

5 Προγραμματισμός σε SQL

Συγγραφέας: Ευαγγελία Πιτουρά

Σκοπός

Στο κεφάλαιο αυτό παρουσιάζεται η γλώσσα SQL που είναι η πιο διαδεδομένη γλώσσα για τη διαχείριση σχεσιακών βάσεων δεδομένων. Η γλώσσα SQL παρέχει εντολές για τον ορισμό του σχήματος μιας βάσης δεδομένων, την εισαγωγή, διαγραφή και ενημέρωση δεδομένων καθώς και για τη διατύπωση ερωτημάτων για την ανάκτηση χρήσιμης πληροφορίας από τα αποθηκευμένα δεδομένα. Σκοπός αυτού του κεφαλαίου είναι η αναλυτική παρουσίαση των βασικών χαρακτηριστικών της γλώσσας και η χρήση της για τον ορισμό και το χειρισμό μιας βάσης δεδομένων.

Προσδοκώμενα Αποτελέσματα

Μετά τη μελέτη αυτού του κεφαλαίου θα μπορείτε, χρησιμοποιώντας τη γλώσσα SQL:

- να ορίσετε το σχήμα μιας σχεσιακής βάσης δεδομένων σε ένα ΣΔΒΔ,
- να διαχειρίζεστε το στιγμιότυπο μιας σχεσιακής βάσης δεδομένων μέσω της εισαγωγής, διαγραφής και ενημέρωσης πλειάδων,
- να ανακτάτε πληροφορία από μια σχεσιακή βάση δεδομένων μέσω της διατύπωσης κατάλληλων ερωτήσεων.

Έννοιες Κλειδιά

- SFW Ερώτηση
- Υπό-ερωτήσεις
- Συναθροιστικές Συναρτήσεις
- Λογική Τριών Τιμών
- Πολυσύνολα

Εισαγωγικές Παρατηρήσεις

Στα προηγούμενα κεφάλαια, μελετήσαμε το σχεδιασμό μιας βάσης δεδομένων. Στο κεφάλαιο αυτό θα δούμε πως μπορούμε να υλοποιήσουμε μια βάση δεδομένων χρησιμοποιώντας ένα σχεσιακό ΣΔΒΔ.

Η γλώσσα SQL (Structured Query Language) αποτελεί την πιο διαδεδομένη γλώσσα για τη διαχείριση σχεσιακών βάσεων δεδομένων. Αρχικά ονομαζόταν SEQUEL και υλοποιήθηκε από την IBM στα πλαίσια του ερευνητικού σχεσιακού συστήματος βάσεων δεδομένων System R. Σήμερα έχει τυποποιηθεί από την ANSI/ISO και υποστηρίζεται από τα περισσότερα Συστήματα Διαχείρισης Βάσεων Δεδομένων.

Η γλώσσα SQL αποτελείται από δύο βασικά τμήματα: τη Γλώσσα Ορισμού Δεδομένων (ΔΟΧ) (Data Definition Language (DDL)) και τη Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ) (Data Manipulation Language (DML)). Η Γλώσσα Ορισμού Δεδομένων, που παρουσιάζεται στην Ενότητα 5.1, περιλαμβάνει τις απαραίτητες εντολές για την δημιουργία και την ενημέρωση του σχήματος μιας σχεσιακής βάσης δεδομένων καθώς και των περιορισμών ακεραιότητας που ισχύουν σε αυτό.

Η Γλώσσα Χειρισμού δεδομένων περιλαμβάνει μια *γλώσσα ερωτήσεων*, (*query language*) αντίστοιχη της σχεσιακής άλγεβρας, για τη διατύπωση ερωτήσεων στο τρέχον στιγμιότυπο μιας βάσης δεδομένων. Στην Ενότητα 5.2, θα παρουσιάσουμε τα βασικά στοιχεία της γλώσσας ερωτήσεων. Η Γλώσσα Χειρισμού δεδομένων περιλαμβάνει επίσης εντολές που έχουν ως αποτέλεσμα την τροποποίηση ενός στιγμιότυπου μιας βάσης δεδομένων με την εισαγωγή, διαγραφή και τροποποίηση πλειάδων. Θα μελετήσουμε αυτές τις εντολές στην Ενότητα 5.3.

Η SQL χρησιμοποιεί τους όρους *πίνακα* για τη σχέση, *γραμμή* για την πλειάδα και *στήλη* για το γνώρισμα, τους οποίους θα χρησιμοποιούμε εναλλακτικά σε αυτό το κεφάλαιο.

5.1 Ορισμός Σχήματος

Η γενική δομή για τον ορισμό του σχήματος μιας νέας σχέσης φαίνεται στην Εικόνα 5.1. Για να ορίσουμε ένα σχήμα σχέσης (πίνακα) βαθμού n , δίνουμε το όνομα της σχέσης και στη συνέχεια προσδιορίζουμε το όνομα και το πεδίο ορισμού για καθένα από τα n γνωρίσματά του.

```

CREATE TABLE <relation-name>
  (<attribute-name> <attribute-type> [<attribute-constraint>]
  {, <attribute-name> <attribute-type> [<attribute-constraint>]}
  [<table-constraint> {, <table-constraint>}])

<attribute-constraint> ::= NOT NULL | DEFAULT <value> | PRIMARY KEY |
<table-constraint>    ::= PRIMARY KEY <attribute-list> |
                        UNIQUE <attribute-list> |
                        FOREIGN KEY <attribute-list>
                        REFERENCES <relation name>(<attribute-list>)
                        [ON DELETE (CASCADE | SET NULL | SET DEFAULT)]
                        [ON UPDATE (CASCADE | SET NULL | SET DEFAULT)]
                        CHECK ( <condition>)

```

Εικόνα 5.1 Εντολές ορισμού ενός νέου σχήματος σχέσης με χρήση του συμβολισμού BNF

Για τον ορισμό του πεδίου ορισμού ενός γνωρίσματος υπάρχουν διαθέσιμοι αρκετοί βασικοί τύποι δεδομένων που περιλαμβάνουν αριθμητικούς τύπους δεδομένων (όπως INT, REAL, FLOAT), τύπους δεδομένων για συμβολοσειρές χαρακτήρων (όπως CHAR(n) για συμβολοσειρές σταθερού μήκους n και VARCHAR(n) για συμβολοσειρές μεταβλητού μήκους έως n), καθώς και τύπους δεδομένων για λογικές μεταβλητές και σειρές δυαδικών ψηφίων. Υπάρχουν επίσης ειδικοί τύποι για την ημερομηνία (DATE) και την ώρα (TIME). Για κάποιο γνώρισμα, μπορούμε να εκφράσουμε ότι δεν επιτρέπεται να πάρει την τιμή NULL χρησιμοποιώντας το διακριτικό NOT NULL ή και να προσδιορίσουμε μια προκαθορισμένη τιμή για αυτό χρησιμοποιώντας το διακριτικό DEFAULT ακολουθούμενο από αυτήν την τιμή.

