομα το τι υλοποιεί, θα δίνεται το σχήμα του και οι περιορισμοί του σε SQL, ενώ, τέλος, θα παρατίθενται συμπληρωμένοι με αρκετές εγγραφές. Να σημειωθεί ότι δεν θα αναφέρεται το πρωτεύον κλειδί κάθε πίνακα, αφού αυτό θα φαίνεται από τη σχεδίαση που προέκυψε από τον Microsoft SQL Server. Έτσι, λοιπόν, έχουμε τους εξής πίνακες:

- ✓ Πίνακας «ΦΟΙΤΗΤΗΣ» ο οποίος περιλαμβάνει όλα τα χαρακτηριστικά ενός φοιτητή, όπως το id του (το AEM), το έτος εισαγωγής, το αν είναι μετεγγραφόμενος και σε ποιο εξάμηνο βρίσκεται.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Date_In	date	<input type="checkbox"/>
	Semester	tinyint	<input type="checkbox"/>
	Moved	bit	<input type="checkbox"/>

Σχήμα 3: Πίνακας ΦΟΙΤΗΤΗΣ

Οι γενικότεροι περιορισμοί που πρέπει να έχει αυτός ο πίνακας είναι δύο. Ο πρώτος αφορά το έτος εισαγωγής του φοιτητή που πρέπει να είναι μεγαλύτερο από το έτος ιδρύσεως του τμήματος (1972). Ο περιορισμός σε SQL φαίνεται παρακάτω.

```
CONSTRAINT [CK_Students_Date] CHECK (([Date_In]>='1972-01-01'))
```

Ο δεύτερος αφορά το εξάμηνο φοίτησης του φοιτητή, το οποίο πρέπει να είναι μεγαλύτερο ή ίσο του 1 και μικρότερο ή ίσο του 20, όπως ορίζεται από τα επιτρεπτά χρόνια φοίτησης. Ο κώδικας σε SQL φαίνεται παρακάτω.

```
CONSTRAINT [CK_Students_Semester] CHECK (([Semester]>=(1) AND [Semester]<=(20)))
```

Στην επόμενη σελίδα, φαίνεται ο εν λόγω πίνακας συμπληρωμένος με δεδομένα.

	Id	Date_In	Semester	Moved
▶	6600	2007-09-05	15	False
	6601	2007-09-01	15	False
	6602	2007-09-13	10	False
	6603	2007-09-07	15	False
	6604	2007-09-21	10	False
	6605	2007-09-29	10	False
	6606	2007-09-11	10	False
	6607	2007-09-25	15	False
	6608	2007-09-22	10	False
	6609	2007-09-06	10	False
	6610	2007-09-23	15	False
	6611	2007-09-19	10	False
	6612	2007-09-26	10	False
	6613	2007-09-04	10	False
	6614	2007-09-03	10	False
	6615	2007-09-08	15	False
	6616	2007-09-04	10	False
	6617	2007-09-23	15	False
	6618	2007-09-05	15	True
	6619	2007-09-20	15	False
	6620	2007-09-08	10	False
	6621	2007-09-12	10	False
	6622	2007-09-09	15	False
	6623	2007-09-10	15	False

Σχήμα 4: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ

- ✓ Πίνακας «ΑΠΟΦΟΙΤΟΣ» ο οποίος περιέχει το έτος αποφοίτησης και το βαθμό όλων των αποφοίτων του τμήματος.

	Column Name	Data Type	Allow Nulls
🔑	Student_Id	int	<input type="checkbox"/>
	Date	date	<input type="checkbox"/>
	Grade	real	<input type="checkbox"/>

Σχήμα 5: Πίνακας ΑΠΟΦΟΙΤΟΣ

Ο πίνακας αυτός έχει έναν περιορισμό αναφορικής ακεραιότητας, ότι το γνώρισμα Student_Id είναι ξένο κλειδί του Id από τον πίνακα ΦΟΙΤΗΤΗΣ.

CONSTRAINT [FK_Students_Graduates] **FOREIGN KEY**([Student_Id])**REFERENCES** [dbo].[Students] ([Id])

Επίσης, περιέχει δύο ακόμη περιορισμούς, ο ένας εκ των οποίων αφορά το έτος αποφοίτησης που θα πρέπει να είναι μεγαλύτερο από το 1977, όσο δηλαδή είναι το έτος ίδρυσης συν τα ελάχιστα έτη φοίτησης.

CONSTRAINT [CK_GraduateStudents_Date] **CHECK** (([Date]>='1977-01-01'))

Ο δεύτερος περιορισμός απλά ορίζει ότι ο βαθμός πτυχίου δεν μπορεί να είναι μικρότερος του 5, όπως και φαίνεται παρακάτω.

CONSTRAINT [CK_GraduateStudents_Grade] **CHECK** (([Grade]>=(5) AND [Grade]<=(10)))

Παρακάτω φαίνεται ο πίνακας ΑΠΟΦΟΙΤΟΣ συμπληρωμένος με κάποιες εγγραφές.

	Student_Id	Date	Grade
▶	6600	2014-07-29	6
	6601	2014-07-29	6
	6603	2014-07-23	9
	6604	2012-07-08	10
	6605	2012-07-16	9
	6606	2012-07-17	9
	6607	2014-07-20	6
	6608	2012-07-20	6
	6609	2012-07-13	6
	6610	2014-07-15	6
	6612	2012-07-22	8
	6613	2012-07-10	6
	6615	2014-07-18	7
	6616	2012-07-01	7
	6617	2014-07-15	10
	6619	2014-07-29	7

Σχήμα 6: Παράδειγμα του πίνακα ΑΠΟΦΟΙΤΟΣ

- ✓ Πίνακας «ΤΟΜΕΑΣ» ο οποίος αποθηκεύει όλους τους τομείς του τμήματος.

	Column Name	Data Type	Allow Nulls
🔑	Id	tinyint	<input type="checkbox"/>
	Title	nvarchar(50)	<input type="checkbox"/>

Σχήμα 7: Πίνακας ΤΟΜΕΑΣ

Ο πίνακας αυτός δε δημιουργεί κανέναν περιορισμό. Καθώς οι μέχρι τώρα τομείς είναι τρεις, ο πίνακας αυτός περιέχει μόνο τρεις εγγραφές, όπως και φαίνεται παρακάτω.

	Id	Title
▶	2	Ενέργειας
	1	Ηλεκτρονικής
	3	Τηλεπικοινωνιών

Σχήμα 8: Παράδειγμα του πίνακα ΤΟΜΕΑΣ

- ✓ Πίνακας «ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ» ο οποίος περιλαμβάνει το έτος, το εξάμηνο και τον τομέα που επέλεξαν οι φοιτητές που πήραν τομέα.

🔑	Student_Id	int	<input type="checkbox"/>
	Branch_Id	tinyint	<input type="checkbox"/>
	Semester	tinyint	<input type="checkbox"/>
▶	Year	smallint	<input type="checkbox"/>
			<input type="checkbox"/>

Σχήμα 9: Πίνακας ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ

Όσον αφορά τους περιορισμούς ακεραιότητας του πίνακα ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ, αυτοί είναι δύο και αναφέρονται στα γνωρίσματα Student_Id και Branch_Id.

CONSTRAINT [FK_BranchStudents_Students] **FOREIGN KEY**([Student_Id])**REFERENCES** [dbo].[Students] ([Id])

CONSTRAINT [FK_BranchStudents_Branches] **FOREIGN KEY**([Branch_Id])**REFERENCES** [dbo].[Branches] ([Id])

Ένας ακόμη περιορισμός που θέτει ο συγκεκριμένος πίνακας είναι ότι το εξάμηνο στο οποίο ο εκάστοτε φοιτητής πήρε τομέα πρέπει να είναι μεγαλύτερο από 6.


CONSTRAINT [CK_BranchStudents_Semester] **CHECK** (([Semester]>=(6)))

Παρακάτω φαίνεται ο συμπληρωμένος πίνακας ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ:

	Student_Id	Branch_Id	Semester	Year
▶	6600	1	6	2010
	6601	2	6	2010
	6602	2	10	2012
	6603	3	13	2013
	6604	1	7	2010
	6605	1	7	2010
	6606	1	9	2011
	6607	3	6	2010
	6608	1	6	2010
	6609	1	7	2010
	6610	2	15	2014
	6611	1	6	2010
	6612	2	7	2010
	6613	1	8	2011
	6614	2	6	2010
	6615	2	14	2014
	6616	1	8	2011
	6617	1	8	2011
	6619	1	6	2010
	6620	2	6	2010

Σχήμα 10: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ_ΤΟΜΕΑ

- ✓ Πίνακας «ΠΡΑΚΤΙΚΗ» που περιέχει όλους τους οργανισμούς και τις χώρες στις οποίες μπορούν οι φοιτητές του τμήματος ΗΜΜΥ να εκπονοούν την πρακτική τους.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Organization	nvarchar(50)	<input type="checkbox"/>
	Country	nvarchar(50)	<input type="checkbox"/>


Σχήμα 11: Πίνακας ΠΡΑΚΤΙΚΗ

Ο πίνακας αυτός δε θέτει κανέναν περιορισμό. Παρακάτω παρατίθεται συμπληρωμένος με κάποια παραδείγματα.

	Id	Organization	Country
	14	Microsoft	USA
	8	Cadence	Αγγλία
	16	Intenel	Ελλάδα
	15	Renel	Ελλάδα
	17	Sunwind	Ελλάδα
	10	ΔΕΔΔΗΕ	Ελλάδα
	13	Δημόκριτος	Ελλάδα
	18	IMET	Ελλάδα
	9	ΙΠΤΗΛ	Ελλάδα
	11	OTE	Ελλάδα
	19	CERN	Switzerland

Σχήμα 12: Παράδειγμα του πίνακα ΠΡΑΚΤΙΚΗ

- ✓ Πίνακας «ΦΟΙΤΗΤΗΣ_ΠΡΑΚΤΙΚΗ» που έχει τις πληροφορίες για τις πρακτικές που έχουν κάνει όλοι οι φοιτητές του τμήματος, όπως το id, το εξάμηνο, το έτος και τη διάρκεια της πρακτικής.

	Column Name	Data Type	Allow Nulls
	Student_Id	int	<input type="checkbox"/>
	Internship_Id	int	<input type="checkbox"/>
	Year	smallint	<input type="checkbox"/>
	Semester	tinyint	<input type="checkbox"/>
	Duration	tinyint	<input type="checkbox"/>

Σχήμα 13: Πίνακας ΦΟΙΤΗΤΗΣ_ΠΡΑΚΤΙΚΗ

Οι περιορισμοί ακεραιότητας του πίνακα αυτού είναι δύο και αφορούν τα γνωρίσματα Student_Id και Internship_Id, όπως ορίζονται και στη συνέχεια.

CONSTRAINT [FK_Students_InternshipStudents] **FOREIGN KEY**([Student_Id])**REFERENCES** [dbo].[Students] ([Id])

CONSTRAINT [FK_Internships_InternshipStudents] **FOREIGN KEY**([Internship_Id])
REFERENCES [dbo].[Internships] ([Id])

Δεν υπάρχουν περαιτέρω περιορισμοί που πρέπει να οριστούν σε αυτόν τον πίνακα. Μερικά παραδείγματα φαίνονται παρακάτω.

	Student_Id	Internship_Id	Year	Semester	Duration
	6623	10	2013	9	1
	6758	17	2012	9	2
	6799	15	2014	10	1
	7202	11	2013	9	2
	7399	13	2015	7	2
	7400	9	2015	7	1
	7414	18	2014	7	2
▶	7422	8	2015	7	3
	7429	14	2015	7	1
	7508	14	2014	6	1

Σχήμα 14: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ_ΠΡΑΚΤΙΚΗ

- ✓ Πίνακας «ERASMUS» που περιέχει πληροφορίες σχετικά με τις χώρες και τα πανεπιστήμια στα οποία μπορεί κάποιος φοιτητής να κάνει Erasmus.

	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	University	nvarchar(50)	<input type="checkbox"/>
	Country	nvarchar(50)	<input type="checkbox"/>

Σχήμα 15: Πίνακας ERASMUS

Ο πίνακας ERASMUS δεν έχει κάποιο περιορισμό ακεραιότητας ή γενικότερο περιορισμό, οπότε παραθέτουμε κατευθείαν τον συμπληρωμένο πίνακα.

	Id	University	Country
	2	Leuven	Belgium
	3	Brussel	Belgium
	4	Eindhoven	The Netherlands
	5	Windesheim	The Netherlands
	6	Aachen	Germany
	7	ETH	Switzerland
	8	Ecole	Switzerland
	9	Delft	The Netherlands

Σχήμα 16: Παράδειγμα του πίνακα ERASMUS

- ✓ Πίνακας «ΦΟΙΤΗΤΗΣ_ERASMUS» που περιέχει πληροφορίες σχετικά με τους φοιτητές που συμμετείχαν στο πρόγραμμα Erasmus, όπως το id, το εξάμηνο, το έτος και τη διάρκεια του προγράμματος Erasmus.

	Column Name	Data Type	Allow Nulls
🔑	Student_id	int	<input type="checkbox"/>
	Erasmus_id	int	<input type="checkbox"/>
	Semester	tinyint	<input type="checkbox"/>
	Year	smallint	<input type="checkbox"/>
	Duration	tinyint	<input type="checkbox"/>

Σχήμα 17: Πίνακας ΦΟΙΤΗΤΗΣ_ERASMUS

Οι περιορισμοί ακεραιότητας αυτού του πίνακα αφορούν τα γνωρίσματα Student_id και Erasmus_id και ορίζονται σε SQL στη συνέχεια. Δεν υπάρχουν περαιτέρω περιορισμοί.


```
CONSTRAINT [FK_ErasmusStudents_Students] FOREIGN KEY([Student_id])REFERENCES
[dbo].[Students] ([Id])
CONSTRAINT [FK_ErasmusStudents_Erasmus] FOREIGN KEY([Erasmus_id])REFERENCES
[dbo].[Erasmus] ([Id])
```

Ο πίνακας ΦΟΙΤΗΤΗΣ_ERASMUS συμπληρωμένος με εγγραφές φαίνεται παρακάτω.

	Student_id	Erasmus_id	Semester	Year	Duration
	7422	2	5	2012	1
	6623	3	6	2010	2
	7414	6	7	2013	1
	7429	8	6	2013	1
	6758	9	5	2010	2
	6799	7	6	2011	1
	7202	4	7	2011	3
	7508	5	5	2014	2
	7400	3	6	2012	1
	7399	9	7	2013	2

Σχήμα 18: Παράδειγμα του πίνακα ΦΟΙΤΗΤΗΣ_ERASMUS

- ✓ Πίνακας «ΚΑΘΗΓΗΤΗΣ» που αναπαριστά τον εκάστοτε καθηγητή και περιέχει το id, το ονοματεπώνυμο, τη βαθμίδα και τα πτυχία που έχει ο κάθε διδάσκων.

	Column Name	Data Type	Allow Nulls
	Id	smallint	<input type="checkbox"/>
	First_Name	nvarchar(50)	<input type="checkbox"/>
	Last_Name	nvarchar(50)	<input type="checkbox"/>
	Rank	nvarchar(50)	<input checked="" type="checkbox"/>
	Description	nvarchar(1000)	<input checked="" type="checkbox"/>


Σχήμα 19: Πίνακας ΚΑΘΗΓΗΤΗΣ

Δεν υπάρχει κάποιος περιορισμός ακεραιότητας ή γενικότερος περιορισμός για αυτόν τον πίνακα, οπότε προχωράμε στην παράθεση του συμπληρωμένου πίνακα.