Στην Εικόνα 5.2, μπορείτε να δείτε ένα παράδειγμα που αφορά στον ορισμό των σχέσεων της βάσης δεδομένων για κινηματογραφικές ταινίες της Εικόνας 3.5. Για την ημερομηνία γέννησης ενός ηθοποιού χρησιμοποιήσαμε τον ειδικό τύπο DATE. Επίσης, ορίσαμε ότι τα γνωρίσματα Sex της σχέσης ACTOR και Role της σχέσης PLAYS δεν μπορούν να πάρουν την τιμή NULL που σημαίνει ότι κατά την εισαγωγή των αντίστοιχων πλειάδων πρέπει να καθορίζονται οι τιμές αυτών των γνωρισμάτων τους.

Η SQL υποστηρίζει τους βασικούς περιορισμούς ακεραιότητας του σχεσιακού μοντέλου που περιγράψαμε στην Ενότητα 3.1.2, συγκεκριμένα, τους περιορισμούς κλειδιού και ξένου κλειδιού. Το πρωτεύον κλειδί ορίζεται με χρήση του διακριτικού

PRIMARY KEY. Αν το υποψήφιο κλειδί αποτελείται από ένα μόνο γνώρισμα, το διακριτικό μπορεί να ακολουθεί τον ορισμό του αντίστοιχου γνωρίσματος. Μπορείτε να δείτε σχετικά παραδείγματα στην Εικόνα 5.2. Στην περίπτωση που υπάρχουν παραπάνω από ένα υποψήφια κλειδιά, μπορούμε να προσδιορίσουμε τα γνωρίσματα του υποψήφιου κλειδιού που δεν επιλέχθηκε ως πρωτεύον κλειδί, χρησιμοποιώντας το διακριτικό UNIQUE. Αυτό σημαίνει ότι δεν μπορεί να υπάρχουν δύο πλειάδες με την ίδια τιμή σε αυτά τα γνωρίσματα.

Τα ξένα κλειδιά ορίζονται στην αναφέρουσα σχέση χρησιμοποιώντας το διακριτικό FOREIGN KEY και στη συνέχεια προσδιορίζοντας την αναφερόμενη σχέση και τα αντίστοιχα γνωρίσματα της χρησιμοποιώντας το διακριτικό REFERENCES, όπως μπορείτε να δείτε στο παράδειγμα της Εικόνας 5.2. Προφανώς, ο ορισμός της αναφερόμενης σχέσης πρέπει να προηγείται του ορισμού της αναφέρουσας. Για παράδειγμα, στην Εικόνα 5.2, ο ορισμός της MOVIE πρέπει να προηγείται του ορισμού της MOVIE-GENRE, και ο ορισμός του ACTOR και MOVIE να προηγείται του ορισμού της PLAYS.

```
CREATE TABLE ACTOR
(Name VARCHAR(30) PRIMARY KEY,
Date-of-birth DATE,
Sex VARCHAR(6) NOT NULL,
Nationality VARCHAR(15));
```

```
CREATE TABLE MOVIE
(Title VARCHAR(30),
Year INT,
Runtime INT,
PRIMARY KEY (Title, Year));
```

```
CREATE TABLE PLAYS
(Name VARCHAR(30),
Title VARCHAR(30),
Year INT,
Role VARCHAR(30),
PRIMARY KEY (Name, Title, Year),
FOREIGN KEY (Name) REFERENCES
    ACTOR(Name)
ON UPDATE CASCADE,
FOREIGN KEY (Title, Year) REFERENCES
    MOVIE(Title, Year)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE MOVIE-GENRE
(Title VARCHAR(30),
Year INT,
Genre VARCHAR(15),
PRIMARY KEY (Title, Year, Genre),
FOREIGN KEY (Title, Year)
    REFERENCES MOVIE(Title, Year)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

Εικόνα 5.2 Ορισμός σε SQL του σχήματος της βάσης δεδομένων για κινηματογραφικές ταινίες

Σε περίπτωση παραβίασης ενός περιορισμού ακεραιότητας, η προκαθορισμένη δράση της SQL είναι να απορρίψει την πράξη που προκαλεί παραβίαση του

περιορισμού. Ειδικά για τις περιπτώσεις που η διαγραφή ή η τροποποίηση μιας πλειάδας της αναφερόμενης σχέσης προκαλεί παραβίαση του ξένου κλειδιού της αναφέρουσας, η SQL παρέχει τη δυνατότητα στον σχεδιαστή του σχήματος να καθορίσει εναλλακτικές ενέργειες. Οι εναλλακτικές αυτές ενέργειες είναι CASCADE, SET NULL, SET DEFAULT και προσδιορίζονται μετά τον ορισμό του ξένου κλειδιού ξεχωριστά για τις διαγραφές (ON DELETE) και τις τροποποιήσεις (ON UPDATE) όπως φαίνεται στην Εικόνα 5.2.

Προσδιορισμός της ενέργειας ON DELETE CASCADE έχει ως αποτέλεσμα στην περίπτωση που η αναφέρουσα πλειάδα διαγράφεται να διαγραφούν αυτόματα όλες οι πλειάδες που αναφέρονται σε αυτήν. Για παράδειγμα, στην Εικόνα 5.2, διαγραφή μιας ταινίας από την σχέση MOVIE έχει ως αποτέλεσμα τη διαγραφή όλων των πλειάδων της σχέσης PLAYS που αναφέρονται σε αυτήν. Αντίστοιχα, αν προσδιοριστεί η ενέργεια ON UPDATE CASCADE, στην περίπτωση που οι τιμές της αναφέρουσας πλειάδας ενημερωθούν, οι τιμές αυτές ανατίθενται αυτόματα και στις πλειάδες που αναφέρονται σε αυτήν. Οι ενέργειες ON DELETE SET NULL και ON UPDATE SET NULL θέτουν την τιμή του ξένου κλειδιού ίση με NULL στην περίπτωση που η πλειάδα στην οποία αναφέρεται διαγράφεται ή αντίστοιχα τροποποιείται. Όμοια, οι ενέργειες ON DELETE SET DEFAULT και ON UPDATE SET DEFAULT θέτουν την τιμή του ξένου κλειδιού ίση με την προκαθορισμένη για αυτό τιμή. Για παράδειγμα, στην Εικόνα 5.2, ο προσδιορισμός ON UPDATE CASCADE για το όνομα του ηθοποιού και ON UPDATE CASCADE για τον τίτλο και το έτος μιας ταινίας έχουν ως αποτέλεσμα οποιαδήποτε αλλαγή στο όνομα ενός ηθοποιού (για παράδειγμα, λόγω διόρθωσης κάποιου τυπογραφικού λάθους) ή αντίστοιχα στα στοιχεία μιας ταινίας να μεταδίδεται αυτόματα στις αντίστοιχες πλειάδες της σχέσης PLAYS ενημερώνοντας τα γνωρίσματα τους με την νέα τιμή.

Τέλος, μπορούμε να ορίσουμε σημασιολογικούς περιορισμούς χρησιμοποιώντας το προσδιοριστικό CHECK ακολουθούμενο από μια συνθήκη. Στην περίπτωση αυτή, η συνθήκη του CHECK υπολογίζεται κάθε φορά που εισάγεται ή ενημερώνεται μια πλειάδα. Η εισαγωγή ή η ενημέρωση γίνεται αποδεκτή, μόνο αν ικανοποιείται η συνθήκη.

```
DROP TABLE <relation-name> [RESTRICT | CASCADE]

ALTER TABLE <relation-name> ADD <attribute-name> <attribute-type>
ALTER TABLE <relation-name> DROP <attribute-name> [RESTRICT |
CASCADE]
```

Εικόνα 5.3 Εντολές αλλαγής σχήματος

Εντολές αλλαγής σχήματος. Η SQL παρέχει εντολές για την εξέλιξη ενός σχήματος μέσω της προσθήκης ή της διαγραφής γνωρισμάτων, περιορισμών, σχέσεων ή άλλων στοιχείων του σχήματος. Μερικές από αυτές τις εντολές φαίνονται στην Εικόνα 5.3.

Η εντολή DROP TABLE οδηγεί στη διαγραφή του σχήματος μια σχέσης. Αν χρησιμοποιηθεί η επιλογή RESTRICT το σχήμα διαγράφεται μόνο αν δεν υπάρχουν άλλες σχέσεις που αναφέρονται σε αυτό μέσω ξένων κλειδίων ή όψεων (θα τις μελετήσουμε παρακάτω) ενώ αν χρησιμοποιηθεί η επιλογή CASCADE διαγράφονται αυτόματα και όλα τα σχήματα που αναφέρονται σε αυτό.

Μπορούμε να τροποποιήσουμε ένα σχήμα με χρήση της εντολής ALTER TABLE προσθέτοντας γνωρίσματα ή περιορισμούς σε αυτό με χρήση του ADD ή να διαγράψουμε γνωρίσματα ή περιορισμούς με χρήση του DROP.

Άσκηση Αυτοαξιολόγησης 5.1: Ορισμός Σχήματος

Δώσετε τις κατάλληλες εντολές SQL για τον ορισμό του σχήματος:

(α) της βάσης δεδομένων για τις πιτσαρίες της Άσκησης Αξιολόγησης 3.3

(β) της βάσης δεδομένων για τα μηνύματα της οποίας το σχεσιακό σχήμα δίνεται στην Εικόνα 3.21.

Άσκηση Αυτοαξιολόγησης 5.2: Ξένα Κλειδιά

Θεωρείστε μια σχεσιακή βάση δεδομένων με δύο πίνακες R, S ορισμένους ως εξής:

```
CREATE TABLE R (A INT PRIMARY KEY, B INT);
```

```
CREATE TABLE S (C INT PRIMARY KEY, D INT DEFAULT 1,
FOREIGN KEY (D) REFERENCES R(A);
```

Έστω το στιγμιότυπο της Εικόνας 5.4.

(α) Δώστε το περιεχόμενο του πίνακα S μετά τη διαγραφή από την R των πλειάδων για τις οποίες A = 6, αν ορίσουμε:

- (i) ON DELETE CASCADE
- (ii) ON DELETE SET NULL,
- (iii) ON DELETE SET DEFAULT
- (iv) τίποτα από τα παραπάνω.

R		S	
A	B	C	D
1	2	2	1
2	4	3	6
3	2	5	1
9	8	4	9
6	8	7	9
5	7		
7	3		

Εικόνα 5.4 Στιγμιότυπο για την Άσκηση Αξιολόγησης 5.2

(β) Υποθέστε τώρα ότι αντί να διαγράφουμε πλειάδες, τροποποιούμε την τιμή του A στη σχέση A από 6 σε 10. Επαναλάβετε το ερώτημα (α) θεωρώντας UPDATE στη θέση του DELETE.

5.2 Ερωτήσεις στην SQL

Στην ενότητα αυτή θα περιγράψουμε το τμήμα της SQL που αφορά στη διατύπωση ερωτήσεων για την ανάκτηση πληροφορίας από τη βάση δεδομένων.

5.2.1 Η Βασική Δομή μιας Ερώτησης SQL

Η βασική δομή μιας ερώτησης SQL ακολουθεί τη select-from-where (SFW) σύνταξη που βλέπετε στην Εικόνα 5.5. Στο FROM προσδιορίζονται οι πίνακες (σχέσεις) R_1, R_2, \dots, R_m που θα συνδυαστούν για τον υπολογισμό του αποτελέσματος, στο WHERE διατυπώνεται η συνθήκη που πρέπει να ικανοποιούν οι πλειάδες του αποτελέσματος και τέλος στο SELECT παρατίθενται τα γνωρίσματα που θέλουμε να εμφανίζονται στο αποτέλεσμα.