	Id	Last_Name	First_Name	Rank	Description
▶	2	Δουλγέρη	Ζωή	Καθηγήτρια	Δίπλ. Ηλεκτρολ...
	3	Θεοχάρης	Ιωάννης	Καθηγητής	Δίπλ. Μηχανολ...
	4	Μήτκας	Περικλής	Καθηγητής	Δίπλ. Ηλεκτρολ...
	5	Πάγκαλος	Γεώργιος	Καθηγητής	Πτυχίο Μαθημ...
	6	Χασάπης	Γεώργιος	Καθηγητής	Δίπλ. Ηλεκτρολ...
	7	Χατζόπουλος	Αλκιβιάδης	Καθηγητής	Πτυχίο Φυσικο...
	8	Κεχαγιάς	Αθανάσιος	Αναπληρωτής ...	Δίπλωμα Ηλεκ...
	9	Κουγιουμτζής	Δημήτριος	Αναπληρωτής ...	Πτυχίο στα Μα...
	10	Μητράκος	Δημήτριος	Αναπληρωτής ...	Δίπλ. Μηχανολ...
	11	Πέτρου	Λουκάς	Αναπληρωτής ...	Δίπλ. Ηλεκτρολ...
	12	Ροβιθάκης	Γεώργιος	Αναπληρωτής ...	Δίπλ. Ηλεκτρολ...
	13	Ντελόπουλος	Αναστάσιος	Αναπληρωτής ...	Δίπλ. Ηλεκτρολ...
	14	Δοκουζιάννης	Σταύρος	Επίκουρος Καθ...	Dpl in Electroni...
	15	Πιτσιάνης	Νικόλαος	Επίκουρος Καθ...	Πτυχίο Τμήματ...
	16	Πιτσούλης	Λεωνίδας	Επίκουρος Καθ...	NULL
	17	Συμεωνίδης	Ανδρέας	Επίκουρος Καθ...	Δίπλωμα Ηλεκ...
	18	Παπαλάμπρου	Κωνσταντίνος	Λέκτορας	Δίπλωμα Μηχ...
	19	Δάϊος	Απόστολος	Επιστημονικός ...	Δίπλ. Μηχανολ...
	20	Καδή	Χριστίνα	Επιστημονικός ...	Πτυχίο Φυσικο...
	21	Σταμούλης	Γεώργιος	Επιστημονικός ...	Δίπλ. Ηλεκτρολ...
	23	Γιαννούλας	Νικόλαος	Εργαστηριακό ...	NULL
	24	Κωνσταντινίδης	Νικόλαος	Εργαστηριακό ...	NULL
	25	Χατζηαντωνίου	Κωνσταντίνος	Εργαστηριακό ...	NULL
	26	Κλούβας	Αλέξανδρος	Καθηγητής	Πτυχίο Φυσική...

Σχήμα 20: Παράδειγμα του πίνακα ΚΑΘΗΓΗΤΗΣ

- ✓ Πίνακας «ΜΑΘΗΜΑ» που αποθηκεύει όλες τις σχετικές με ένα μάθημα πληροφορίες, όπως το όνομα, το εξάμηνο, το αν διδάσκεται σε περισσότερα του ενός τμήματα και το αν έχει εργαστήριο.

	Column Name	Data Type	Allow Nulls
	Id	smallint	<input type="checkbox"/>
	Title	nvarchar(50)	<input type="checkbox"/>
	Semester	tinyint	<input type="checkbox"/>
	Splitted	bit	<input type="checkbox"/>
	Lab	bit	<input type="checkbox"/>

Σχήμα 21: Πίνακας ΜΑΘΗΜΑ

Ο μοναδικός περιορισμός που υπάρχει στον πίνακα ΜΑΘΗΜΑ αφορά το εξάμηνο στο οποίο διδάσκεται το εκάστοτε μάθημα, το οποίο θα πρέπει να είναι μεγαλύτερο ή ίσο του ένα.



CONSTRAINT [CK_Subjects_Semester] **CHECK** ((([Semester]>=(1))))

Μερικές εγγραφές του πίνακα αυτού έχουν συμπληρωθεί, όπως φαίνεται και παρακάτω.

	Id	Title	Semester	Splitted	Lab
▶	1	Γραμμική Άλγεβρα	1	True	False
	2	Λογισμός Ι	1	True	False
	3	Συστήματα Υπολογιστών	1	True	False
	4	Τεχνικές Σχεδίασης με χρήση Η/Υ	1	False	False
	5	Τεχνική Μηχανική	1	True	False
	6	Φυσική Ι	1	False	False
	7	Διαφορικές Εξισώσεις	2	False	False
	8	Δομημένος Προγραμματισμός	2	False	False
	9	Εφαρμοσμένη Θερμοδυναμική	2	False	False
	10	Ηλεκτρικά Κυκλώματα Ι	2	False	False
	11	Θεωρία Πιθανοτήτων και Στατιστική	2	False	False
	12	Λογισμός ΙΙ	2	False	False
	13	Θεωρία Σημάτων και Γραμμικών Συστημάτων	3	True	False
	14	Εφαρμοσμένα Μαθηματικά Ι	3	False	False
	15	Ηλεκτρικά Κυκλώματα ΙΙ	3	True	False
	16	Ηλεκτρολογικά Υλικά	3	False	False
	17	Ηλεκτρομαγνητικό Πεδίο Ι	3	True	False
	18	Ηλεκτρονική Ι	3	False	False
	19	Προγραμματιστικές Τεχνικές	3	True	False
	20	Αριθμητική Ανάλυση	4	False	False
	21	Αρχιτεκτονική Υπολογιστών	4	False	False
	22	Εισαγωγή στην Ενεργειακή Τεχνολογία Ι	4	False	False
	23	Ηλεκτρικά Κυκλώματα ΙΙΙ	4	False	False
	24	Ηλεκτρομαγνητικό Πεδίο ΙΙ	4	True	False

Σχήμα 22: Παράδειγμα του πίνακα ΜΑΘΗΜΑ

- ✓ Σχέση «ΜΑΘΗΜΑ_ΤΟΜΕΑΣ» η οποία δηλώνει το αν ένα μάθημα είναι υποχρεωτικό, επιλογής ή ελεύθερης επιλογής για κάθε έναν από τους τομείς του τμήματος.

	Column Name	Data Type	Allow Nulls
	Subject_Id	smallint	<input type="checkbox"/>
	Branch_Id	tinyint	<input type="checkbox"/>
	Type	nvarchar(2)	<input type="checkbox"/>

Σχήμα 23: Σχέση ΜΑΘΗΜΑ_ΤΟΜΕΑΣ

Οι περιορισμοί αναφορικής ακεραιότητας του πίνακα αυτού αφορούν τα γνωρίσματα Subject_Id και Branch_Id, όπως ορίζονται και με την SQL παρακάτω.

CONSTRAINT [FK_BranchSubjectTypes_Subject] **FOREIGN KEY**([Subject_Id])**REFERENCES** [dbo].[Subjects] ([Id])

CONSTRAINT [FK_BranchSubjectTypes_Branches] **FOREIGN KEY**([Branch_Id])**REFERENCES** [dbo].[Branches] ([Id])

Επίσης, ορίζεται και ο περιορισμός για το γνώρισμα Type, το οποίο πρέπει να παίρνει μόνο τις τιμές Y, E ή EE.


CONSTRAINT [CK_BranchSubjectTypes_Type] **CHECK** (([Type]=N'Y' OR [Type]=N'E' OR [Type]=N'EE'))

Στη συνέχεια, δίνεται ο πίνακας ΜΑΘΗΜΑ_ΤΟΜΕΑΣ συμπληρωμένος με κάποιες εγγραφές.

	Subject_Id	Branch_Id	Type
	95	1	Y
	107	1	Y
	51	1	Y
	73	1	Y
	62	1	Y
	57	3	E
	67	1	E
	43	1	E
	43	2	Y
	43	3	E
	46	3	Y
	54	3	Y
	47	2	Y
	130	2	E

Σχήμα 24: Παράδειγμα της σχέσης ΜΑΘΗΜΑ_ΤΟΜΕΑΣ

- ✓ Πίνακας «ΕΞΕΤΑΣΤΙΚΗ» που δίνει id για κάθε συνδυασμό έτους και περιόδου από την αρχή της ίδρυσης του τμήματος.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Year	smallint	<input type="checkbox"/>
	Period	nchar(1)	<input type="checkbox"/>

Σχήμα 25: Πίνακας ΕΞΕΤΑΣΤΙΚΗ

Όσον αφορά τους περιορισμούς, αυτοί είναι δύο και αφορούν, αφενός, το γνώρισμα Year που θα πρέπει να είναι μεγαλύτερο ή ίσο του έτους ίδρυσης του τμήματος.

CONSTRAINT [CK_Exams_Year] **CHECK** (([Year]>=(1972)))

Αφετέρου, το γνώρισμα Period θα πρέπει να λαμβάνει τις τιμές Χ (Χειμερινό), Ε (Εαρινό) ή Σ (Σεπτέμβριος).


CONSTRAINT [CK_Exams_Period] **CHECK** (([Period]=N'Χ' OR [Period]=N'Ε' OR [Period]=N'Σ'))

Μερικά παραδείγματα του πίνακα ΕΞΕΤΑΣΤΙΚΗ φαίνονται στην επόμενη σελίδα.

	Id	Year	Period
▶	2	2007	Ε
	5	2008	Ε
	8	2009	Ε
	11	2010	Ε
	14	2011	Ε
	17	2012	Ε
	20	2013	Ε
	23	2014	Ε
	3	2007	Σ
	6	2008	Σ
	9	2009	Σ
	12	2010	Σ
	15	2011	Σ
	18	2012	Σ

Σχήμα 26: Παράδειγμα του πίνακα ΕΞΕΤΑΣΤΙΚΗ

- ✓ Πίνακας «ΔΙΔΑΣΚΑΛΙΑ» ο οποίος αποθηκεύει το τμήμα (ή τα τμήματα) για κάθε μάθημα και για κάθε έτος.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Year	smallint	<input type="checkbox"/>
	Class	nchar(3)	<input type="checkbox"/>
	Subject_id	smallint	<input type="checkbox"/>

Σχήμα 27: Πίνακας ΔΙΔΑΣΚΑΛΙΑ


Ο μοναδικός περιορισμός ακεραιότητας αφορά το γνώρισμα Subject_id και δηλώνεται στη συνέχεια με τη χρήση SQL.

```
CONSTRAINT [FK_Teachings_Subject] FOREIGN KEY([Subject_id])REFERENCES
[dbo].[Subjects] ([Id])
```

Επίσης, υπάρχει και ο περιορισμός το γνώρισμα Class να παίρνει τιμές A-X, X-Ω ή A-Ω, όπου το X δηλώνει το γράμμα βάσει του οποίου χωρίζεται το τμήμα στα δύο, πχ το K ή το Λ.



```
CONSTRAINT [CK_Teachings_Class] CHECK (([Class]=N'A-X' OR [Class]=N'X-Ω' OR
[Class]=N'A-Ω'))
```

Τέλος, παραθέτουμε συμπληρωμένο τον πίνακα ΔΙΔΑΣΚΑΛΙΑ.

	Id	Year	Class	Subject_id
	1	2007	A-X	1
	141	2008	A-X	1
	281	2009	A-X	1
	421	2010	A-X	1
	561	2011	A-X	1
	701	2012	A-X	1
	841	2013	A-X	1
	981	2014	A-X	1
	3	2007	A-X	2
	143	2008	A-X	2
	283	2009	A-X	2
	423	2010	A-X	2
	563	2011	A-X	2
	703	2012	A-X	2
	843	2013	A-X	2
	983	2014	A-X	2
	5	2007	A-X	3
	145	2008	A-X	3

Σχήμα 28: Παράδειγμα του πίνακα ΔΙΔΑΣΚΑΛΙΑ

- ✓ Σχέση «ΚΑΘΗΓΗΤΗΣ_ΔΙΔΑΣΚΑΛΙΑ» η οποία παρέχει την πληροφορία σε ποια τμήματα διδάσκει ένας καθηγητής για κάθε έτος και για τα μαθήματα που τον αφορούν.

	Column Name	Data Type	Allow Nulls
	Professor_Id	smallint	<input type="checkbox"/>
	Teaching_Id	int	<input type="checkbox"/>

Σχήμα 29: Σχέση ΚΑΘΗΓΗΤΗΣ_ΔΙΔΑΣΚΑΛΙΑ

Και για τα δύο γνωρίσματα του πίνακα υπάρχουν περιορισμοί ακεραιότητας και αναγράφονται στη συνέχεια.

CONSTRAINT [FK_ProfessorGroups_Professors] **FOREIGN KEY**([Professor_Id])**REFERENCES** [dbo].[Professors] ([Id])



CONSTRAINT [FK_ProfessorGroups_Teachings] **FOREIGN KEY**([Teaching_Id])**REFERENCES** [dbo].[Teachings] ([Id])

Δεν υπάρχουν κάποιοι άλλοι περιορισμοί που θα πρέπει να οριστούν βάσει λογικής. Στην επόμενη σελίδα, φαίνεται η σχέση ΚΑΘΗΓΗΤΗΣ_ΔΙΔΑΣΚΑΛΙΑ συμπληρωμένη με μερικά δεδομένα.

	Professor_Id	Teaching_Id
	61	314
	61	174
	61	34
	61	454
	61	594
	61	734
	61	874
	61	1014
	4	116
	4	676
	4	37
	4	177
	2	894
	2	1034
	14	250
	14	390
	6	351
	17	113
	17	253

Σχήμα 30: Παράδειγμα της σχέσης ΚΑΘΗΓΗΤΗΣ_ΔΙΔΑΣΚΑΛΙΑ

- ✓ Σχέση «ΕΞΕΤΑΣΗ_ΜΑΘΗΜΑΤΟΣ» η οποία αποθηκεύει τον αριθμό των εγγεγραμμένων φοιτητών και των ημερών που χρειάστηκαν για την ανακοίνωση των αποτελεσμάτων για κάποιο μάθημα σε κάποια εξεταστική.

	Column Name	Data Type	Allow Nulls
	Teachings_Id	int	<input type="checkbox"/>
	Exams_Id	int	<input type="checkbox"/>
	Subscribed	int	<input checked="" type="checkbox"/>
	Result_Days	tinyint	<input checked="" type="checkbox"/>

Σχήμα 31: Σχέση ΕΞΕΤΑΣΗ_ΜΑΘΗΜΑΤΟΣ

Οι περιορισμοί αναφορικής ακεραιότητας αυτής της σχέσης αφορούν τα γνωρίσματα του πρωτεύοντος κλειδιού και αναφέρονται με SQL παρακάτω.

```
CONSTRAINT [FK_SubjectExams_Teachings] FOREIGN KEY([Teachings_Id])
REFERENCES [dbo].[Teachings] ([Id])
CONSTRAINT [FK_SubjectExams_Exams] FOREIGN KEY([Exams_Id])
REFERENCES [dbo].[Exams] ([Id])
```

Τίθενται, επίσης, και κάποια γενικότεροι περιορισμοί. Ο πρώτος αφορά τις ημέρες αποτελεσμάτων που θα πρέπει να είναι μη αρνητικός αριθμός.

```
CONSTRAINT [CK_SubjectExams_Result_Days] CHECK (([Result_Days]>=(0)))
```

Ο δεύτερος περιορισμός ορίζει ότι και οι εγγεγραμμένοι σε ένα μάθημα πρέπει να είναι μη αρνητικός αριθμός.

```
CONSTRAINT [CK_SubjectExams_Subscribed] CHECK (([Subscribed]>=(0)))
```

Ακόμη, παραθέτουμε τη συμπληρωμένη με εγγραφές σχέση ΕΞΕΤΑΣΤΙΚΗ_ΜΑΘΗΜΑΤΟΣ.

	Teachings_Id	Exams_Id	Subscribed	Result_Days
▶	1	1	259	12
	1	3	259	12
	2	1	259	12
	2	3	259	12
	3	1	225	10
	3	3	225	10
	4	1	225	10
	4	3	225	10
	5	1	255	14
	5	3	255	14
	6	1	255	14
	6	3	255	14
	7	1	228	11
	7	3	228	11
	8	1	272	16
	8	3	272	16

Σχήμα 32: Παράδειγμα της σχέσης ΕΞΕΤΑΣΗ_ΜΑΘΗΜΑΤΟΣ

- ✓ Πίνακας «ΑΙΘΟΥΣΑ» ο οποίος δηλώνει τη χωρητικότητα και το κτίριο στο οποίο στεγάζονται οι διαθέσιμες αίθουσες του τμήματος.

	Column Name	Data Type	Allow Nulls
	Building	nvarchar(50)	<input type="checkbox"/>
	Capacity	int	<input type="checkbox"/>
🔑	Name	nvarchar(50)	<input type="checkbox"/>

Σχήμα 33: Πίνακας ΑΙΘΟΥΣΑ

Ο μοναδικός περιορισμός που τίθεται εδώ αφορά τη χωρητικότητα των διαθέσιμων αιθουσών, η οποία προφανώς πρέπει να είναι μεγαλύτερη του μηδενός.


CONSTRAINT [CK_Halls_Capacity] **CHECK** (([Capacity]>=(0)))

Ακολουθεί ο συμπληρωμένος με εγγραφές πίνακας ΑΙΘΟΥΣΑ.

	Building	Capacity	Name
	Δ	100	A1
	Δ	50	A2
	Δ	70	A3
	Δ	50	A4
	Δ	200	A5
	Δ	60	A6
	Δ	50	A7
	Δ	250	Αμφιθέατρο Π...

Σχήμα 34: Παράδειγμα του πίνακα ΑΙΘΟΥΣΑ

- ✓ Πίνακας «ΣΥΓΓΡΑΜΜΑ» που αποθηκεύει το ISBN και τον τίτλο ενός συγγράμματος.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Title	nvarchar(50)	<input type="checkbox"/>
	ISBN	varchar(13)	<input type="checkbox"/>

Σχήμα 35: Πίνακας ΣΥΓΓΡΑΜΜΑ

Για το συγκεκριμένο πίνακα, δεν υπάρχει κάποιος περιορισμός που πρέπει αν ικανοποιείται. Παρακάτω, δίνεται ο συμπληρωμένος πίνακας ΣΥΓΓΡΑΜΜΑ.