```
SELECT A1, A2, ..., An
FROM R1, R2, ..., Rm
WHERE P
```

Εικόνα 5.5 Η βασική SFW δομή μιας ερώτησης SQL

Η βασική SFW ερώτηση της SQL αντιστοιχεί στην έκφραση

$$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_m))$$

της σχεσιακής άλγεβρας με την μόνη διαφορά ότι στην SQL δε γίνεται απαλοιφή διπλότυπων όπως γίνεται στη σχεσιακή άλγεβρα. Δηλαδή, αρχικά υπολογίζεται το καρτεσιανό γινόμενο των σχέσεων που εμφανίζονται στο FROM. Στη συνέχεια, εκτιμάται η συνθήκη P του WHERE σε κάθε πλειάδα του γινομένου και επιλέγονται αυτές που την ικανοποιούν. Τέλος, προβάλλονται οι τιμές των γνωρισμάτων που εμφανίζονται στο SELECT. Μια SFW ερώτηση καλείται και *Select-Project-Join (SPJ)* ερώτηση.

Η συνθήκη του WHERE είναι μια λογική έκφραση που αποτελείται από μία ή περισσότερες λογικές προτάσεις συνδεδεμένες με AND, OR, και NOT. Η βασική λογική πρόταση είναι της μορφής *A op B*, όπου A είναι ένα γνώρισμα των σχέσεων που εμφανίζονται στο FROM, B είναι επίσης ένα γνώρισμα των σχέσεων του FROM ή μια τιμή από το πεδίο τιμών του A και *op* ένας τελεστής σύγκρισης. Ο τελεστής *op* μπορεί να είναι = ή <> και για γνωρίσματα των οποίων οι τύποι δεδομένων υποστηρίζουν διάταξη μπορεί να είναι επίσης ένα από τα ακόλουθα: <, >, <=, >=.

Ας δούμε μερικά παραδείγματα της βασικής SFW ερώτησης χρησιμοποιώντας τη βάση δεδομένων για τις κινηματογραφικές ταινίες που ορίσαμε στην Εικόνα 5.2. Η Ερώτηση 5.1 δίνει ως αποτέλεσμα τον τίτλο και το έτος των ταινιών στις οποίες εμφανίστηκε ο ηθοποιός George Clooney. Παρατηρείστε ότι οι σταθερές συμβολοσειρές περικλείονται με απλά εισαγωγικά.

```
SELECT Title, Year
FROM PLAYS
WHERE Name = 'George Clooney';
```

Ερώτηση 5.1 Παράδειγμα της βασικής SFW δομής μιας ερώτησης SQL

Η συνθήκη της πρότασης SELECT μπορεί να περιέχει απλές αριθμητικές εκφράσεις. Για παράδειγμα, η Ερώτηση 5.2 μας δίνει τη διάρκεια μιας ταινίας σε ώρες.


```
SELECT Title, Year, Duration/60
FROM MOVIES;
```

Ερώτηση 5.2 Παράδειγμα της βασικής SFW δομής μιας ερώτησης SQL με μαθηματική έκφραση στην πρόταση SELECT

Σημειώστε ότι όπως φαίνεται και στην παραπάνω ερώτηση, η πρόταση του WHERE είναι προαιρετική. Όταν αυτή δεν υπάρχει σημαίνει ότι δεν καθορίζεται συνθήκη επιλογής και όλοι οι πιθανοί συνδυασμοί πλειάδων των σχέσεων στο FROM εμφανίζονται στο αποτέλεσμα.

Όπως είπαμε, μια βασική διαφορά της SQL από τη σχεσιακή άλγεβρα είναι ότι στην SQL δεν γίνεται απαλοιφή διπλότυπων. Θυμηθείτε ότι στη σχεσιακή άλγεβρα, μια σχέση ορίζεται ως ένα *σύνολο* από πλειάδες. Αντίθετα, η SQL θεωρεί μια σχέση ως *πολυσύνολο* (*multiset*) ή *bag* (*θύλακα*), δηλαδή, επιτρέπει πολλαπλές εμφανίσεις της ίδιας πλειάδας σε μια σχέση και ο αριθμός αυτών των εμφανίσεων έχει σημασία. Ας δούμε ένα παράδειγμα. Η Ερώτηση 5.3 μας δίνει τα πιθανά είδη ταινιών.

```
SELECT Genre
FROM MOVIE-GENRE;
```

Ερώτηση 5.3 Παράδειγμα σημασιολογίας πολυσυνόλου

Στην Εικόνα 5.6(α) βλέπετε το αποτέλεσμα της Ερώτησης 5.3 στο στιγμιότυπο της Εικόνας 3.5 και στην Εικόνα 5.6(β) το αποτέλεσμα της $\pi_{\text{Genre}}(\text{MOVIE-GENRE})$, που υποστηρίζει απαλοιφή διπλότυπων.

Genre
drama
science-fiction
crime
comedy
crime
romance
drama
thriller
music

(α)

Genre
drama
science-fiction
crime
comedy
romance
thriller
music

(β)

Εικόνα 5.6 Αποτέλεσμα της Ερώτησης 5.3 για το είδος ταινίας (α) χωρίς της χρήση DISTINCT, (β) με χρήση DISTINCT, ισοδύναμο με το $\pi_{\text{Genre}}(\text{MOVIE-GENRE})$

Η SQL δίνει τη δυνατότητα να επιλέξουμε τη σημασιολογία συνόλου με τη χρήση του διακριτικού DISTINCT μετά το SELECT. Η Ερώτηση 5.4 δίνει το αποτέλεσμα της Εικόνας 5.6(β).

```
SELECT DISTINCT Genre
FROM MOVIE-GENRE;
```

Ερώτηση 5.4 Παράδειγμα σημασιολογίας συνόλου

Ας δούμε τώρα πιο περίπλοκες ερωτήσεις που χρειάζονται συνένωση. Θεωρείστε ότι θέλουμε να βρούμε τα διαφορετικά είδη ταινιών στα οποία πρωταγωνιστεί ο ηθοποιός George Clooney. Για να απαντήσουμε σε αυτό το ερώτημα πρέπει να συνδυάσουμε πληροφορία από δύο πίνακες: τους πίνακες PLAYS και MOVIE-GENRE, όπως φαίνεται στην Ερώτηση 5.5.

```
SELECT DISTINCT Genre
FROM PLAYS, MOVIE-GENRE
WHERE Name = 'George Clooney' AND PLAYS.Title = MOVIE-GENRE.Title
AND PLAYS.Year = MOVIE-GENRE.Year;
```

Ερώτηση 5.5 Παράδειγμα με πολλούς πίνακες στην πρόταση FROM

Όπως φαίνεται στην Ερώτηση 5.5, μπορούμε να αναφερθούμε σε ένα γνώρισμα έχοντας ως πρόθεμα το όνομα της σχέσης στην οποία ανήκει, για παράδειγμα MOVIE-GENRE.Title. Αυτός ο συμβολισμός είναι υποχρεωτικός στην περίπτωση που στην ερώτηση συμμετέχουν γνωρίσματα με το ίδιο όνομα, έτσι ώστε να γίνεται διάκριση σε ποια σχέση αυτά ανήκουν.

Ας δούμε ακόμα ένα παράδειγμα. Αν θέλουμε να δούμε τις εθνικότητες των ηθοποιών που παίζουν στην ταινία Amour του 2012, θα συνδυάσουμε πληροφορία από τους πίνακες PLAYS και ACTOR όπως φαίνεται στην Ερώτηση 5.6.

```
SELECT DISTINCT Nationality
FROM PLAYS, ACTOR
WHERE Title = 'Amour' AND Year = 2012 AND PLAYS.Name = ACTOR.Name;
```

Ερώτηση 5.6 Παράδειγμα με πολλούς πίνακες στην πρόταση FROM

Τέλος, αν θέλουμε να ανακτήσουμε τις τιμές όλων των γνωρισμάτων των πλειάδων που επιλέγονται, μπορούμε να αντικαταστήσουμε τη λίστα των γνωρισμάτων στο SELECT με το σύμβολο '*'. Για παράδειγμα, η Ερώτηση 5.7 μας δίνει το περιεχόμενο του πίνακα ACTOR.

```
SELECT *
FROM ACTOR;
```

Ερώτηση 5.7 Παράδειγμα χρήσης ‘*’

Μετονομασία. Μπορούμε να μετονομάσουμε τα γνωρίσματα στο αποτέλεσμα μιας ερώτησης με χρήση του διακριτικού AS. Για παράδειγμα, στην Ερώτηση 5.2 που δίνει τη διάρκεια μιας ταινίας σε ώρες μπορούμε να δώσουμε όνομα στο τρίτο γνώρισμα, όπως φαίνεται στην Ερώτηση 5.8.

```
SELECT Title, Year, Duration/60 AS Hour-Duration
FROM MOVIES;
```

Ερώτηση 5.8 Παράδειγμα χρήσης του διακριτικού AS στην πρόταση του SELECT.

Το διακριτικό AS στη μετονομασία είναι προαιρετικό. Επίσης, μπορούμε να δώσουμε προσωρινά ονόματα στις σχέσεις που εμφανίζονται στο FROM. Αυτό είναι ιδιαίτερα χρήσιμο όταν για να απαντήσουμε σε ένα ερώτημα χρειάζεται να υπολογίσουμε το καρτεσιανό γινόμενο μιας σχέσης με τον εαυτό της. Ας δούμε ένα τέτοιο παράδειγμα. Θα θέλαμε να βρούμε τις ταινίες στις οποίες συμπρωταγωνιστεί ο George Clooney με τη Sandra Bullock. Ένας τρόπος για να διατυπώσουμε αυτό το ερώτημα είναι να πάρουμε το καρτεσιανό γινόμενο PLAYS x PLAYS και να επιλέξουμε τους συνδυασμούς πλειάδων P1 της πρώτης PLAYS με τις πλειάδες P2 της δεύτερης PLAYS που αφορούν την ίδια ταινία, απαιτώντας στην πρώτη πλειάδα το όνομα ηθοποιού είναι George Clooney και στη δεύτερη πλειάδα το όνομα ηθοποιού είναι Sandra Bullock. Αυτό μπορούμε να το εκφράσουμε με την Ερώτηση 5.9.

```
SELECT P1.Title, P1.Year
FROM PLAYS AS P1, PLAYS AS P2
WHERE P1.Title = P2.Title AND P1.Year = P2.Year AND
      P1.Actor = 'George Clooney' AND P2.Actor = 'Sandra Bullock';
```

Ερώτηση 5.9 Παράδειγμα μεταβλητών πλειάδων με χρήση του διακριτικού AS στην πρόταση του FROM.

Οι P1 και P2 ονομάζονται και *μεταβλητές πλειάδας (tuple variables)*. Ένας διαδικαστικός τρόπος για να κατανοήσετε τη σημασιολογία του παραπάνω ερωτήματος είναι και ο παρακάτω:

για κάθε πλειάδα P1 της PLAYS

για κάθε πλειάδα P2 της PLAYS

αν η συνθήκη του WHERE ισχύει στο (P1, P2)

δώσε στην έξοδο τα γνωρίσματα των P1, P2 που εμφανίζονται στο SELECT

Διάταξη των αποτελεσμάτων. Η SQL επιτρέπει στο χρήστη να διατάξει τις πλειάδες του αποτελέσματος μιας ερώτησης σύμφωνα με τις τιμές ενός ή περισσότερων γνωρισμάτων, χρησιμοποιώντας την πρόταση ORDER BY ακολουθούμενη από τα γνωρίσματα με βάση τα οποία θα γίνει η διάταξη. Ένα παράδειγμα φαίνεται στην Ερώτηση 5.10(α). Η προκαθορισμένη διάταξη είναι κατά αύξουσα σειρά τιμών. Άρα, το αποτέλεσμα της Ερώτησης 5.10(α) είναι οι ταινίες στις οποίες εμφανίζεται ο ηθοποιός George Clooney από την παλιότερη στην πιο πρόσφατη. Οι ταινίες της ίδιας χρονιάς διατάσσονται αλφαβητικά με βάση τον τίτλο τους.

```
SELECT Title, Year
FROM PLAYS
WHERE Actor = 'George Clooney'
ORDER BY Year, Title;
```

(α)