	Id	Title	ISBN
	17	Αναγνώριση Προτύπων	9789604891450
	5	Ανάλυση Ηλεκτρικών Κυκλ...	9789604182022
	3	Αντικειμενοστραφής Ανάπτ...	9602099135
	22	Βασικές αρχές για τα συστήμ...	9789604611836
▶	24	Δίκτυα Υπολογιστών	9789604614479
	7	Επεξεργασία Αναλογικών Ση...	9789601201573
	6	Ηλεκτρομαγνητικό Πεδίο	9789605243241
	15	Λειτουργικά Συστήματα	9789609184811
	11	Οργάνωση και Σχεδίαση Υπο...	9789604613526
	16	Προγραμματίζοντας τον μικ...	9789608050518
	12	Συστήματα Αυτομάτου Ελέγ...	9609160611
	18	Συστήματα Βάσεων Δεδομέν...	9608105870
	21	Συστήματα Βάσεων Δεδομέν...	9789605126230
	20	Συστήματα Διαχείρισης Βάσε...	9789604184118
	14	Συστήματα Επικοινωνίας	9789607182685
	23	Σχεδίαση Ολοκληρωμένων Κ...	9789607182678
	4	Τεχνολογία Λογισμικού	9789603517832
	8	Ψηφιακή Σχεδίαση: Αρχές & ...	9602097280

Σχήμα 36: Παράδειγμα του πίνακα ΣΥΓΓΡΑΜΜΑ

- ✓ Σχέση «ΔΗΛΩΣΗ_ΣΥΓΓΡΑΜΜΑΤΟΣ» που περιλαμβάνει τη δήλωση ενός φοιτητή για επιλογή ενός συγγράμματος στα μαθήματα που δήλωσε.

	Column Name	Data Type	Allow Nulls
🔑	Student_Id	int	<input type="checkbox"/>
🔑	Subject_id	smallint	<input type="checkbox"/>
🔑	Book_id	int	<input type="checkbox"/>
	Date	date	<input checked="" type="checkbox"/>

Σχήμα 37: Σχέση ΔΗΛΩΣΗ_ΣΥΓΓΡΑΜΜΑΤΟΣ

Ο περιορισμός αναφορικής ακεραιότητας αναφέρεται στα γνωρίσματα του πρωτεύοντος κλειδιού και δίνεται παρακάτω.

```

CONSTRAINT [FK_BookApplications_Students] FOREIGN KEY([Student_Id])REFERENCES
[dbo].[Students] ([Id])
CONSTRAINT [FK_BookApplications_Subjects] FOREIGN KEY([Subject_id])REFERENCES
[dbo].[Subjects] ([Id])
CONSTRAINT [FK_BookApplications_Books] FOREIGN KEY([Book_id])REFERENCES
[dbo].[Books] ([Id])

```

Παρακάτω, φαίνονται μερικές εγγραφές του πίνακα ΔΗΛΩΣΗ_ΣΥΓΓΡΑΜΜΑΤΟΣ.

	Student_Id	Subject_id	Book_id	Date
	7414	17	17	2014-10-21
	7420	107	18	2015-01-25
	7398	73	16	2013-12-18
	7209	10	5	2009-04-12
	7404	95	3	2014-05-13
	7603	13	7	2012-11-05
	7634	17	6	2012-01-10
	7500	62	15	2012-12-12
	7345	21	11	2010-03-14
	7444	32	12	2013-11-18
	7103	95	4	2008-04-12
	7333	25	8	2012-05-05

Σχήμα 38: Παράδειγμα της σχέσης ΔΗΛΩΣΗ_ΣΥΓΓΡΑΜΜΑΤΟΣ

- ✓ Σχέση «ΒΑΘΜΟΛΟΓΙΑ» που αποθηκεύει τις βαθμολογίες των φοιτητών του τμήματος για κάθε μάθημα και κάθε εξεταστική.

	Column Name	Data Type	Allow Nulls
🔑	Student_Id	int	<input type="checkbox"/>
🔑	Teachings_Id	int	<input type="checkbox"/>
🔑	Period	nchar(1)	<input type="checkbox"/>
	Grade	real	<input type="checkbox"/>

Σχήμα 39: Σχέση ΒΑΘΜΟΛΟΓΙΑ

Ο περιορισμός αναφορικής ακεραιότητας αναφέρεται στα γνωρίσματα Student_Id και Teachings_Id της σχέσης ΒΑΘΜΟΛΟΓΙΑ και δηλώνεται παρακάτω με τη χρήση SQL.

```
CONSTRAINT [FK_Grades_Students] FOREIGN KEY([Student_Id])REFERENCES
[dbo].[Students] ([Id])
CONSTRAINT [FK_Grades_Teachings] FOREIGN KEY([Teachings_Id])REFERENCES
[dbo].[Teachings] ([Id])
```

Ένας ακόμη περιορισμός που τίθεται αφορά το γνώρισμα Period, το οποίο θα πρέπει να παίρνει μόνο τις τιμές X, E ή Σ.

```
CONSTRAINT [CK_Grades] CHECK (([Period]=N'X' OR [Period]=N'E' OR [Period]=N'Σ'))
```

Επίσης, το γνώρισμα Grade θα πρέπει να βρίσκεται στο διάστημα [0,10].

CONSTRAINT [CK_Grades_Grade] **CHECK** (([Grade]>=(0) AND [Grade]<=(10)))

Τέλος, δίνεται συμπληρωμένος ο πίνακας ΒΑΘΜΟΛΟΓΙΑ με κάποιες ενδεικτικές τιμές.

	Student_Id	Teachings_Id	Period	Grade
▶	6600	21	X	2
	6600	26	Σ	2
	6600	27	Σ	3
	6600	144	X	2
	6600	147	Σ	4
	6600	147	X	4
	6600	170	Σ	3
	6600	177	Σ	1
	6600	282	X	1
	6600	291	E	1
	6600	292	Σ	1
	6600	307	Σ	8
	6600	309	Σ	2
	6600	316	Σ	1
	6600	317	Σ	3
	6600	318	X	6

Σχήμα 40: Παράδειγμα της σχέσης ΒΑΘΜΟΛΟΓΙΑ

- ✓ Σχέση «ΔΙΑΛΕΞΗ» η οποία περιέχει την απαραίτητη πληροφορία σχετικά με το πότε και σε ποια αίθουσα διδάσκεται το εκάστοτε μάθημα.

	Column Name	Data Type	Allow Nulls
🔑	Hall_Id	nvarchar(50)	<input type="checkbox"/>
🔑	Subject_Id	smallint	<input type="checkbox"/>
	Day	nchar(2)	<input type="checkbox"/>
	Start_Hour	tinyint	<input type="checkbox"/>
	Duration	tinyint	<input type="checkbox"/>

Σχήμα 41: Σχέση ΔΙΑΛΕΞΗ

Ο περιορισμός αναφορικής ακεραιότητας αφορά τα γνώρισμα του πρωτεύοντος κλειδιού της σχέσης ΔΙΑΛΕΞΗ και δίνεται παρακάτω.

CONSTRAINT [FK_Classes_Halls] **FOREIGN KEY**([Hall_Id])**REFERENCES** [dbo].[Halls] ([Name])

CONSTRAINT [FK_Classes_Subject] **FOREIGN KEY**([Subject_Id])**REFERENCES** [dbo].[Subjects] ([Id])

Ένας ακόμη περιορισμός αναφέρεται στο ότι το γνώρισμα Day μπορεί να παίρνει μόνο τις τιμές ΔΕ, ΤΡ, ΤΕ, ΠΕ ή ΠΑ, όπως ορίζεται και στη συνέχεια.

```
CONSTRAINT [CK_Classes_Day] CHECK (([Day]=N'ΔΕ' OR [Day]=N'ΤΡ' OR [Day]=N'ΤΕ' OR [Day]=N'ΠΕ' OR [Day]=N'ΠΑ'))
```

Επιπλέον, η ώρα έναρξης κάποιου μαθήματος πρέπει να βρίσκεται στο διάστημα [8,20].

```
CONSTRAINT [CK_Classes_Start_Hour] CHECK (([Start_Hour]>=(8) AND [Start_Hour]<=(20)))
```

Παρακάτω, φαίνεται η σχέση ΔΙΑΛΕΞΗ με συμπληρωμένες κάποιες εγγραφές.

	Hall_Id	Subject_Id	Day	Start_Hour	Duration
	A1	17	ΤΕ	9	2
	A5	32	ΔΕ	17	2
	A1	1	ΔΕ	18	2
	A1	6	ΤΡ	12	2
	A7	95	ΠΕ	9	2
	A7	107	ΠΑ	9	3
	A1	26	ΤΡ	15	2
	A2	51	ΤΕ	16	2
	A7	73	ΔΕ	9	2
	A6	62	ΠΕ	14	2
	Αμφιθέατρο Π...	21	ΠΑ	12	2
	A1	2	ΔΕ	11	2
	A5	12	ΤΡ	9	3
	A1	8	ΠΕ	17	2
	A5	10	ΠΑ	9	2

Σχήμα 42: Παράδειγμα της σχέσης ΔΙΑΛΕΞΗ

Ρόλοι Χρηστών

Τα δεδομένα που διατηρεί η βάση HMMYStat αφορούν τόσο την οργάνωση και τη διδασκαλία των μαθημάτων του τμήματος, όσο και πληροφορίες σχετικά με τη σταδιοδρομία και τις δραστηριότητες των φοιτητών. Λόγω της παρουσίας αυτής της ποικιλομορφίας των δεδομένων, κρίθηκε σκόπιμη η υλοποίηση κάποιων ομάδων χρηστών, οι οποίες κατά τη χρήση της βάσης θα έχουν γενικά διαφορετικά δικαιώματα πάνω στους πίνακες του σχήματος. Οι ομάδες που ορίστηκαν είναι τυπικές και έχουν ως σκοπό την κάλυψη της βασικής λειτουργίας της βάσης, ενώ δύνανται να επεκταθούν ή και να πληθύνουν εάν εμφανιστεί ανάγκη. Αυτές είναι οι εξής:

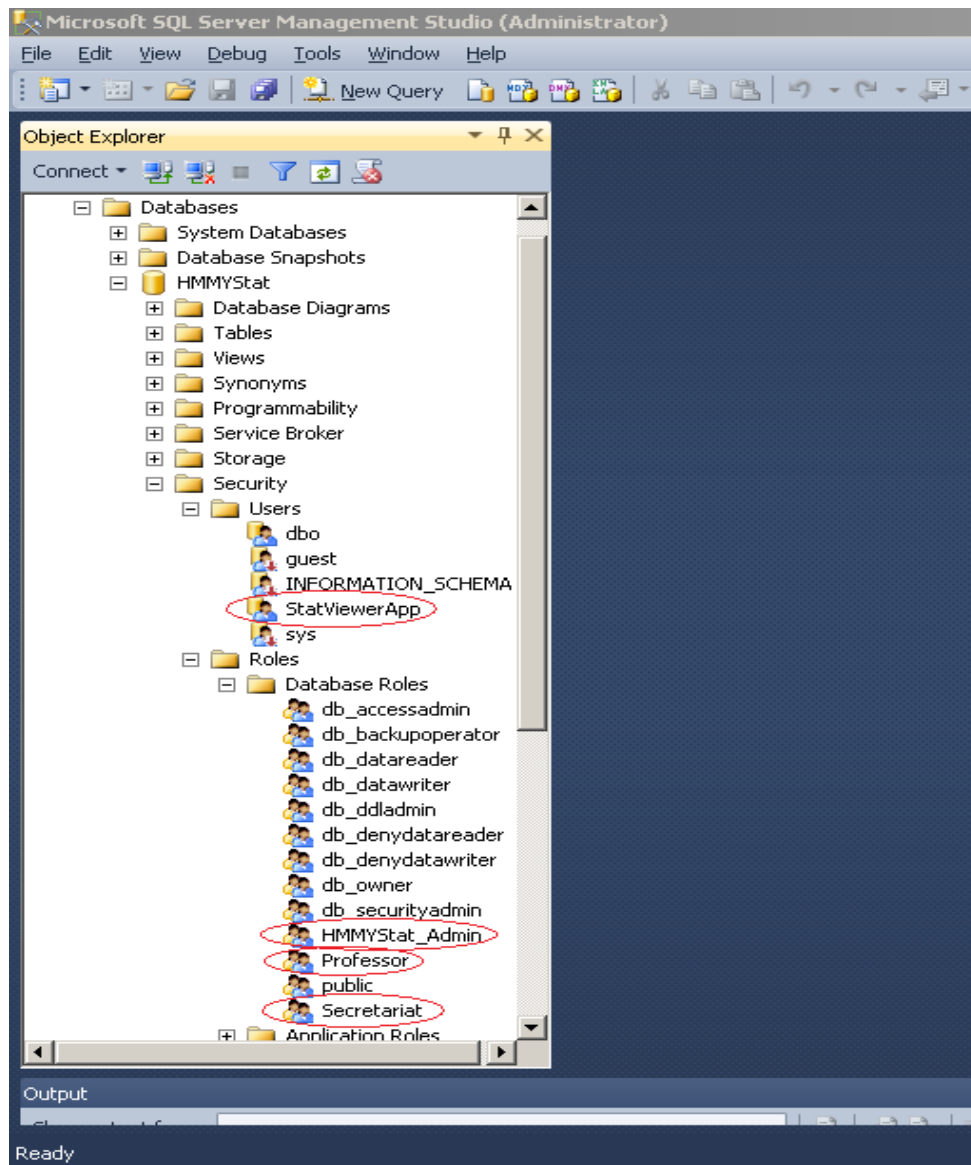
- HMMYStatAdmin: Σε αυτή την κατηγορία ανήκει ο διαχειριστής των δεδομένων που εισάγονται και αποθηκεύονται στη βάση. Έχει πλήρη δικαιώματα επιλογής, εισαγωγής, ενημέρωσης και διαγραφής πάνω σε όλους τους πίνακες του σχήματος και διαθέτει τη δυνατότητα μετάδοσης των δικαιωμάτων αυτών σε άλλους χρήστες. Είναι υπεύθυνος για την παρακολούθηση και τη ρύθμιση των ενεργειών των άλλων χρηστών της βάσης. Ο καθορισμός αυτών των δικαιωμάτων ορίστηκε ως ρόλος, εντός της βάσης, καθώς είναι αναμενόμενη η ύπαρξη περισσότερων του ενός διαχειριστών.
- Secretariat: Σε αυτή την κατηγορία ανήκει ένας χρήστης που εκτελεί χρέη γραμματείας για τη βάση, δηλαδή διαθέτει πλήρη δικαιώματα στο σύνολο των πινάκων της βάσης. Ωστόσο, δεν διαθέτει την ικανότητα μετάδοσης των δικαιωμάτων του, εφόσον αυτό εμπίπτει στα καθήκοντα του διαχειριστή. Είναι υπεύθυνος για την εισαγωγή του κύριου όγκου των δεδομένων που αποθηκεύονται στη βάση, γι' αυτό και κρίθηκε απαραίτητο να οριστεί ως ρόλος, καθώς θα υπάρχουν περισσότεροι από ένας χρήστες επιφορτισμένοι με αυτό το καθήκον.
- Professor: Σε αυτή την κατηγορία ανήκει ένας χρήστης ο οποίος διδάσκει στο τμήμα. Είναι ρόλος παρόμοιος με τη γραμματεία. Έχει δικαιώματα εισαγωγής, ενημέρωσης και διαγραφής για πίνακες που αφορούν έναν καθηγητή (Π.χ. ΒΑΘΜΟΛΟΓΙΑ, ΚΑΘΗΓΗΤΗΣ, ΔΙΔΑΣΚΑΛΙΑ). Έτσι το γέμισμα της βάσης μπορεί να γίνει και από αυτή την ομάδα ανθρώπων. Κρίθηκε ορθό, να οριστεί ως ρόλος εντός της βάσης, καθώς στο τμήμα διδάσκουν περισσότεροι από ένας καθηγητές οι οποίοι θα χρησιμοποιούν ξεχωριστούς χρήστες της βάσης.

Γενικά, θεωρήθηκε πως είναι χρήσιμο οι χρήστες που έχουν τη δυνατότητα εισαγωγής και αλλοίωσης των δεδομένων να έχουν ατομικούς λογαριασμούς σύνδεσης στη βάση, ώστε να είναι δυνατή η παρακολούθηση της δραστηριότητας τους. Γενικά, η σχεδίαση της βάσης έγινε θεωρώντας πως αυτοί οι χρήστες διακρίνονται από σοβαρότητα και σύνεση, παρ' όλα αυτά η δυνατότητα εύρεσης του χρήστη ο οποίος έχει κατ' επανάληψη κάνει κατάχρηση των δικαιωμάτων του είναι ζωτικής σημασίας για την εγκυρότητα των διατηρούμενων δεδομένων.