```
SELECT Title, Year
FROM PLAYS
WHERE Actor = 'George Clooney'
ORDER BY Year DESC, Title ASC
LIMIT 10;
```

(β)

Ερώτηση 5.10 Παράδειγμα διάταξης των αποτελεσμάτων

Αν θέλουμε τις τιμές σε φθίνουσα διάταξη μπορούμε να το προσδιορίσουμε με τη λέξη κλειδί DESC. Αντίστοιχα, η λέξη κλειδί ASC δηλώνει ρητά τη χρήση αύξουσας διάταξης. Για παράδειγμα, η Ερώτηση 5.10(β) διατάσει τις ταινίες ξεκινώντας από την πιο πρόσφατη και διατηρώντας την αύξουσα διάταξη με βάση τον τίτλο για τις ταινίες του ίδιου έτους.

Περιορισμός μεγέθους αποτελέσματος. Μπορούμε να περιορίσουμε το πλήθος των πλειάδων στο αποτέλεσμα μιας ερώτησης με χρήση του προσδιοριστικού LIMIT ακολουθούμενο από τον αριθμό των πλειάδων που θέλουμε να έχουμε στο αποτέλεσμα. Για παράδειγμα, η Ερώτηση 5.10(β) επιστρέφει μόνο τις 10 πρώτες πλειάδες.

Άσκηση Αυτοαξιολόγησης 5.3: SPJ Ερωτήματα

Θεωρείστε τη βάση δεδομένων για τις πιτσαρίες που ορίσατε στην προηγούμενη Άσκηση Αυτοαξιολόγησης 5.1 της οποίας ένα στιγμιότυπο μπορείτε να δείτε στην Εικόνα 3.12. Διατυπώστε ερωτήσεις σε SQL για τα ερωτήματα (α)-(στ) παρακάτω. Αν

θέλετε ανατρέξετε στις αντίστοιχες Ασκήσεις Αυτοαξιολόγησης στις οποίες διατυπώσαμε τα ίδια ερωτήματα σε σχεσιακή άλγεβρα, συγκεκριμένα στην Άσκηση Αυτοαξιολόγησης 3.3 για τα ερωτήματα (α)-(γ) και στην Άσκηση Αυτοαξιολόγησης 3.6 για τα ερωτήματα (δ)-(στ).

Ποια είναι:

- (α) τα ονόματα των πιτσών που περιέχουν μανιτάρι (mushroom),
- (β) τα συστατικά της πίτσας με όνομα Special,
- (γ) τα συστατικά που αρέσουν τουλάχιστον σε έναν φοιτητή,
- (δ) τα μαγαζιά που σερβίρουν (τουλάχιστον) μία πίτσα που περιέχει: μανιτάρι,
- (ε) τα μαγαζιά που σερβίρουν (τουλάχιστον) μια πίτσα που περιέχει τουλάχιστον ένα συστατικό που αρέσει στον Δημήτρη.
- (στ) τις πίτσες (τα ονόματα τους) που έχουν τουλάχιστον δυο διαφορετικά συστατικά.

5.2.2 Τελεστές Συνόλου

Η SQL υποστηρίζει την ένωση (UNION), τομή (INTERSECT) και διαφορά (EXCEPT) συνόλων. Η σύνταξη φαίνεται στην Εικόνα 5.7. Όπως και στη σχεσιακή άλγεβρα, οι πράξεις συνόλου εφαρμόζονται μόνο σε συμβατές μεταξύ τους σχέσεις, δηλαδή, πρέπει να υπάρχει μια-προς-μια ισοδυναμία των τύπων των γνωρισμάτων που εμφανίζονται στη λίστα του SELECT των δύο SFW ερωτήσεων.

```
(SELECT <attribute-list>
FROM <relation-list>
WHERE <condition>)
UNION [ALL] | INTERSECT [ALL] | EXCEPT
[ALL]
(SELECT <attribute-list>
FROM <relation-list>
WHERE <condition>)
```

Εικόνα 5.7 Πράξεις συνόλου στην SQL

Ειδικά για τις πράξεις συνόλου, η SQL υιοθετεί σημασιολογία συνόλου, δηλαδή, οι διπλότυπες πλειάδες απαλείφονται από το αποτέλεσμα.. Ως παράδειγμα δείτε τις Ερωτήσεις 5.11(α)-(γ) που αναφέρονται στα ίδια δύο βασικά SFW με διαφορετικές πράξεις συνόλου κάθε φορά. Τα αποτελέσματα τους στο στιγμιότυπο της Εικόνας 3.5

δίνονται στην Εικόνα 5.8. Αρχικά γίνεται απαλοιφή στα αποτελέσματα των δύο SFW (Εικόνα 5.8(α) και (β) αντίστοιχα) και μετά εκτελούνται οι πράξεις συνόλου όπως ακριβώς οι αντίστοιχοι τελεστές της σχεσιακής άλγεβρας με απαλοιφή διπλότυπων (στην ένωση) (Εικόνα 5.8(γ)-(ε) αντίστοιχα).

```
(SELECT Genre
FROM MOVIE-GENRE
WHERE Year >=2012)
UNION
(SELECT Genre
FROM MOVIE-GENRE
WHERE Year =2012);
(α)
```

```
(SELECT Genre
FROM MOVIE-GENRE
WHERE Year >=2012)
INTERSECT
(SELECT Genre
FROM MOVIE-GENRE
WHERE Year =2012);
(β)
```

```
(SELECT Genre
FROM MOVIE-GENRE
WHERE Year >=2012)
EXCEPT
(SELECT Genre
FROM MOVIE-GENRE
WHERE Year =2012);
(γ)
```

Ερώτηση 5.11 Παράδειγμα ερωτήσεων συνόλου: (α) ένωση, (β) τομή, (γ) διαφορά

Η SQL υποστηρίζει επίσης πράξεις πολυσυνόλου στις οποίες δε γίνεται απαλοιφή διπλότυπων, αλλά αντίθετα κάθε πλειάδα διπλότυπη ή όχι θεωρείται διαφορετική. Οι πράξεις πολυσυνόλου εκφράζονται με τη χρήση του διακριτικού ALL, δηλαδή UNION ALL, INTERSECT ALL και EXCEPT ALL. Ένας απλός τρόπος να κατανοήσετε τη σημασιολογία τους είναι να θεωρήσετε ότι κάθε πλειάδα ενός πίνακα σχετίζεται με ένα μετρητή που μετρά πόσες φορές εμφανίζεται η πλειάδα στον πίνακα. Έστω ότι μια πλειάδα εμφανίζεται στον ένα πίνακα m_1 φορές και στο άλλον πίνακα m_2 φορές, η πλειάδα θα εμφανίζεται $m_1 + m_2$ φορές στο UNION ALL των δύο πινάκων, $\min\{m_1, m_2\}$ φορές στο INTERSECT ALL των δύο πινάκων και $m_1 - m_2$ φορές, αν $m_1 > m_2$ ή καθόλου αλλιώς, στο EXCEPT ALL του πρώτου πίνακα από τον δεύτερο πίνακα.

Genre	Genre	Genre	Genre	Genre	Genre	Genre	Genre
drama	drama	drama	drama	science-fiction	drama	drama	science-fiction
romance	romance	romance	romance	comedy	romance	romance	comedy
science-fiction		science-fiction		thriller	science-fiction		drama
comedy		comedy			comedy		thriller
drama		drama			drama		
thriller		thriller			thriller		
					drama		
					romance		

(α) (β) (γ) (δ) (ε) (στ) (ζ) (η)

Εικόνα 5.8 Αποτελέσματα ερωτήσεων συνόλου στο στιγμιότυπο της Εικόνας 3.5: (α) αποτέλεσμα του πρώτου SFW των Ερωτήσεων 5.11, (β) αποτέλεσμα του δεύτερου SFW των

Ερωτήσεων 5.11, (γ) αποτέλεσμα του UNION (Ερώτηση 5.11(α)), (δ) αποτέλεσμα του INTERSECT (Ερώτηση 5.11(β)), (ε) αποτέλεσμα του EXCEPT (Ερώτηση 5.11(γ)), (στ) αποτέλεσμα αν θεωρήσουμε UNION ALL αντί UNION στην Ερώτηση 5.11(α), (ζ) αποτέλεσμα αν θεωρήσουμε INTERSECT ALL αντί INTERSECT στην Ερώτηση 5.11(β), (η) αποτέλεσμα αν θεωρήσουμε EXCEPT ALL αντί EXCEPT στην Ερώτηση 5.11(γ)

Στην Εικόνα 5.8(στ)-(ζ), μπορείτε να δείτε τα αποτελέσματα των πράξεων πολυσυνόλου για τις Ερωτήσεις 5.11(α)-(γ) αν αντικαταστήσουμε τις πράξεις συνόλου με την ALL εκδοχή τους. Ας δούμε ακόμα ένα παράδειγμα. Θεωρείστε τους πίνακες BAG1 και BAG2 που δίνονται στην Εικόνα 5.9(α) και 5.9(β). Στην Εικόνα 5.9 βλέπετε τα αποτελέσματα των Ερωτήσεων 5.12(α)-(γ).