- StatViewerApp: Είναι ο χρήστης που θα μπορούσε χρησιμοποιεί μία ενδεχόμενη εφαρμογή προβολής στατιστικών. Έχει δικαίωμα προσπέλασης όλων των πινάκων του σχήματος της βάσης με εξαίρεση τα χαρακτηριστικά που περιλαμβάνουν το ID των φοιτητών. Αυτό έγινε για την προστασία των δεδομένων των φοιτητών, σε περίπτωση που κλαπεί ο λογαριασμός του προγραμματιστή της εφαρμογής. Τέλος, θεωρήθηκε σκόπιμο να δημιουργηθεί ένας χρήστης,

αντί ενός ρόλου, καθώς δεν είναι μεγάλη η πιθανότητα δημιουργίας περισσότερων της μιας εφαρμογών. Ταυτόχρονα η επέκταση των δικαιωμάτων του χρήστη, ή η δημιουργία νέου, είναι δυνατή όταν παρουσιαστεί τέτοια ανάγκη.

Παρακάτω, εμφανίζονται οι ρόλοι χρηστών όπως δημιουργήθηκαν από το περιβάλλον του Microsoft SQL Server.



Σχήμα 43: Ρόλοι χρηστών

Για λόγους ευκολότερης αναφοράς, επισημαίνονται τα δικαιώματα πρόσβασης κάθε χρήστη για κάθε πίνακα, όπως αυτά ορίστηκαν από την πρώτη εργασία :

Πίνακας Χρήστης	Students	GraduateStudents	Branches	BranchesStudents
HMMYStatAdmin	SIUD	SIUD	SIUD	SIUD
Secretariat	SIUD	SIUD	SIUD	SIUD
Professor	S	S	S	S
StatViewerApp	S*	S*	S	S*

Πίνακας Χρήστης	Internships	InternshipStudents	Erasmus	ErasmusStudents
HMMYStatAdmin	SIUD	SIUD	SIUD	SIUD
Secretariat	SIUD	SIUD	SIUD	SIUD
Professor	SIU	S	SIU	S
StatViewerApp	S	S*	S	S*

Πίνακας Χρήστης	Professors	Subjects	BranchSubjectTypes	Exams
HMMYStatAdmin	SIUD	SIUD	SIUD	SIUD
Secretariat	SIUD	SIUD	SIUD	SIUD
Professor	S	S	S	S
StatViewerApp	S	S	S	S

Πίνακας Χρήστης	Teachings	ProfessorGroups	SubjectExams	Halls
HMMYStatAdmin	SIUD	SIUD	SIUD	SIUD
Secretariat	SIUD	SIUD	SIUD	SIUD
Professor	SIUD	SIUD	SIUD	S
StatViewerApp	S	S	S	S

Πίνακας Χρήστης	Books	BookApplications	Grades	Classes
HMMYStatAdmin	SIUD	SIUD	SIUD	SIUD
Secretariat	SIUD	SIUD	SIUD	SIUD
Professor	SIU	S	SIUD	SIUD
StatViewerApp	S	S	S*	S

Όψεις

Σε αυτήν την ενότητα, θα παραθέσουμε τις δύο όψεις που υλοποιήθηκαν κατά το σχεδιασμό της βάσης HMMYStat στην πρώτη φάση της εργασίας. Εδώ όμως θα γίνει αναφορά στον SQL κώδικα, αντί της σχεσιακής άλγεβρας που χρησιμοποιήθηκε στην πρώτη εργασία, ενώ θα παρουσιαστούν και τα σχετικά αποτελέσματα. Οι όψεις που θα δημιουργηθούν λέγονται Year_Statistics και Exams_Subjects. Οι όψεις αυτές απαιτούσαν πολλά ενδιάμεσα βήματα, για το οποία δημιουργήθηκαν επί μέρους όψεις. Οι επί μέρους όψεις ενώθηκαν όλες μαζί ώστε να δημιουργηθεί η τελική. Παρακάτω παραθέτουμε όλη την διαδικασία.

Όψη Year_Statistics

- **Students_YearIn**

```
CREATE VIEW [dbo].[Students_YearIn]
AS
SELECT      Id, YEAR(Date_In) AS Year_In, Semester, Moved
FROM        dbo.Students
```

Η όψη αυτή απομονώνει το έτος εισαγωγής των φοιτητών του τμήματος από το γνώρισμα Date_In του πίνακα Students, το μετονομάζει σε Year_In και εμφανίζει τις στήλες Id, Year_In, Semester και Moved. Μερικά από τα αποτελέσματα που εμφανίζονται στην εν λόγω όψη φαίνονται στη συνέχεια.

	Id	Year_In	Semester	Moved
	6600	2007	15	False
	6601	2007	15	False
	6602	2007	10	False
	6603	2007	15	False
	6604	2007	10	False
	6605	2007	10	False
	6606	2007	10	False
	6607	2007	15	False
	6608	2007	10	False
	6609	2007	10	False
	6610	2007	15	False
	6611	2007	10	False
	6612	2007	10	False
	6613	2007	10	False
	6614	2007	10	False
	6615	2007	15	False
	6616	2007	10	False

Σχήμα 44: Αποτελέσματα όψης Students_YearIn

- **Year_Entrants**

```
CREATE VIEW [dbo].[Year_Entrants]
AS
SELECT      Year_In AS Year, COUNT(*) AS Entrants
FROM        dbo.Students_YearIn
GROUP BY Year_In
```

Η Year_Entrants χρησιμοποιεί την όψη Students_YearIn που δημιουργήθηκε προηγουμένως, ώστε να υπολογίσει τον αριθμό των εισακτέων για κάθε έτος. Το Year_In μετονομάζεται σε Year, το άθροισμα των εισακτέων κάθε έτους ονομάζεται Entrants και εμφανίζονται αυτά τα δύο γνωρίσματα.

	Year	Entrants
	2013	200
	2010	500
	2007	200
	2008	200
	2014	200
	2011	200
	2012	200
	2009	200

Σχήμα 45: Αποτελέσματα όψης Year_Entrants

- **Year_Graduates**

```
CREATE VIEW [dbo].[Year_Graduates]
AS
SELECT      Grad_Year AS Year, COUNT(*) AS Graduates
FROM        (SELECT      YEAR(Date) AS Grad_Year
              FROM        dbo.GraduateStudents) AS derivedtbl_1
GROUP BY Grad_Year
```

Η όψη αυτή απομονώνει το έτος από το γνώρισμα Date του πίνακα GraduateStudents και το μετονομάζει σε Grad_Year. Στη συνέχεια, μετράει τον αριθμό όλων των απόφοιτων για κάθε έτος και τα ομαδοποιεί με βάση το έτος. Τα γνωρίσματα που εμφανίζονται από την όψη ονομάζονται Year και Graduates.

	Year	Graduates
►	2012	70
	2013	68
	2014	296

Σχήμα 46: Αποτελέσματα όψης Year_Graduates

- **Year_internships**

```
CREATE VIEW [dbo].[Year_Internships]
AS
SELECT      Year, COUNT(*) AS Internships_This_Year
FROM        dbo.InternshipStudents
GROUP BY Year
```

Για τη δημιουργία αυτής της όψης, χρησιμοποιήθηκε ο πίνακας InternshipStudents από τον οποίο καταμετρήθηκε ο αριθμός των φοιτητών που εκπόνησαν πρακτική για κάθε έτος. Τα γνωρίσματα που εμφανίζει η όψη ονομάζονται Year και Internship_This_Year.

	Year	Internships_Th...
	2012	1
	2013	2
	2014	3
	2015	4

Σχήμα 47: Αποτελέσματα όψης Year_Internships

- **Year_Transferred**

```
CREATE VIEW [dbo].[Year_Transferred]
AS
SELECT      Year_In AS Year, COUNT(*) AS Transferred_Students
FROM        dbo.Students_YearIn
WHERE       (Moved = 1)
GROUP BY   Year_In
```

Η όψη αυτή χρησιμοποιεί τα δεδομένα της όψης Students_YearIn. Συγκεκριμένα, καταμετρούνται αυτές τις πλειάδες για τις οποίες ισχύει ότι Moved=1, δηλαδή ότι ο φοιτητής έχει μετεγγραφεί, και ομαδοποιούνται ως προς το έτος. Τα αποτελέσματα φαίνονται στην παρακάτω εικόνα.

	Year	Transferred_St...
	2007	16
	2008	21
	2009	20
	2010	25
	2011	18
	2012	15
	2013	16
	2014	21

Σχήμα 48: Αποτελέσματα όψης Year_Transferred

- **Year_Erasmus**

```
CREATE VIEW [dbo].[Year_Erasmus]
AS
SELECT      Year, COUNT(*) AS Erasmus_This_Year
FROM        dbo.ErasmusStudents
GROUP BY   Year
```

Για την εξαγωγή των ζητούμενων αποτελεσμάτων αυτής της όψης, έγινε χρήση του πίνακα ErasmusStudents, από όπου καταμετρήθηκαν όλοι οι φοιτητές που συμμετείχαν στο πρόγραμμα Erasmus για κάθε έτος. Τα αποτελέσματα φαίνονται παρακάτω.

	Year	Erasmus_This_...
	2010	2
	2011	2
	2012	2
	2013	3
	2014	1

Σχήμα 49: Αποτελέσματα όψης Year_Erasmus

- **Year_Picked_Electronics**

```
CREATE VIEW [dbo].[Year_Picked_Electronics]
AS
SELECT      Year, COUNT(*) AS Picked_Electronics
FROM        dbo.BranchStudents
WHERE       (Branch_Id = 1)
GROUP BY Year
```

Για να εμφανιστεί ο αριθμός των ατόμων που διάλεξαν τομέα Ηλεκτρονικής, από τον πίνακα BranchStudents καταμετρώνται οι πλειάδες που έχουν Branch_Id=1 και ομαδοποιούνται ως προς το έτος. Υπενθυμίζεται ότι έχει γίνει η κάτωθι αντιστοίχιση τομέων και id.

Id	Τομείς
1	Ηλεκτρονική
2	Ενέργειας
3	Τηλεπικοινωνιών

Τα αποτελέσματα αυτής της όψης φαίνονται παρακάτω.

	Year	Picked_Electro...
	2010	31
	2011	60
	2012	73
	2013	216
	2014	195

Σχήμα 50: Αποτελέσματα όψης Year_Picked_Electronics

- **Year_Picked_Energy**

```
CREATE VIEW [dbo].[Year_Picked_Energy]
AS
SELECT      Year, COUNT(*) AS Picked_Energy
FROM        dbo.BranchStudents
WHERE       (Branch_Id = 2)
GROUP BY Year
```

Η λογική αυτής της όψης είναι ίδια με της Year_Picked_Electronics, μόνο που πρέπει Branch_Id=2. Τα αποτελέσματα φαίνονται στην επόμενη σελίδα.

	Year	Picked_Energy
	2010	23
	2011	34
	2012	39
	2013	124
	2014	108

Σχήμα 51: Αποτελέσματα όψης Year_Picked_Energy

- **Year_Picked_Telecommunications**

```
CREATE VIEW [dbo].[Year_Picked_Telecommunications]
AS
SELECT      Year, COUNT(*) AS Picked_Telecommunications
FROM        dbo.BranchStudents
WHERE       (Branch_Id = 3)
GROUP BY Year
```

Εδώ πρέπει να ισχύει Branch_Id=3, αλλά κατά τα άλλα η λογική είναι ίδια με των προηγούμενων δύο όψεων.

	Year	Picked_Teleco...
	2010	7
	2011	26
	2012	39
	2013	82
	2014	91

Σχήμα 52: Αποτελέσματα όψης Year_Picked_Telecommunications

- **Year_Average_Grade**

```
CREATE VIEW [dbo].[Year_Average_Grade]
AS
SELECT      Year, AVG(Grade) AS Average_Grade
FROM        (SELECT      YEAR(Date) AS Year, Grade
              FROM        dbo.GraduateStudents) AS derivedtbl_1
GROUP BY Year
```

Από τον πίνακα GraduateStudents και το γνώρισμα Date απομονώνεται το έτος αποφοίτησης κάθε φοιτητή και μετονομάζεται σε Year. Στη συνέχεια, υπολογίζεται ο μέσος όρος των βαθμολογιών των αποφοίτων και ομαδοποιείται ως προς το έτος.

	Year	Average_Grade
	2012	7,714285714285...
	2013	8,147058823529...
	2014	7,908783783783...

Σχήμα 53: Αποτελέσματα όψης Year_Average_Grade

Τελική όψη:

- **Year_Statistics**

```
CREATE VIEW [dbo].[Year_Statistics]
AS
SELECT      TOP (100) PERCENT A.Year, A.Entrants, dbo.Year_Graduates.Graduates,
              dbo.Year_Transferred.Transferred_Students,
              dbo.Year_Internships.Internships_This_Year,
              B.Erasmus_This_Year, dbo.Year_Picked_Energy.Picked_Energy,
              dbo.Year_Picked_Electronics.Picked_Electronics,
              dbo.Year_Picked_Telecommunications.Picked_Telecommunications,
              dbo.Year_Average_Grade.Average_Grade
FROM        dbo.Year_Entrants AS A LEFT OUTER JOIN
              dbo.Year_Erasmus AS B ON A.Year = B.Year LEFT OUTER JOIN
              dbo.Year_Average_Grade ON B.Year =
              dbo.Year_Average_Grade.Year LEFT OUTER JOIN
              dbo.Year_Graduates ON dbo.Year_Average_Grade.Year =
              dbo.Year_Graduates.Year LEFT OUTER JOIN
              dbo.Year_Picked_Energy ON dbo.Year_Graduates.Year =
              dbo.Year_Picked_Energy.Year LEFT OUTER JOIN
              dbo.Year_Picked_Electronics ON dbo.Year_Picked_Energy.Year =
              dbo.Year_Picked_Electronics.Year LEFT OUTER JOIN
              dbo.Year_Internships ON dbo.Year_Picked_Electronics.Year =
              dbo.Year_Internships.Year LEFT OUTER JOIN
              dbo.Year_Picked_Telecommunications ON
              dbo.Year_Internships.Year = dbo.Year_Picked_Telecommunications.Year LEFT OUTER
              JOIN
              dbo.Year_Transferred ON
              dbo.Year_Picked_Telecommunications.Year = dbo.Year_Transferred.Year
ORDER BY A.Year DESC
```

Εδώ, ενώνουμε όλες τις προηγούμενες όψεις με LEFT OUTER JOIN ώστε να εμφανιστούν τα στατιστικά δεδομένα για όλα τα έτη, κατά φθίνουσα σειρά, ακόμη και αν κάποια πεδία είναι κενά σε κάποια έτη. Τα αποτελέσματα που θέλουμε να εμφανιστούν είναι τα Year, Entrants, Graduates, Transferred_Students, Internships_This_Year, Erasmus_This_Year, Picked_Energy, Picked_Electronics, Picked_Telecommunications και Average_Grade, όπως φαίνεται και στη συνέχεια.

	Year	Entrants	Graduates	Transferred_St...	Internships_Th...	Erasmus_This_...	Picked_Energy	Picked_Electro...	Picked_Teleco...	Average_Grade
	2013	200	68	16	2	3	124	216	82	8,147058823529...
	2010	500	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL
	2007	200	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	2008	200	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	2014	200	296	21	3	1	108	195	91	7,908783783783...
	2011	200	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL
	2012	200	70	15	1	2	39	73	39	7,714285714285...
	2009	200	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Σχήμα 54: Αποτελέσματα όψης Year_Statistics

Όψη Exams_Subjects

Όπως η όψη με τα στατιστικά έτους, έτσι και αυτή χωρίζεται σε επιμέρους υπο-όψεις.

- **Exams_Basic_Info**

```
SELECT      A.Teachings_Id, B.Subject_id, A.Exams_Id, C.Period
FROM        dbo.SubjectExams AS A INNER JOIN
            dbo.Teachings AS B ON A.Teachings_Id = B.Id INNER JOIN
            dbo.Exams AS C ON A.Exams_Id = C.Id
```

Η όψη αυτή εμφανίζει την περίοδο και τα id διδασκαλίας, μαθήματος και εξεταστικής από τον πίνακα που προκύπτει αν συνενώσουμε τους πίνακες SubjectsExams και Teachings για τα id διδασκαλίας που είναι κοινά και τους πίνακες SubjectExams και Exams για τα id εξεταστικής που είναι κοινά στους επιμέρους πίνακες. Επειδή ο αριθμός των αποτελεσμάτων είναι αρκετά μεγάλος για να χωρέσει σε μία αναφορά, παραθέτουμε στην επόμενη σελίδα μέρος των αποτελεσμάτων που προέκυψαν.