```
(SELECT *
FROM BAG1)
UNION ALL
(SELECT *
FROM BAG2);
(α)
```

```
(SELECT *
FROM BAG1)
INTERSECT ALL
(SELECT *
FROM BAG2);
(β)
```

```
(SELECT *
FROM BAG1)
EXCEPT ALL
(SELECT *
FROM BAG2);
(γ)
```

Ερώτηση 5.12 Παράδειγμα ερωτήσεων πολυσυνόλου: (α) ένωση, (β) τομή, (γ) διαφορά

BAG1	BAG2			
Fruit	Fruit	Fruit	Fruit	Fruit
apple	apple	apple	apple	apple
apple	orange	apple	orange	
orange	orange	orange		
orange		orange		
		apple		
		orange		
		orange		
(α)	(β)	(γ)	(δ)	(ε)

Εικόνα 5.9 (α) Ο πίνακας BAG1 (β) ο πίνακας BAG2 και (γ)-(ε) αποτελέσματα των Ερωτήσεων 5.12(α)-(γ) αντίστοιχα, δηλαδή, (γ) UNION ALL, (δ) INTERSECT ALL, (ε) EXCEPT ALL

Άσκηση Αυτοαξιολόγησης 5.4: Ερωτήσεις SQL

Θεωρείστε τη βάση δεδομένων για τις πίτσες που ορίσατε στην προηγούμενη Άσκηση Αυτοαξιολόγησης 5.1 της οποίας ένα στιγμιότυπο μπορείτε να δείτε στην Εικόνα 3.12. Διατυπώστε ερωτήσεις σε SQL για τα ερωτήματα (α)–(δ) παρακάτω. Αν θέλετε ανατρέξτε στην Άσκηση Αυτοαξιολόγησης 3.4 στην οποία διατυπώσαμε τα ίδια ερωτήματα σε σχεσιακή άλγεβρα. Ποιες πίτσες (τα ονόματα τους):

- (α) δεν περιέχουν μανιτάρι,
- (β) περιέχουν μανιτάρι ή ζαμπόν (ham),
- (γ) περιέχουν μανιτάρι και ζαμπόν,
- (δ) περιέχουν μανιτάρι αλλά όχι ζαμπόν.

Άσκηση Αυτοαξιολόγησης 5.5: Σημασιολογία Πολυσυνόλου

Θεωρίστε την παρακάτω ερώτηση (Ερώτηση 5.12) που αναφέρεται στον πίνακα FOLLOWS(FOLLOWER, FOLLOWEE) της Άσκησης Αυτοαξιολόγησης 5.1(β).

```
(SELECT FOLLOWER  
FROM FOLLOWS)  
EXCEPT ALL  
(SELECT FOLLOWEE  
FROM FOLLOWS);
```

Ερώτηση 5.13 Ερώτηση SQL για την Άσκηση Αυτοαξιολόγησης 5.5

Μπορείτε να πείτε με απλά λόγια ποιοι χρήστες θα εμφανίζονται στο αποτέλεσμα; Υπόδειξη: σκεφτείτε την έννοια των μετρητών για τη σημασιολογία των πολυσυνόλων που συζητήσαμε σε αυτήν την ενότητα.

5.2.3 Εμφωλευμένες Ερωτήσεις

Εμφωλευμένες ερωτήσεις (nested queries) ή υπό-ερωτήσεις (subqueries) είναι ερωτήσεις που εμφανίζονται μέσα σε άλλες ερωτήσεις. Αυτές περιλαμβάνουν ερωτήσεις που εμφανίζονται ως ένας από τους πίνακες στο FROM και ερωτήσεις που εμφανίζονται ως συνθήκες στο WHERE. Τέλος, οι ερωτήσεις που επιστρέφουν μια μόνο γραμμή, καλούνται βαθμωτές (scalar) και μπορούν να εμφανιστούν όπου μπορεί να εμφανιστεί μια τιμή. Στη συνέχεια, περιγράφουμε αναλυτικά τα παραπάνω.

Εμφωλευμένες ερωτήσεις στο FROM. Το αποτέλεσμα μιας ερώτησης είναι ένας πίνακας και μπορεί να χρησιμοποιηθεί ως ένας από τους πίνακες στο FROM. Για παράδειγμα, η Ερώτηση 5.14 δίνει ως αποτέλεσμα τα ονόματα των ηθοποιών που έπαιξαν σε ταινίες που έχουν ως είδος και musical και drama. Ας δούμε γιατί. Το αποτέλεσμα της ερώτησης στο FROM είναι ένας πίνακας M με τις ταινίες που έχουν είδος, musical και drama. Μέσω της συνένωσης του πίνακα M με τον πίνακα PLAYS παίρνουμε τα ονόματα των ηθοποιών που παίζουν σε αυτές τις ταινίες.