	Teachings_Id	Subject_id	Exams_Id	Period
1	1	1	1	X
2	1	1	3	Σ
3	2	1	1	X
4	2	1	3	Σ
5	3	2	1	X
6	3	2	3	Σ
7	4	2	1	X
8	4	2	3	Σ
9	5	3	1	X
10	5	3	3	Σ
11	6	3	1	X
12	6	3	3	Σ
13	7	4	1	X
14	7	4	3	Σ
15	8	5	1	X
16	8	5	3	Σ
17	9	5	1	X
18	9	5	3	Σ
19	10	6	1	X

Σχήμα 55: Αποτελέσματα όψης Exams-Basic_Info

- **Exams_Taken_Results**

```

SELECT      B.Subject_id, A.Teachings_Id, B.Exams_Id, A.Subscribed,
A.Result_Days
FROM        dbo.SubjectExams AS A INNER JOIN
            dbo.Exams_Basic_Info AS B ON A.Teachings_Id =
B.Teachings_Id AND A.Exams_Id = B.Exams_Id

```

Η εν λόγω όψη εμφανίζει τα id μαθήματος, διδασκαλίας και εξεταστικής, καθώς και τους εγγεγραμμένους και τις ημέρες που πέρασαν για να βγουν αποτελέσματα σε ένα μάθημα κάποια εξεταστική. Για να προκύψει αυτό, συνενώνονται οι πίνακες SubjectExams και Exams_Basic_Info για τα εκείνα τα id διδασκαλίας και εξεταστικής που είναι κοινά στους δύο πίνακες. Μερικά από τα αποτελέσματα φαίνονται στην επόμενη σελίδα.

	Subject_id	Teachings_Id	Exams_Id	Subscribed	Result_Days
1	1	1	1	259	12
2	1	1	3	259	12
3	1	2	1	259	12
4	1	2	3	259	12
5	2	3	1	225	10
6	2	3	3	225	10
7	2	4	1	225	10
8	2	4	3	225	10
9	3	5	1	255	14
10	3	5	3	255	14
11	3	6	1	255	14
12	3	6	3	255	14
13	4	7	1	228	11
14	4	7	3	228	11
15	5	8	1	272	16
16	5	8	3	272	16
17	5	9	1	272	16
18	5	9	3	272	16
19	6	10	1	270	18

Σχήμα 56: Αποτελέσματα όψης Exams_Taken_Results

- **Exams_Passed**

```

SELECT      B.Subject_id, A.Teachings_Id, B.Exams_Id, A.Passed
FROM        (SELECT      Teachings_Id, Period, COUNT(*) AS Passed
              FROM        dbo.Grades
              WHERE        (Grade >= 5)
              GROUP BY    Teachings_Id, Period) AS A INNER JOIN
              dbo.Exams_Basic_Info AS B ON A.Teachings_Id =
B.Teachings_Id AND A.Period = B.Period

```

Η όψη αυτή εμφανίζει τον αριθμό των ατόμων που πέρασαν σε κάθε εξεταστική και για κάθε μάθημα. Για να γίνει αυτό, από τον πίνακα Grades επιλέγονται οι βαθμολογίες που είναι μεγαλύτερες ή ίσες του 5 και ομαδοποιούνται με βάση την περίοδο και το id διδασκαλίας. Ο πίνακας που προκύπτει συνενώνεται με την όψη Exams_Basic_Info για εκείνα τα id διδασκαλίας και για εκείνες τις περιόδους που είναι κοινές στους δύο πίνακες.

	Subject_id	Teachings_Id	Exams_Id	Passed
1	1	1	1	6
2	1	1	3	9
3	1	2	1	12
4	1	2	3	15
5	2	3	1	13
6	2	3	3	6
7	2	4	1	11
8	2	4	3	11
9	3	5	1	14
10	3	5	3	15
11	3	6	1	11

Σχήμα 57: Αποτελέσματα όψης Exams_Passed

- **Exams_Failed**

```

SELECT      B.Subject_id, A.Teachings_Id, B.Exams_Id, A.Failed
FROM        (SELECT      Teachings_Id, Period, COUNT(*) AS Failed
              FROM        dbo.Grades
              WHERE        (Grade < 5)
              GROUP BY    Teachings_Id, Period) AS A INNER JOIN
              dbo.Exams_Basic_Info AS B ON A.Teachings_Id =
B.Teachings_Id AND A.Period = B.Period

```

Η λογική είναι όμοια με αυτή της προηγούμενης όψης, μόνο που εδώ κρατάμε τις εγγραφές για τις οποίες ισχύει ότι Grade<5. Μερικά αποτελέσματα φαίνονται παρακάτω.

	Subject_id	Teachings_Id	Exams_Id	Failed
1	1	1	1	70
2	1	1	3	58
3	1	2	1	56
4	1	2	3	54
5	2	3	1	51
6	2	3	3	57
7	2	4	1	55
8	2	4	3	45
9	3	5	1	56
10	3	5	3	56
11	3	6	1	63

Σχήμα 58: Αποτελέσματα όψης Exams_Failed

- **Exams_Participation**

```
SELECT      B.Subject_id, A.Teachings_Id, B.Exams_Id, A.Participation
FROM        (SELECT      Teachings_Id, Period, COUNT(*) AS Participation
              FROM        dbo.Grades
              GROUP BY    Teachings_Id, Period) AS A INNER JOIN
              dbo.Exams_Basic_Info AS B ON A.Teachings_Id =
B.Teachings_Id AND A.Period = B.Period
```

Η όψη αυτή εμφανίζει τον αριθμό των ατόμων που συμμετείχαν σε κάθε εξεταστική και για κάθε μάθημα. Για να γίνει αυτό, από τον πίνακα Grades καταμετρώνται όλες οι πλειάδες και ομαδοποιούνται ως προς την περίοδο και το id διδασκαλίας. Ο πίνακας που προκύπτει συνενώνεται με την όψη Exams_Basic_Info για εκείνα τα id διδασκαλίας και για εκείνες τις περιόδους που είναι κοινές στους δύο πίνακες.

Μερικά αποτελέσματα αυτής της όψης φαίνονται στην επόμενη σελίδα.

	Subject_id	Teachings_Id	Exams_Id	Participation
1	1	1	1	76
2	1	1	3	67
3	1	2	1	68
4	1	2	3	69
5	2	3	1	64
6	2	3	3	63
7	2	4	1	66
8	2	4	3	56
9	3	5	1	70
10	3	5	3	71
11	3	6	1	74

Σχήμα 59: Αποτελέσματα όψης Exams_Participation

- **Exams_Average_Grade**

```
SELECT      B.Subject_id, B.Teachings_Id, B.Exams_Id, AVG(A.Grade) AS
Average_Grade
FROM        dbo.Grades AS A INNER JOIN
              dbo.Exams_Basic_Info AS B ON A.Teachings_Id =
B.Teachings_Id AND A.Period = B.Period
GROUP BY    B.Exams_Id, B.Subject_id, B.Teachings_Id
```

Η όψη αυτή εμφανίζει το μέσο όρο ομαδοποιημένο ως προς κάθε μάθημα και κάθε εξεταστική. Για να γίνει αυτό, συνενώνουμε τον πίνακα Grades και την όψη Exams_Basic_Info για εκείνα τα id διδασκαλίας και εκείνες τις περιόδους που είναι κοινές στους δύο πίνακες.

	Subject_id	Teachings_Id	Exams_Id	Average_Grade
1	8	292	9	3,34640522875817
2	19	726	18	3,97368421052632
3	7	991	24	5,39457202505219
4	26	1014	24	5,50684931506849
5	26	594	13	3,35294117647059
6	2	4	3	3,44642857142857
7	20	27	3	3,17213114754098
8	11	155	5	3,16666666666667
9	2	703	18	3,24183006535948
10	27	456	10	3,35064935064935
11	24	731	17	3,66666666666667

Σχήμα 60: Αποτελέσματα όψης Exams_Average_Grade

Τελική όψη:

- **Exams_Subjects**

```
SELECT TOP (100) PERCENT H.Title, H.Semester, F.Year, F.Period,
G.Teachings_Id, G.Subscribed, G.Participation, G.Passed, G.Failed,
G.Average_Grade,
G.Result_Days
FROM dbo.Exams AS F INNER JOIN
(SELECT Z.Subject_id, Z.Teachings_Id,
Z.Exams_Id, X.Average_Grade, B.Failed, C.Participation, D.Passed, E.Subscribed,
E.Result_Days
FROM dbo.Exams_Basic_Info AS Z LEFT
OUTER JOIN
dbo.Exams_Average_Grade
AS X ON Z.Teachings_Id = X.Teachings_Id AND Z.Exams_Id = X.Exams_Id LEFT OUTER
JOIN
dbo.Exams_Failed AS B ON
Z.Teachings_Id = B.Teachings_Id AND Z.Exams_Id = B.Exams_Id LEFT OUTER JOIN
dbo.Exams_Participation
AS C ON Z.Teachings_Id = C.Teachings_Id AND Z.Exams_Id = C.Exams_Id LEFT OUTER
JOIN
dbo.Exams_Passed AS D ON
Z.Teachings_Id = D.Teachings_Id AND Z.Exams_Id = D.Exams_Id LEFT OUTER JOIN
dbo.Exams_Taken_Results
AS E ON Z.Teachings_Id = E.Teachings_Id AND Z.Exams_Id = E.Exams_Id) AS G ON F.Id
= G.Exams_Id LEFT OUTER JOIN
dbo.Subjects AS H ON G.Subject_id = H.Id
ORDER BY H.Semester, F.Year, G.Teachings_Id
```

Εδώ, ενώνουμε όλες τις προηγούμενες όψεις με LEFT OUTER JOIN ώστε να εμφανιστούν τα στατιστικά δεδομένα για όλες τις εξεταστικές και τα μαθήματα. Τα γνωρίσματα που θέλουμε να εμφανιστούν είναι τα Title, Semester, Year, Period, Teachings_Id, Subscribed, Participation, Passed, Failed, Average_Grade και Result_Days, όπως φαίνεται και στη συνέχεια.

	Title	Semester	Year	Period	Teachings_Id	Subscribed	Participation	Passed	Failed	Average_Grade	Result_Days
1	Γραμμική Άλγεβρα	1	2007	Σ	1	259	67	9	58	3,11940298507463	12
2	Γραμμική Άλγεβρα	1	2007	X	1	259	76	6	70	2,77631578947368	12
3	Γραμμική Άλγεβρα	1	2007	X	2	259	68	12	56	3,32352941176471	12
4	Γραμμική Άλγεβρα	1	2007	Σ	2	259	69	15	54	3,50724637681159	12
5	Λογισμός Ι	1	2007	X	3	225	64	13	51	3,4375	10
6	Λογισμός Ι	1	2007	Σ	3	225	63	6	57	2,85714285714286	10
7	Λογισμός Ι	1	2007	X	4	225	66	11	55	3,48484848484848	10
8	Λογισμός Ι	1	2007	Σ	4	225	56	11	45	3,44642857142857	10
9	Συστήματα Υπολογιστών	1	2007	Σ	5	255	71	15	56	3,70422535211268	14
10	Συστήματα Υπολογιστών	1	2007	X	5	255	70	14	56	3,57142857142857	14
11	Συστήματα Υπολογιστών	1	2007	X	6	255	74	11	63	3,25675675675676	14

Σχήμα 61: Αποτελέσματα όψης Exams_Subjects

Παραδείγματα Ερωτημάτων

Αντικείμενο αυτής της ενότητας είναι η παράθεση ενδεικτικών ερωτημάτων. Τα ερωτήματα αυτά περιστρέφονται γύρω από την λογική εξαγωγής στατιστικής πληροφορίας. Πολλά από αυτά τα ερωτήματα έχουν ήδη αναλυθεί σε σχεσιακή άλγεβρα στο πρώτο παραδοτέο.

1. Μέσοι όροι μαθημάτων 5^{ου} εξαμήνου για το 2013:

```
SELECT Title, Average_Grade
FROM Exams_Subjects
WHERE Semester = 5
AND Year = 2013
```

	Title	Average_Grade
1	Αναλογικές Τηλεπικοινωνίες	3,79194630872483
2	Διάδοση Ηλεκτρομαγνητικού Κύματος I	3,47794117647059
3	Διάδοση Ηλεκτρομαγνητικού Κύματος I	3,44848484848485
4	Δομές Δεδομένων	3,64262295081967
5	Εισαγωγή στην Ενεργειακή Τεχνολογία II	3,63667820069204
6	Ηλεκτρικές Μετρήσεις I	3,50501672240803
7	Ηλεκτρονική II	3,62365591397849
8	Συστήματα Αυτομάτου Ελέγχου I	3,43617021276596
9	Αναλογικές Τηλεπικοινωνίες	3,68617021276596
10	Διάδοση Ηλεκτρομαγνητικού Κύματος I	3,5952380952381
11	Διάδοση Ηλεκτρομαγνητικού Κύματος I	3,87362637362637
12	Δομές Δεδομένων	3,87278106508876
13	Εισαγωγή στην Ενεργειακή Τεχνολογία II	3,85790884718499
14	Ηλεκτρικές Μετρήσεις I	3,68405797101449
15	Ηλεκτρονική II	3,58072916666667
16	Συστήματα Αυτομάτου Ελέγχου I	3,51889168765743

Σχήμα 62: Αποτελέσματα πρώτου query

2. Αριθμός ατόμων που πέρασαν σε μαθήματα του καθηγητή Δημάκη:

```
SELECT ES.Title, ES.Passed
FROM Exams_Subjects ES, ProfessorGroups PG, Professors P
WHERE P.Last_Name = N'Δημάκης'
AND PG.Professor_Id = P.Id
AND ES.Teachings_Id = PG.Teaching_Id
```

	Title	Passed
1	Αναλογικές Τηλεπικοινωνίες	16
2	Αναλογικές Τηλεπικοινωνίες	18
3	Αναλογικές Τηλεπικοινωνίες	10
4	Αναλογικές Τηλεπικοινωνίες	16
5	Αναλογικές Τηλεπικοινωνίες	21
6	Αναλογικές Τηλεπικοινωνίες	23
7	Αναλογικές Τηλεπικοινωνίες	22
8	Αναλογικές Τηλεπικοινωνίες	39
9	Αναλογικές Τηλεπικοινωνίες	32
10	Αναλογικές Τηλεπικοινωνίες	47
11	Αναλογικές Τηλεπικοινωνίες	42
12	Αναλογικές Τηλεπικοινωνίες	59
13	Αναλογικές Τηλεπικοινωνίες	69
14	Αναλογικές Τηλεπικοινωνίες	84
15	Αναλογικές Τηλεπικοινωνίες	150
16	Αναλογικές Τηλεπικοινωνίες	267

Σχήμα 63: Αποτελέσματα δεύτερου query

3. Πού κάνουν πρακτική οι φοιτητές του τομέα Ηλεκτρονικής τα τελευταία 5 χρόνια:

```
SELECT IST.Student_Id, I.Organization, I.Country
FROM InternshipStudents IST, BranchStudents BS, Internships I
WHERE IST.Student_Id = BS.Student_Id
AND I.Id = IST.Internship_Id
AND 2015-IST.Year <= 5
AND BS.Branch_Id = 1
```

	Student_Id	Organization	Country
1	6799	Renel	Ελλάδα
2	7202	OTE	Ελλάδα
3	7399	Δημόκριτος	Ελλάδα
4	7400	ΙΠΤΗΛ	Ελλάδα
5	7414	IMET	Ελλάδα

Σχήμα 64: Αποτελέσματα τρίτου query

4. Πόσοι φοιτητές παίρνουν τομέα ανά εξάμηνο:

```
SELECT Semester, COUNT(*) AS Students_Picked
FROM dbo.BranchStudents
GROUP BY Semester
ORDER BY Semester
```

	Semester	Students_Picked
1	6	347
2	7	326
3	8	147
4	9	153
5	10	87
6	11	28
7	12	18
8	13	22
9	14	10
10	15	10

Σχήμα 65: Αποτελέσματα τέταρτου query

5. Συνολικό ποσοστό επιτυχίας εξεταστικής Σεπτεμβρίου 2012:

```
SELECT A.Year, A.Period, ((A.Passed * 1.0) / (A.Passed + A.Failed)) AS
Success_Rate
FROM(
    SELECT Year,Period,SUM(Passed) AS Passed,SUM(Failed) AS Failed
    FROM dbo.Exams_Subjects
    WHERE Period = N'Σ' AND Year = 2012
    GROUP BY Year,Period) A
```

	Year	Period	Success_Rate
1	2012	Σ	0.214768024395

Σχήμα 66: Αποτελέσματα πέμπτου query

6. Ποια μαθήματα του 6^{ου} εξαμήνου είχαν τη μεγαλύτερη προτίμηση για το εαρινό εξάμηνο του 2012:

```
SELECT Title
FROM Exams_Subjects
WHERE Year = 2012 AND Period = N'E' AND Semester = 6
EXCEPT
SELECT DISTINCT A.Title
FROM (
    SELECT Title, Subscribed
    FROM dbo.Exams_Subjects
    WHERE Year = 2012 AND Period = N'E' AND Semester = 6) A
, (
    SELECT Title, Subscribed
    FROM dbo.Exams_Subjects
    WHERE Year = 2012 AND Period = N'E' AND Semester = 6) B
WHERE A.Subscribed < B.Subscribed
```

	Title
1	Διακριτά Μαθηματικά

Σχήμα 67: Αποτελέσματα έκτου query

7. Κατά μέσο όρο πόσους εισακτέους δέχεται το τμήμα από το 2000 και έπειτα:

```
SELECT AVG(Entrants) AS Average_Entrants
FROM Year_Statistics
WHERE Year >= 2000
```

	Average_Entrants
1	237

Σχήμα 68: Αποτέλεσμα έβδομου query

8. Πόσοι ήταν οι περισσότεροι φοιτητές που δήλωσαν τομέα τηλεπικοινωνιών σε ένα έτος από το 2004 και έπειτα:

```
SELECT Picked_Telecommunications
FROM Year_Statistics
WHERE Year > 2004 AND Picked_Telecommunications IS NOT NULL
EXCEPT
SELECT A.Picked_Telecommunications
FROM (
    SELECT DISTINCT Picked_Telecommunications
    FROM Year_Statistics
    WHERE Year > 2004 AND Picked_Telecommunications IS NOT NULL) A , (
    SELECT DISTINCT Picked_Telecommunications
    FROM Year_Statistics
    WHERE Year > 2004 AND Picked_Telecommunications IS NOT NULL) B
WHERE A.Picked_Telecommunications < B.Picked_Telecommunications
```

	Picked_Telecommunications
1	91

Σχήμα 69: Αποτέλεσμα όγδοου query

9. Προτίμηση συγγραμματος για τις βάσεις δεδομένων την τετραετία 2010-2014:

```
SELECT ISBN, Title, Number_Of_Picks
FROM (
    SELECT Book_id, COUNT(*) AS Number_Of_Picks
    FROM (
        SELECT Book_id
        FROM BookApplications B JOIN (
            SELECT Id
            FROM dbo.Subjects
            WHERE Title = N'Βάσεις Δεδομένων' AND Semester = 9
        ) C ON B.Subject_Id = C.Id
        WHERE Date >='20101001' AND Date <='20150301'
    ) D
    GROUP BY D.Book_id
) E JOIN Books ON Books.Id = E.Book_id
```

	ISBN	Title	Number_Of_Picks
1	9608105870	Συστήματα Βάσεων Δεδομένων	5
2	9789604184118	Συστήματα Διαχείρισης Βάσεων Δεδομένων	3
3	9789605126230	Συστήματα Βάσεων Δεδομένων	2

Σχήμα 70: Αποτελέσματα ένατου query

10. Πόσες φορές, κατά μέσο όρο, χρειάζεται κάποιος φοιτητής να δώσει το μάθημα ΣΑΕ Ι του πέμπτου εξαμήνου μέχρι να το περάσει:

```
SELECT AVG(E.Times_Taken) AS Average_Times
FROM (
    SELECT COUNT(*) AS Times_Taken
    FROM Grades B JOIN (
        SELECT C.Id
        FROM Teachings C JOIN (
            SELECT Id
            FROM dbo.Subjects
            WHERE Title = N'Συστήματα Αυτομάτου Ελέγχου Ι' AND Semester =5
        ) D ON C.Subject_id = D.Id
    ) A ON A.Id = B.Teachings_Id
    GROUP BY B.Student_Id
) AS E
```

	Average_Times
1	2

Σχήμα 71: Αποτέλεσμα δέκατου query

11. Πόσα μαθήματα πέρασαν οι φοιτητές την χρονιά που έκαναν πρακτική:

```
SELECT G.Student_Id,E.Year,E.Period,Count(G.Student_Id) Passed
FROM Grades G,SubjectExams SE,Exams E, InternshipStudents I
WHERE G.Grade >= 5
AND E.Id = SE.Exams_Id
AND SE.Teachings_Id = G.Teachings_Id
AND E.Year = I.Year
AND I.Student_Id = G.Student_Id
GROUP BY G.Student_Id,E.Year,E.Period
HAVING Count(G.Student_Id) >= 5
```

	Student_Id	Year	Period	Passed
1	6799	2014	Σ	9
2	7414	2014	Σ	11
3	7508	2014	Σ	10
4	6799	2014	X	8
5	7414	2014	X	8
6	7508	2014	X	7

Σχήμα 72: Αποτελέσματα ενδέκατου query

12. Πόσα μαθήματα έχει περάσει ο κάθε φοιτητής:

```
SELECT Student_Id,Count(Student_Id)
FROM Grades
WHERE Grade >= 5
GROUP BY Student_Id
```

	Student_Id	(No column name)
1	6600	19
2	6601	20
3	6602	17
4	6603	24
5	6604	25
6	6605	19
7	6606	18
8	6607	20
9	6608	18
10	6609	22
11	6610	24
12	6611	23
13	6612	18
14	6613	17
15	6614	17
16	6615	23
17	6616	21

Σχήμα 73: Αποτελέσματα δωδέκατου query

13. Φοιτητές που έχουν περάσει πάνω από 5 μαθήματα σε μία εξεταστική περίοδο:

```

SELECT G.Student_Id, E.Year, E.Period, Count(G.Student_Id) Passed
FROM Grades G, SubjectExams SE, Exams E
WHERE G.Grade >= 5
AND E.Id = SE.Exams_Id
AND SE.Teachings_Id = G.Teachings_Id
GROUP BY G.Student_Id, E.Year, E.Period
HAVING Count(G.Student_Id) >= 5

```

	Student_Id	Year	Period	Passed
1	7160	2014	E	5
2	7584	2014	X	6
3	8052	2014	X	7
4	7118	2014	X	7
5	7864	2014	E	5
6	6801	2014	X	8
7	6722	2014	X	5
8	6876	2013	X	5
9	7624	2014	X	5
10	6864	2013	Σ	5
11	6987	2014	Σ	13
12	8201	2014	Σ	5
13	7066	2014	Σ	5
14	7782	2014	X	5
15	6917	2014	X	6
16	7511	2014	E	6
17	7145	2014	Σ	10

Σχήμα 74: Αποτελέσματα δέκατου-τρίτου query

Triggers

Στην ενότητα αυτή περιγράφονται τα διάφορα triggers που υλοποιήθηκαν στους πίνακες. Συνολικά η βάση έχει 8 triggers:

```
select name as 'Trigger', object_name(parent_obj) as 'Table'
from sysobjects
where xtype = 'TR'
```

	Trigger	Table
1	ErasmusStudents_Semester	ErasmusStudents
2	Exams_Year	Exams
3	InternshipStudents_Semester	InternshipStudents
4	HigherSemesterSubject	Grades
5	SubjectExam	Grades
6	AlreadyPassed	Grades
7	BranchStudents_Semester	BranchStudents
8	BranchStudents_PassedSubjects	BranchStudents

Στην συνέχεια παραθέτουμε τον SQL κώδικα δημιουργίας του καθενός και μία σύντομη περιγραφή του.

- ErasmusStudents_Semester

```
CREATE TRIGGER [dbo].[ErasmusStudents_Semester] ON [dbo].[ErasmusStudents]
AFTER INSERT,UPDATE
AS
IF EXISTS (SELECT * FROM Students S, INSERTED i
WHERE S.Id = i.Student_id
AND i.Semester > S.Semester)
BEGIN
    RAISERROR (N'Tο εξάμηνο Erasmus πρέπει να είναι μικρότερο ή ίσο από το εξάμηνο
φοίτησης (Student.Semester)',-1,-1)
    ROLLBACK TRANSACTION
END
```

Το trigger αυτό ελέγχει αν το τρέχον εξάμηνο φοίτησης του φοιτητή είναι τουλάχιστον ίσο με το εξάμηνο στο οποίο αυτός έκανε πρακτική. Αν δεν είναι, ακυρώνει το INSERT ή το UPDATE.

- Exams_Year

```
CREATE TRIGGER [dbo].[Exams_Year] ON [dbo].[Exams]
AFTER INSERT,UPDATE
AS
IF NOT EXISTS (SELECT * FROM Teachings T, INSERTED i
WHERE T.Year = i.Year)
BEGIN
    RAISERROR (N'Δεν έχουν διδαχτεί μαθήματα για αυτό το έτος!','-1,-1)
    ROLLBACK TRANSACTION
END
```

Το trigger αυτό ελέγχει, αν πάει να δημιουργηθεί μία εξεταστική για ένα έτος που δεν έχουν γίνει διδασκαλίες, από ανθρώπινο λάθος. Δηλαδή κανονικά στην αρχή του εξαμήνου δημιουργούνται οι διδασκαλίες και στο τέλος του εξαμήνου οι εξεταστικές.

- InternshipStudents_Semester

```
CREATE TRIGGER [dbo].[InternshipStudents_Semester] ON [dbo].[InternshipStudents]
AFTER INSERT,UPDATE
AS
IF EXISTS (SELECT * FROM Students S, INSERTED i
WHERE S.Id = i.Student_id
AND i.Semester > S.Semester)
BEGIN
    RAISERROR (N'Το εξάμηνο πρακτικής πρέπει να είναι μικρότερο ή ίσο από το εξάμηνο
φοίτησης (Student.Semester)', -1, -1)
    ROLLBACK TRANSACTION
END
```

Το trigger της πρακτικής για το Erasmus.

- HigherSemesterSubject

```
CREATE TRIGGER [dbo].[HigherSemesterSubject] ON [dbo].[Grades]
AFTER INSERT,UPDATE
AS
IF EXISTS (SELECT *
FROM Teachings T, inserted i, Subjects S, Students St
WHERE i.Teachings_Id = T.Id
AND S.Id = T.Subject_id
AND St.Id = i.Student_Id
AND S.Semester > St.Semester)
BEGIN
    RAISERROR (N'Το εξάμηνο του μαθήματος είναι μεγαλύτερο από το εξάμηνο φοίτησης του
φοιτητή.', -1, -1)
    ROLLBACK TRANSACTION
END
```

Το trigger αυτό ενεργοποιείται όταν γίνεται μία εισαγωγή ή ενημέρωση στον πίνακα Grades με βαθμολογία μαθήματος μεγαλύτερου εξαμήνου από αυτό που είναι ο φοιτητής.

- SubjectExam

```
CREATE TRIGGER [dbo].[SubjectExam] ON [dbo].[Grades]
AFTER INSERT,UPDATE
AS IF NOT EXISTS(SELECT *
FROM SubjectExams SE, inserted i, Exams E
WHERE i.Teachings_Id = SE.Teachings_Id
AND E.Id = SE.Exams_Id
AND i.Period = E.Period)
BEGIN
    RAISERROR(N'Το μάθημα δεν έχει εγγραφή στον πίνακα SubjectExams (δεν έχει
εξεταστεί)', -1, -1)
    ROLLBACK TRANSACTION
```

END

Το trigger αυτό ενεργοποιείται, όταν περνιέται ένας βαθμός για ένα μάθημα το οποίο δεν έχει εξεταστεί (ή τουλάχιστον δεν έχει εγγραφεί στον πίνακα με τις εξεταστικές μαθήματος).

- Already_Passed

```
CREATE TRIGGER [dbo].[AlreadyPassed] ON [dbo].[Grades]
AFTER INSERT,UPDATE
AS
IF EXISTS(SELECT *
FROM Grades G, inserted i, Teachings T1,Teachings T2, Subjects S
WHERE i.Teachings_Id = T1.Id
AND G.Student_Id = i.Student_Id
AND G.Teachings_Id != i.Teachings_Id
AND G.Teachings_Id = T2.Id /* past grades of this student */
AND T1.Subject_id = T2.Subject_id /* past grades on this (inserted) subject */
AND G.Grade >= 5)
BEGIN
    RAISERROR(N'Ο φοιτητής έχει περάσει αυτό το μάθημα',-1,-1)
    ROLLBACK TRANSACTION
END
```

Το trigger αυτό ενεργοποιείται όταν ένας φοιτητής παίρνει ξανά βαθμό για μάθημα που έχει περάσει.

- BranchStudents_Semester

```
CREATE TRIGGER [dbo].[BranchStudents_Semester] ON [dbo].[BranchStudents]
AFTER INSERT,UPDATE
AS
IF EXISTS (SELECT * FROM Students S, INSERTED i
WHERE S.Id = i.Student_id
AND i.Semester > S.Semester)
BEGIN
    RAISERROR (N'Το εξάμηνο τομέα πρέπει να είναι μικρότερο ή ίσο από το εξάμηνο
φοίτησης (Student.Semester)',-1,-1)
    ROLLBACK TRANSACTION
END
```

Το trigger αυτό ελέγχει αν το εξάμηνο το οποίο παίρνει ο φοιτητής τομέα είναι μικρότερο ή ίσο με το εξάμηνο φοίτησης. Από την άλλη, ο έλεγχος για το αν είναι μεγαλύτερο του 6 γίνεται από check constraint.

- BranchStudents_PassedSubjects

```
CREATE TRIGGER [dbo].[BranchStudents_PassedSubjects] ON [dbo].[BranchStudents]
AFTER INSERT AS
DECLARE @passed int
SELECT @passed = COUNT(*)
FROM Grades G, inserted i, Subjects S, Teachings T
WHERE G.Grade >= 5
AND i.Student_Id = G.Student_Id
AND G.Teachings_Id = T.Id
AND T.Subject_id = S.Id
AND S.Semester <= 5 /* Only check core cycle exams */
```

```
if @passed < 17
BEGIN
    RAISERROR (N'Ο φοιτητής δεν έχει περάσει 17 μαθήματα!', -1, -1)
    ROLLBACK TRANSACTION
END
```

Το trigger αυτό ελέγχει αν ο νέος φοιτητής που παίρνει τομέα έχει 17 μαθήματα περασμένα. Ο έλεγχος γίνεται από τις βαθμολογίες του φοιτητή στον πίνακα Grades. Το trigger αυτό, ενδεχομένως, να επηρεάζει την απόδοση κατά τις μαζικές εισαγωγές φοιτητών. Από την άλλη, αυτές πραγματοποιούνται μία φορά τον χρόνο οπότε δεν θα έπρεπε να απασχολεί.

Εισαγωγή δεδομένων στην βάση

Τρόποι εισαγωγής

Προκειμένου να δοκιμάσουμε όλα τα queries, τις όψεις, τους περιορισμούς ακεραιότητας και τα triggers που σχεδιάσαμε στην βάση, αποφασίσαμε να την γεμίσουμε με πραγματικά και μη δεδομένα. Για πίνακες των οποίων το γέμισμα ήταν πολύπλοκο ή απαιτούσε πολλά δεδομένα, αυτό έγινε αυτοματοποιημένα. Για τους υπόλοιπους που δεν υπήρχε αυτή η ευχέρεια, το γέμισμα έγινε χειροκίνητα με πραγματικά ή απλώς ρεαλιστικά δεδομένα. Πραγματικά είναι τα δεδομένα τα οποία όντως ισχύουν για το τμήμα ΗΜΜΥ ΑΠΘ, ενώ ρεαλιστικά αυτά τα οποία «μοιάζουν» με τα πραγματικά, αλλά καμία σχέση δεν έχουν με την πραγματικότητα.

Σκοπός αυτής της ενότητας είναι, να περιγράψει την διαδικασία με την οποία εισήχθησαν δεδομένα αυτόματα στην βάση. Επίσης περιέχει και πολλά παραδείγματα ερωτημάτων τόσο επιλογής όσο και εισαγωγής. Αρχικά θα κάνουμε μία αναφορά, για λόγους πληρότητας, στους πίνακες που γέμισαν χειροκίνητα. Συγκεκριμένα από αυτούς τους πίνακες, αυτοί με πραγματικά δεδομένα είναι οι εξής:

- ΚΑΘΗΓΗΤΗΣ¹
- ERASMUS
- ΠΡΑΚΤΙΚΗ
- ΤΟΜΕΑΣ
- ΣΥΓΓΡΑΜΜΑ
- ΑΙΘΟΥΣΑ
- ΜΑΘΗΜΑ-ΤΟΜΕΑΣ

Ενώ με ρεαλιστικά οι εξής:

- ΦΟΙΤΗΤΗΣ – ERASMUS
- ΦΟΙΤΗΤΗΣ – ΠΡΑΚΤΙΚΗ
- ΔΗΛΩΣΗ ΣΥΓΓΡΑΜΜΑΤΟΣ
- ΔΙΑΛΕΞΗ
- ΚΑΘΗΓΗΤΗΣ-ΔΙΔΑΣΚΑΛΙΑ

Οι υπόλοιποι πίνακες γέμισαν αυτοματοποιημένα:

- ΦΟΙΤΗΤΗΣ – ρεαλιστικά δεδομένα
- ΑΠΟΦΟΙΤΟΣ - ρεαλιστικά δεδομένα
- ΦΟΙΤΗΤΗΣ – ΤΟΜΕΑ - ρεαλιστικά δεδομένα
- ΒΑΘΜΟΛΟΓΙΑ - ρεαλιστικά δεδομένα
- ΜΑΘΗΜΑ – πραγματικά δεδομένα
- ΔΙΔΑΣΚΑΛΙΑ – πραγματικά δεδομένα
- ΕΞΕΤΑΣΗ-ΜΑΘΗΜΑΤΟΣ - πραγματικά δεδομένα
- ΕΞΕΤΑΣΤΙΚΗ – πραγματικά δεδομένα

¹ <http://ee.auth.gr/school/faculty-staff/>

Αυτοματοποίηση

Για τους πίνακες που γέμισαν αυτοματοποιημένα χρησιμοποιήθηκε η γλώσσα προγραμματισμού Ruby². Στην Ruby, οι εξωτερικές βιβλιοθήκες ονομάζονται «gems». Προκειμένου να επικοινωνήσουμε με την βάση η οποία γράφηκε σε SQL Server, χρησιμοποιήθηκε το gem TinyTds³. Με το TinyTds μπορούμε να δημιουργήσουμε έναν client, με τα στοιχεία ενός Login του SQL Server:

```
client = TinyTds::Client.new username: 'Main\Tasos', password: '*****',  
host: 'localhost', database: 'HMMYStat'
```

Μέσω του client αυτού μπορούμε να υποβάλουμε πάσης φύσεως ερωτήματα μέσω της εντολής

```
client.execute(sql)
```

όπου sql ένα **string** με το SQL ερώτημα. Συνεπώς, μπορούμε να έχουμε ελεγχόμενες από το πρόγραμμα μεταβλητές, μέσα στο SQL string αυτό, και έτσι η παραμετροποίηση των queries απλοποιείται. Το TinyTds προτιμήθηκε από την κλασική λύση τύπου Active Record⁴, μιας και τα ερωτήματα έχουν όλα γραφεί σε SQL λόγω της εργασίας.

Εισαγωγή Δεδομένων

Για την εισαγωγή των δεδομένων γράφτηκαν απλοί αλγόριθμοι ανάλογα με τις απαιτήσεις σε περιορισμών ακεραιότητας, triggers και τις ανάγκες σε δοκιμαστικά queries και όψεις. Επίσης τα triggers επηρεάζουν και την σειρά με την οποία γεμίζουν οι πίνακες. Για παράδειγμα δεν γίνεται να γεμίσει ο πίνακας ΦΟΙΤΗΤΗΣ_TΟΜΕΑΣ πριν τον πίνακα ΒΑΘΜΟΛΟΓΙΑ λόγω του trigger των 17 περασμένων μαθημάτων.

Στη συνέχεια θα παραθέσουμε μία περιγραφή και τον κώδικα της κάθε εισαγωγής δεδομένων σε Ruby:

- ΦΟΙΤΗΤΗΣ

Αρχικά γεμίσαμε τον πίνακα ΦΟΙΤΗΤΗΣ στην βάση, ώστε να έχουμε φοιτητές. Αποφασίσαμε να εισάγουμε 1600 άτομα από το 2007 μέχρι το 2014, 200 άτομα κάθε έτος. (Κατά τις δοκιμές μπήκαν 300 παραπάνω φοιτητές στο 2010, τα οποία διατηρήθηκαν. Άρα συνολικά 1900 φοιτητές).

² <http://el.wikipedia.org/wiki/Ruby>

³ https://github.com/rails-sqlserver/tiny_tds

⁴ http://en.wikipedia.org/wiki/Active_record_pattern

<u>Id</u>	Ημερομηνία_Εισαγωγής	Εξάμηνο	Μετεγγραφόμενος
int	date	tinyint	bit
	>= '1-1-1972', NN	>= 1 AND <= 20, NN	NN

1. Τα ΑΕΜ των φοιτητών ξεκινάνε από το 6600
2. Τυχαία ημερομηνία εισαγωγής τον Σεπτέμβριο
3. Το εξάμηνο υπολογίζεται με βάση τη χρονιά εισαγωγής
4. Τυχαία τιμή για το αν είναι μετεγγραφόμενος ή όχι με πιθανότητα 80% να μην είναι

Ο κώδικας (με μπλε θα επισημαίνονται τα SQL ερωτήματα):

```
students = Hash.new {|h,k| h[k] = Hash.new}

year = 2007 # students from 2007
1600.times do |i|
  random_day = rand(1...30) # random day that he came in
  if i%200 == 0 and i >= 200 # 200 students per year
    year += 1
  end
  semester = (2015-year)*2-1 # calculate his semester

  students[i]["Id"] = 6600 + i # AEM starting from 6600
  if random_day < 10
    students[i]["Date_In"] = "#{year}090#{random_day}".to_i
  else
    students[i]["Date_In"] = "#{year}09#{random_day}".to_i
  end
  students[i]["Semester"] = semester
  students[i]["Moved"] = usually_zero(0.8) # 80% chance he hasn't moved

  result = client.execute("INSERT INTO [Students] VALUES
  (#{students[i]["Id"]}, '#{students[i]["Date_In"]}', #{students[i]["Semester"]},
  #{students[i]["Moved"]})") <- SQL Εισαγωγή φοιτητών στην βάση
  result.cancel
end
```

- ΜΑΘΗΜΑ

Επειδή δεν υπάρχει πουθενά λίστα μαθημάτων με τα αντίστοιχα εξάμηνά τους, αξιοποιήθηκε μία βιβλιοθήκη που έχει γραφεί για ειδοποίηση ανακοινώσεων από το eTHMMY⁵ (main branch⁶) . Στην βιβλιοθήκη αυτή υπάρχει συνάρτηση *get_all_courses_by_semester()* η οποία επιστρέφει όλα τα μαθήματα που είναι καταγεγραμμένα στο eTHMMY με τα αντίστοιχα εξάμηνά τους.

Τα μαθήματα και τα εξάμηνα αποθηκεύτηκαν σε δύο αρχεία, subjects.txt και semesters.txt. Τα μαθήματα βασικού κύκλου που έλειπαν από το eTHMMY συμπληρώθηκαν στη βάση αργότερα. Το αν χωρίζεται σε τμήματα ή όχι ένα μάθημα, επίσης συμπληρώθηκε μετά. Το αν έχει εργαστήριο ή όχι δεν μας ενδιέφερε προς το παρόν.

<u>Id</u>	Όνομα	Εξάμηνο	Χωρίζεται	Εργαστήριο
smallint	nvarchar(50)	tinyint	bit	bit
≥0	NN	≥1 AND ≤9, NN	NN	NN

1. Όνομα μαθήματος από subjects.txt
2. Εξάμηνο μαθήματος από semesters.txt

Κώδικας:

```
subjects = []
File.open('subjects.txt', 'r') do |f|
  f.each_line do |line|
    line.gsub! /\r\n/, '' # crlf
    line.gsub! /'/, ''
    subjects << line
  end
end

semesters = []
File.open('semesters.txt', 'r') do |f|
  f.each_line do |line|
    line.gsub! /\r\n/, '' # crlf
    semesters << line
  end
end
```

⁵ <https://github.com/TasosDhm/thmmy-notifier/tree/master>

⁶ <https://github.com/iodim/thmmy-notifier>


```

end

subjects.each_with_index do |s,i|
  result = client.execute("INSERT INTO [Subject] (Title,Semester)
VALUES(N\'#{s}\',#{semesters[i]})") <- Εισαγωγή μαθημάτων στην βάση
  result.cancel
end

```

- ΔΙΔΑΣΚΑΛΙΑ

Για κάθε έτος, για κάθε μάθημα, για κάθε τμήμα μαθήματος δημιουργούμε μία διδασκαλία.

<u>Id</u>	Id_Μαθήματος	Έτος	Τμήμα
int	smallint	smallint	nchar(3)
	FOREIGN KEY ΜΑΘΗΜΑ (<u>Id</u>)	NN	{ "Α-Χ", "Χ- Ω", "Α-Ω" }, NN

1. Το id του μαθήματος το παίρνουμε από τον πίνακα ΜΑΘΗΜΑ.
2. Το έτος είναι στο διάστημα 2007..2014.
3. Το τμήμα προκύπτει ανάλογα με το αν το γνώρισμα *Splitted* του πίνακα ΜΑΘΗΜΑ είναι 1.

Κώδικας:

```

result = client.execute("SELECT Id,Splitted FROM Subjects") <- SQL επιλογής
μαθημάτων
subjects = []
result.each do |r|
  subjects << r
end
result.cancel

for i in 2007..2014 do
  class1 = 'Α-Χ'
  class2 = 'Χ-Ω'
  no_class = 'Α-Ω'
  subjects.each do |s|

```

```

        id = s["Id"]
        if s["Splitted"]
            result = client.execute("INSERT INTO [Teachings]
(Year,Class,Subject_Id) VALUES ({i},N'#{class1}',#{id})") <- SQL εισαγωγής
διδασκαλιών
            result.cancel
            result = client.execute("INSERT INTO [Teachings]
(Year,Class,Subject_Id) VALUES ({i},N'#{class2}',#{id})")
            result.cancel
        else
            result = client.execute("INSERT INTO [Teachings]
(Year,Class,Subject_Id) VALUES ({i},N'#{no_class}',#{id})")
            result.cancel
        end
    end
end
end

```

- ΕΞΕΤΑΣΤΙΚΗ

Στον πίνακα αυτόν απλώς δημιουργήθηκαν αυτόματα τρεις εξεταστικές για κάθε έτος.

<u>Id</u>	Έτος	Περίοδος
int	year	nchar
≥0	≥1972,NN	{"X","E","Σ"},NN

Κώδικας:

```

for i in 2007..2014 do
    result = client.execute("INSERT INTO [Exams] (Year,Period) VALUES
({i},N'X')") <- SQL εισαγωγής ΕΞΕΤΑΣΤΙΚΩΝ
    result.cancel
    result = client.execute("INSERT INTO [Exams] (Year,Period) VALUES
({i},N'E')")
    result.cancel
    result = client.execute("INSERT INTO [Exams] (Year,Period) VALUES
({i},N'Σ')")
    result.cancel
end

```

end

- ΕΞΕΤΑΣΗ-ΜΑΘΗΜΑΤΟΣ

Τώρα που έχουμε όλες τις διδασκαλίες και όλες τις εξεταστικές, για κάθε διδασκαλία για κάθε εξεταστική δημιουργούμε εξεταστικές μαθήματος.

<u>Id Εξεταστικής</u>	<u>Id Διδασκαλίας</u>	Εγγεγραμμένοι	Ημέρες_Αποτελεσμάτων
int	int	smallint	tinyint
FOREIGN KEY ΕΞΕΤΑΣΤΙΚΗ (<u>Id</u>)	FOREIGN KEY ΔΙΔΑΣΚΑΛΙΑ (<u>Id</u>)	NN	

1. Τα id εξεταστικής και διδασκαλίας πάρθηκαν από τους πίνακες ΕΞΕΤΑΣΤΙΚΗ και ΔΙΔΑΣΚΑΛΙΑ, αντίστοιχα.
2. Οι εγγεγραμμένοι και οι ημέρες αποτελεσμάτων είναι τυχαίοι αριθμοί στα διαστήματα 200..300 και 10..20 αντίστοιχα.
3. Οι εξεταστικές των μαθημάτων με περιττό εξάμηνο είναι Χειμερινό, Σεπτέμβριος ενώ των ζυγών Εαρινό, Σεπτέμβριος.

Κώδικας:

```
odd_semesters = []
result = client.execute("SELECT * FROM Subjects WHERE Semester % 2 = 1") <-
SQL για να επιλέξουμε τα μαθήματα χειμερινού
result.each do |r|
  odd_semesters << r["Id"]
end
result.cancel

even_semesters = []
result = client.execute("SELECT * FROM Subjects WHERE Semester % 2 = 0") <-
SQL για να επιλέξουμε τα μαθήματα εαρινού
result.each do |r|
  even_semesters << r["Id"]
end
result.cancel
```

```

odd_semesters.each do |os|
  subscribed = rand(200..300)
  days = rand(10..20)
  sql = "INSERT INTO SubjectExams
(Teachings_Id,Exams_Id,Subscribed,Result_Days)
      (SELECT T.Id,E.Id,#{subscribed},#{days} FROM Teachings T, Exams E
      WHERE T.Subject_Id=#{os}
      AND E.Year=T.Year
      AND (E.Period=N'X' OR E.Period=N'Σ'))" <- SQL για να εισάγουμε
ΕΞΕΤΑΣΤΙΚΗ μαθήματος στην βάση
  result = client.execute(sql)
  result.cancel
end

even_semesters.each do |es|
  subscribed = rand(200..300)
  days = rand(10..20)
  sql = "INSERT INTO SubjectExams
(Teachings_Id,Exams_Id,Subscribed,Result_Days)
      (SELECT T.Id,E.Id,#{subscribed},#{days} FROM Teachings T, Exams E
      WHERE T.Subject_Id=#{es}
      AND E.Year=T.Year
      AND (E.Period=N'E' OR E.Period=N'Σ'))"
  result = client.execute(sql)
  result.cancel
end

```

- ΒΑΘΜΟΛΟΓΙΑ

Πλέον, έχουμε φοιτητές, μαθήματα, εξεταστικές, διδασκαλίες και εξετάσεις μαθημάτων. Οπότε, πληρούνται όλοι οι περιορισμοί στον πίνακα ΒΑΘΜΟΛΟΓΙΑ για να δημιουργήσουμε εικονικές βαθμολογίες.

<u>Id Διδασκαλίας</u>	<u>Id Φοιτητή</u>	<u>Περίοδος</u>	Βαθμός
int	int	nVarChar(50)	real
FOREIGN KEY ΔΙΔΑΣΚΑΛΙΑ (<u>Id</u>)	FOREIGN KEY ΦΟΙΤΗΤΗΣ (<u>Id</u>)	{ "X", "E", "Σ" }, NN	≥0 AND ≤10, NN

Ο αλγόριθμος παραγωγής τυχαίων βαθμολογιών είναι αρκετά πολύπλοκος και απαιτεί μία σύντομη βήμα-βήμα επεξήγηση. Αναφέρουμε ότι δημιουργήθηκαν μόνο βαθμολογίες βασικού κύκλου μιας και αυτές ικανοποίησαν τις απαιτήσεις μας σε δοκιμές μέχρι και τον τομέα (οι απόφοιτοι δεν έχουν περιορισμό).

Αρχικά, επιλέγουμε όλους τους φοιτητές και τους αποθηκεύουμε στην μνήμη του προγράμματος:

```
result = client.execute("SELECT Id,Semester FROM Students")
students = result.each
result.cancel
```

Για κάθε έναν από αυτούς τους φοιτητές, οι οποίοι πλέον βρίσκονται στο hash students.

```
students.each do |s|
```

1. Μιας και ο φοιτητής δεν υπάρχει, παράγουμε ένα τυχαίο τμήμα να ανήκει (1 ή 2).
2. Βρίσκουμε ποια μαθήματα βασικού κύκλου έχει παρακολουθήσει από το εξάμηνό του από την βάση με SQL ερώτημα.
3. Για κάθε ένα από αυτά τα μαθήματα ανάλογα με το ποιο εξάμηνο είναι ο φοιτητής δίνουμε μία πιθανότητα να το έχει περάσει.

(με κόκκινο χρώμα τα βήματα)

```
#generate a random student class since these are non existing students
1 -> student_class = rand(1..2)
2 -> #find which core courses the student has taken
sql = "SELECT S.Id, S.Splitted
FROM Subjects S, Students St
WHERE St.Semester >= S.Semester
AND S.Semester <= 5
AND St.Id = #{s["Id"]}" <- SQL επιλογής μαθημάτων που παρακολούθησε
result = client.execute(sql)
```

```

student_core_courses = result.each
result.cancel
#randomize for each of those core courses "which ones he passed"
passed_subjects = {}
3 -> student_core_courses.each do |scc|
    subject_id = scc["Id"].to_i
    random_chance = rand(0.0..1.0) #random chance that the student
passed the subject. Different for each student and each subject
    if s["Semester"] <= 5
        if usually_zero(random_chance) == 0 # a core cycle student
has a mediocre chance to pass a core course
            passed_subjects[subject_id] = 1
        else
            passed_subjects[subject_id] = 0
        end
    elsif s["Semester"] > 5 and s["Semester"] <= 10
        if usually_zero(random_chance + 0.1) == 0 # a
specialization cycle student has decent chance to pass a core course
            passed_subjects[subject_id] = 1
        else
            passed_subjects[subject_id] = 0
        end
    elsif s["Semester"] > 10
        if usually_zero(random_chance + 0.15) == 0 # an over normal
period student has a good chance to pass a core course
            passed_subjects[subject_id] = 1
        else
            passed_subjects[subject_id] = 0
        end
    end
end
end

```

Πλέον, έχουμε στην μνήμη έναν hash με τα μαθήματα του φοιτητή και αν τα πέρασε ή όχι. Τώρα, για κάθε ένα από αυτά τα μαθήματα (και ενώ ακόμα είμαστε σε έναν φοιτητή):

```
passed_subjects.each do |subject_id,passed|
```

1. Βρίσκουμε σε ποιες εξεταστικές εξετάστηκε αυτό:

```
#find the exams the subject was examed
```

```

if student_core_courses[i]["Splitted"] == true
  if student_class == 1
    sql = "SELECT SE.Teachings_Id, E.Period
          FROM SubjectExams SE, Teachings T, Subjects S, Exams

          WHERE S.Id = #{subject_id}
          AND T.Subject_Id = S.Id
          AND SE.Teachings_Id = T.Id
          AND E.Id = SE.Exams_Id
          AND T.Class = N'A-X'" <- SQL εξεταστικών μαθήματος
  else
    sql = "SELECT SE.Teachings_Id, E.Period
          FROM SubjectExams SE, Teachings T, Subjects S, Exams

          WHERE S.Id = #{subject_id}
          AND T.Subject_Id = S.Id
          AND SE.Teachings_Id = T.Id
          AND E.Id = SE.Exams_Id
          AND T.Class = N'X-Ω'"
  end
else
  sql = "SELECT SE.Teachings_Id, E.Period
        FROM SubjectExams SE, Teachings T, Subjects S, Exams E
        WHERE S.Id = #{subject_id}
        AND T.Subject_Id = S.Id
        AND SE.Teachings_Id = T.Id
        AND E.Id = SE.Exams_Id
        AND T.Class = N'A-Ω'"
End
result = client.execute(sql)
subject_exams = result.each

exams = []
subject_exams.each do |se|
  exams << se["Period"]

```

```

end
teachings = []
subject_exams.each do |se|
    teachings << se["Teachings_Id"]
end

number_of_exams = exams.length

```

2. Υπολογίζουμε πόσες εξεταστικές πήρε για να το περάσει (αν το πέρασε τελικά, passed = 1).
3. Υπολογίζουμε τι βαθμό πήρε σε κάθε μία από αυτές τις εξεταστικές.

```

#Generate random grades and number of failed attempts

#How many tries did it take?
2 -> number_of_tries = rand(1..rand(3..4))

#Which exams did the tries happen?
last_try = 0
tries = Array.new(number_of_tries) {|a| a={"Period" => nil,
"Teachings_Id" => nil, "Grade" => nil}}
3 -> for k in 1..number_of_tries do
    upper_limit = number_of_exams-(number_of_tries-k) #make
sure that the first try doesnt happen in 2014 while the last exam is in 2014
and relevant senarios
    random_index = rand(last_try...upper_limit)
    random_grade = nil
    if k == number_of_tries
        if passed == 1 #if this is the successful try
            random_grade = rand(5..10)
        else
            random_grade = rand(1..4)
        end
    else
        random_grade = rand(1..4)
    end
    tries[k-1]["Period"] = exams[random_index]
    tries[k-1]["Teachings_Id"] = teachings[random_index]
    tries[k-1]["Grade"] = random_grade
end

```



```

last_try = random_index + 1
end

```

Τέλος, εισάγουμε όλες αυτές τις «προσπάθειες» που βρίσκονται στο hash tries, στη βάση:

```

tries.each do |try|
  sql = "INSERT INTO Grades
(Student_Id,Teachings_Id,Period,Grade)
VALUES ({s["Id"]}, #{try["Teachings_Id"]},
N'#{try["Period"]}', #{try["Grade"]})" <- SQL εισαγωγής βαθμολογιών
  result = client.execute(sql)
  result.cancel
end

```

Επειδή ο συγκεκριμένος κώδικας είναι μεγάλος, προστέθηκε και στα αρχεία τις εργασίας με όνομα

`generate_grades.rb`

Η όλη διαδικασία χρονομετρήθηκε και διήρκεσε περίπου 90 δευτερόλεπτα, ενώ έγινε εισαγωγή 123048 εγγραφών στον πίνακα Grades.

Τελικά, κανένα από τα trigger δεν διαμαρτυρήθηκε και, πλέον, μπορούσαμε να δούμε και στατιστικά μαθήματος μέσω της όψης ΣΤΑΤΙΣΤΙΚΑ_ΜΑΘΗΜΑΤΟΣ.

- ΦΟΙΤΗΤΗΣ – ΤΟΜΕΑΣ

Τώρα που έχουμε βαθμολογίες, μπορούμε να βάλουμε φοιτητές στους τομείς που έχουν 17 μαθήματα και πάνω, χωρίς να διαμαρτυρηθεί το trigger του πίνακα.

<u>Id Φοιτητή</u>	<u>Id Τομέα</u>	Εξάμηνο	Year
int	tinyint	tinyint	smallint
FOREIGN KEY ΦΟΙΤΗΤΗΣ (<u>Id</u>)	FOREIGN KEY ΤΟΜΕΑΣ (<u>Id</u>)	<=ΦΟΙΤΗΤΗΣ (Εξάμηνο) AND >=6, NN	NN

1. Βρίσκουμε τους φοιτητές που έχουν περάσει 17 μαθήματα.
2. Για κάθε έναν από αυτούς δίνουμε ένα τυχαίο τομέα (50% ηλεκτρονική, 30% Ενέργεια 20% Τηλεπικοινωνίες).

Κώδικας:

```

# select students eligible for branch
1 -> sql = "SELECT S.Id,S.Semester,Year(S.Date_In) Year

```

```

FROM Grades G,Students S
WHERE G.Student_Id = S.Id
AND G.Grade >= 5
AND S.Semester >=6
GROUP BY S.Id,S.Semester,S.Date_In
HAVING COUNT(*) >= 17" <- SQL φοιτητών με 17 περασμένα
result = client.execute(sql)
eligible_students = result.each
result.cancel

2 -> eligible_students.each do |es|
  #which semester did the student choose the branch?
  random_semester = rand(6..es["Semester"])
  year = es["Year"].to_i + ((random_semester.to_i)/2).ceil
  #which branch did he pick?
  random_branch = nil
  case rand(1..100)
  when 1..50 then random_branch = 1
  when 50..80 then random_branch = 2
  when 80..100 then random_branch = 3
  end
  sql = "INSERT INTO BranchStudents
        (Student_Id,Branch_Id,Semester,Year) VALUES
        ({es["Id"]},{random_branch},{random_semester},{year})" <- SQL εισαγωγής
        φοιτητών στους BranchStudents
  result = client.execute(sql)
  result.cancel
end

```

- ΑΠΟΦΟΙΤΟΣ

<u>Id Φοιτητή</u>	Ημερομηνία_Αποφοίτησης	Βαθμός_Πτυχίου
int	date	real
FOREIGN KEY ΦΟΙΤΗΤΗΣ (<u>Id</u>)	>= '1-1-1977', NN	>=5 AND <= 10, NN

1. Βρίσκουμε τους φοιτητές που έχουν εξάμηνο πάνω από 10.
2. Τυχαία αποφασίζουμε αν αποφοίτησαν ή όχι.

Κώδικας:

```
# find all the eligible for graduation students
1 -> sql = "SELECT BS.Student_Id,S.Semester, Year(S.Date_In) Year
FROM BranchStudents BS, Students S
WHERE S.Id = BS.Student_Id
AND S.Semester >= 10" <- SQL φοιτητών με εξάμηνο > 10
result = client.execute(sql)
eligible_graduates = result.each
result.cancel

2 -> eligible_graduates.each do |eg|
  graduate = nil
  case eg["Semester"]
  when 10
    graduate = rand < 0.7 ? 1 : 0
  when 10...14
    graduate = rand < 0.8 ? 1 : 0
  else
    graduate = rand < 0.95 ? 1: 0
  end
  if graduate == 1
    random_day = rand(1...30) # random graduation date
    year = eg["Year"].to_i + ((eg["Semester"].to_i)/2).ceil
    graduation_date = nil
    if random_day < 10
      graduation_date = "#{year}070#{random_day}"
    end
  end
end
```

```

else
    graduation_date = "#{year}07#{random_day}"
end
random_grade = rand(6..10)
sql = "INSERT INTO GraduateStudents VALUES
({eg["Student_Id"]}, '#{graduation_date}', '#{random_grade})" <- SQL εισαγωγής
αποφοίτων
result = client.execute(sql)
result.cancel
end
end

```

Παράρτημα

Στην ενότητα αυτή παραθέτουμε κάποιες ιδέες οι οποίες θα μπορούσαν να οδηγήσουν στην σύνδεση της βάσης με ένα λογισμικό. Η λογική θα μπορούσε να ήταν η εξής:

- Η διαχείριση της βάσης - δηλαδή η εισαγωγή, η διαγραφή και η επεξεργασία των δεδομένων - να γίνεται από μία εφαρμογή διαχείρισης.
- Η προβολή των στατιστικών σε μορφή διαγραμμάτων και απλών «απαντήσεων» να γίνεται από μία εφαρμογή προβολής στατιστικών.

Οι δύο αυτές εφαρμογές θα βρίσκονται σε μία πλατφόρμα, η οποία θα υπάρχει στο εσωτερικό δίκτυο του πανεπιστημίου. Στην πλατφόρμα η σύνδεση θα γίνεται μέσω ακαδημαϊκού λογαριασμού. Στους φοιτητές θα εμφανίζεται η δεύτερη υπο-εφαρμογή ενώ στην γραμματεία/διαχειριστές η πρώτη.

Πολύ συνοπτικά η εφαρμογή διαχείρισης θα έπρεπε να ικανοποιεί της παρακάτω απαιτήσεις:

1. Η γραμματεία να μπορεί να εισάγει, να διαγράφει και να ενημερώνει δεδομένα σε όλους τους πίνακες.
2. Ο διαχειριστής πρέπει να μπορεί να εισάγει, να διαγράφει και να ενημερώνει δεδομένα σε όλους τους πίνακες.
3. Ο διαχειριστής πρέπει να μπορεί να δίνει δικαιώματα στους πίνακες.
4. Η εφαρμογή πρέπει να παρέχει τρόπο εποπτείας των πινάκων.
5. Η εφαρμογή πρέπει να εμφανίζει τα μηνύματα από τα triggers και τους περιορισμούς ακεραιότητας.
6. Η εφαρμογή πρέπει να παρέχει τρόπο εισαγωγής δεδομένων από αρχείο
7. Η εφαρμογή πρέπει να παρέχει τρόπο σύνδεσης των χρηστών της

Δηλαδή θα αποτελεί, ένα “layer of abstraction” μεταξύ SQL Management Studio και ανθρώπου, μιας και οι χρήστες που θα εισάγουν δεδομένα δεν γνωρίζουν SQL.

Από την άλλη η εφαρμογή προβολής στατιστικών έχει πολύπλοκες απαιτήσεις. Δηλαδή πρέπει:

- Να γίνει ανάλυση των ερωτημάτων που θα μπορούσε να υποβάλλει ένας φοιτητής
- Να οργανωθούν τα ερωτήματα σε γραφικό περιβάλλον
- Να οργανωθούν τα ερωτήματα σε επίπεδο κώδικα

Και τα παραπάνω αφορούν μόνο τις απαιτήσεις για συνεργασία με την βάση.

Στην συνέχεια παραθέτουμε δύο mockups της εφαρμογής διαχείρισης και ένα παραδειγματικό web application για προβολή στατιστικών.

Σύνδεση στην πλατφόρμα

HMMYStat

username

password

[forgot password?](#)

Login

Εισαγωγή βαθμολογιών από την γραμματεία

HMMYStat

GRADES

STUDENTS

GRADUATES

ERASMUS

INTERNSHIP

ERASMUS-STUDENTS

INTERNSHIP-STUDENT

BRANCH-STUDENTS

SUBJECTS

BOOKS

PROFESSORS

PROFESSOR-GROUPS

BOOK-APPLICATIONS

TEACHINGS

EXAMS

SUBJECT-EXAMS

BRANCH-SUBJECT-TV

BRANCHES

HALLS

CLASSES

Table Preview

Insert Update Delete View

Choose Subject Exams:

Subject: Databases Year: 2015 Period: Winter Class: All

Choose insert method:

Method: File Type: CSV → CSV Row format: AEM,Grade

File: C:/Users/Soula/DB_Grades_201 Browse

Insert Grades

Web App

Εδώ παραθέτουμε απλώς λειτουργικότητα back end και front end, για να δείξουμε πως θα μπορούσε η πληροφορία η οποία βρίσκεται στην βάση να εμφανιστεί στην οθόνη του χρήστη. Το απλό αυτό web app αποτελείται από 1 κουμπί και 1 περιοχή εμφάνισης γραφήματος. Είναι γραμμένο σε Ruby και αξιοποιεί τα εξής gems:

- Sinatra⁷ – framework για στήσιμο απλών minimal web app
- TinyTds⁸ για επικοινωνία με την βάση όπως στην ενότητα εισαγωγής δεδομένων
- Nyaplot⁹ για εμφάνιση γραφημάτων σε μορφή ιστοσελίδας

Η λογική είναι ότι με το πάτημα του κουμπιού

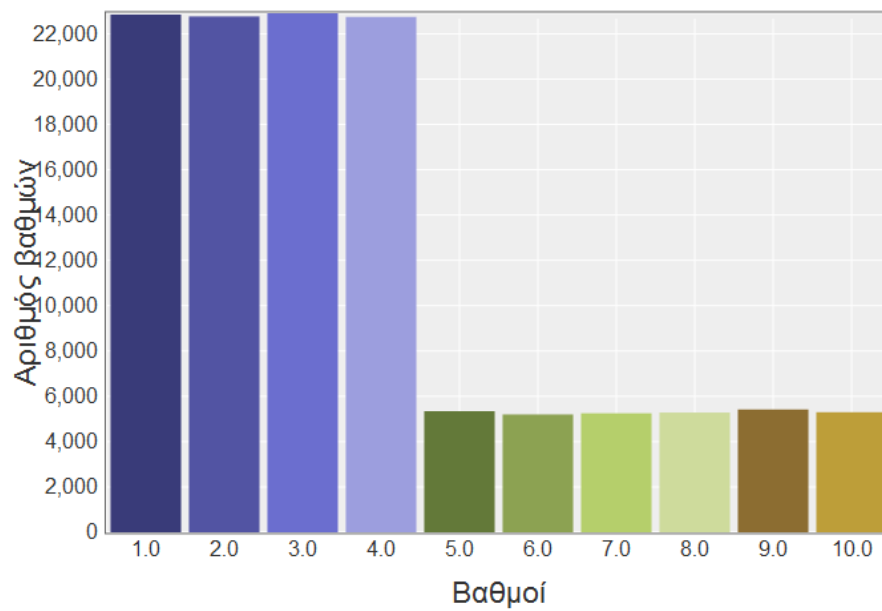
1. Στέλνεται POST HTTP request στον default web server της Ruby
2. Ο sinatra διαχειρίζεται το request και καλεί την συνάρτηση `get_all_grades`
3. Η συνάρτηση εκτελεί μέσω του `tinytds` ερώτημα στην βάση για ανάκτηση όλων των βαθμών
4. Η συνάρτηση αποθηκεύει το αποτέλεσμα στην μνήμη και καλεί την συνάρτηση `plot`
5. Η συνάρτηση `plot` δημιουργεί ένα αντικείμενο `Nyaplot`, το παραμετροποιεί κατάλληλα ώστε να εμφανιστεί ως ραβδόγραμμα με τα δεδομένα που υπολογίσαμε και το κάνει `export` σε `html`.
6. Το `plot` εμφανίζεται στην οθόνη, όντας `iframe` στο `index.html`

Το αποτέλεσμα:

⁷ <https://github.com/sinatra/sinatra>

⁸ https://github.com/rails-sqlserver/tiny_tds

⁹ <https://github.com/domitry/nyaplot>



Grades all time

Σημείωση: Στο σημείο αυτό μπορούμε να δούμε την φοβερή επιτυχία του αλγορίθμου παραγωγής τυχαίων βαθμολογιών, ο οποίος έκοψε όλους τους εικονικούς φοιτητές 90000 φορές από τις ~120000...

Ο κώδικας της εφαρμογής βρίσκεται στον φάκελο WebAppExample της εργασίας.