```

SELECT P.Name
FROM PLAYS as P,
    (SELECT Year, Title
     FROM MOVIE-GENRE
     WHERE Genre = 'musical')
INTERSECT
(SELECT Year, Title
 FROM MOVIE-GENRE
 WHERE Genre = 'drama')
as M
WHERE P.Year = M.Year and P.Title = M.Title;

```

Ερώτηση 5.14 Παράδειγμα ερώτησης εμφωλευμένης στο FROM

Εμφωλευμένες ερωτήσεις ως συνθήκη. Μπορούμε να έχουμε SFW ερωτήσεις στις συνθήκες του WHERE, όπως φαίνεται στην Εικόνα 5.10, όπου <X> είναι τελεστές που θα εξετάσουμε στη συνέχεια. Η SFW ερώτηση στη συνθήκη του WHERE καλείται *εσωτερική*, ενώ η ερώτηση της οποίας μέρος είναι το WHERE καλείται *εξωτερική*. Η εσωτερική ερώτηση αποτιμάται μια φορά για κάθε πλειάδα της εξωτερικής ερώτησης. Στη συνέχεια, μελετάμε τα διαφορετικά είδη των τελεστών.

```

SELECT <attribute-list> (εξωτερική ερώτηση)
FROM <relation-list>
WHERE <X> (SELECT <attribute-list> (εσωτερική ερώτηση)
          FROM <relation-list>
          WHERE <condition>)

```

Εικόνα 5. 10 Γενική μορφή μιας ερώτησης εμφωλευμένης στην πρόταση WHERE μιας άλλης ερώτησης

Ο τελεστής in (not in). Ο τελεστής IN (NOT IN) εξετάζει αν μια πλειάδα ανήκει στο αποτέλεσμα της εσωτερικής ερώτησης και επιστρέφει TRUE (FALSE) αν η πλειάδα ανήκει και FALSE (TRUE) αλλιώς.

Η Ερώτηση 5.15(α) είναι ένα παράδειγμα χρήσης του IN και η Ερώτηση 5.15(β) ένα παράδειγμα χρήσης του NOT IN. Η Ερώτηση 5.15(α) επιστρέφει τις ταινίες για τις οποίες υπάρχει ταινία με τον ίδιο τίτλο η οποία γυρίστηκε αργότερα. Η εσωτερική ερώτηση είναι μια *συσχετισμένη (correlated)* υπό-ερώτηση που αναφέρεται σε μια πλειάδα της εξωτερικής ερώτησης. Η εσωτερική υπό-ερώτηση εκτιμάται για κάθε μία πλειάδα της εξωτερικής υπό-ερώτησης. Η Ερώτηση 5.15(β) επιστρέφει τις ταινίες που δεν εμφανίζονται στον πίνακα PLAYS, δηλαδή τις ταινίες χωρίς ηθοποιούς.

```

SELECT DISTINCT M1.Title, M1.Year
FROM MOVIE AS M1
WHERE M1.Title IN
    (SELECT M2.Title
     FROM MOVIE AS M2
     WHERE M2.Year > M1.Year);
    (α)

```

```

SELECT M.Title, M.Year
FROM MOVIES AS M
WHERE (M.Title, M.Year) NOT IN
    (SELECT Title, Year
     FROM PLAYS);
    (β)

```

Ερώτηση 5.15. Παράδειγμα (α) in υπό-ερώτησης και (β) not in υπό-ερώτησης

Ο τελεστής exists (not exists). Ο τελεστής EXISTS (NOT EXISTS) εξετάζει αν το αποτέλεσμα της εσωτερικής ερώτησης είναι ή όχι κενό και επιστρέφει TRUE (FALSE) αν το αποτέλεσμα δεν είναι κενό και FALSE (TRUE) αλλιώς.

Η Ερώτηση 5.16(α) δίνει ένα παράδειγμα χρήσης του EXISTS και η Ερώτηση 5.16(β) ένα παράδειγμα χρήσης του NOT EXISTS. Η Ερώτηση 5.16(α) επιστρέφει το ίδιο αποτέλεσμα με την Ερώτηση 5.15(α), δηλαδή τις ταινίες για τις οποίες υπάρχει ταινία με τον ίδιο τίτλο η οποία γυρίστηκε αργότερα από αυτές.

Η Ερώτηση 5.16(β) είναι πιο περίπλοκη: επιστρέφει τους ηθοποιούς που έχουν παίξει σε όλες τις ταινίες που έχει παίξει ο ηθοποιός George Clooney. Είναι η ερώτηση που είδαμε πως μπορεί να εκφραστεί στη σχεσιακή άλγεβρα με την πράξη της διαίρεσης ή εναλλακτικά με χρήση αφαίρεσης στην Ενότητα 3.3.6. Ας τη δούμε αναλυτικά. Η εσωτερική ερώτηση είναι μια αφαίρεση δυο πινάκων. Ο πρώτος πίνακας περιέχει τις ταινίες στις οποίες έπαιξε ο ηθοποιός George Clooney. Ο δεύτερος πίνακας περιέχει τις ταινίες στις οποίες έπαιξε ο υπό εξέταση ηθοποιός P1. Προσέξτε ότι πρόκειται πάλι για μια συσχετισμένη υπό-ερώτηση, δηλαδή για υπό-ερώτηση που αναφέρεται σε πλειάδα της εξωτερικής ερώτησης. Το αποτέλεσμα της αφαίρεσης επιστρέφει μη κενό σύνολο μόνο αν υπάρχει ταινία στην οποία έπαιξε ο ηθοποιός George Clooney αλλά όχι ο υπό-εξέταση ηθοποιός.

```

SELECT DISTINCT M1.Title, M1.Year
FROM MOVIE AS M1
WHERE EXISTS
    (SELECT *
     FROM MOVIE AS M2
     WHERE M2.Title = M1.Title
     AND M2.Year > M1.Year);
    (α)

```

```

SELECT DISTINCT P1.Name
FROM PLAYS as P1
WHERE NOT EXISTS
    ((SELECT Title, Year
     FROM PLAYS
     WHERE Name = 'George Clooney')
    EXCEPT
    (SELECT P2.Title, P2.Year
     FROM PLAYS as P2
     WHERE P2.Name = P1.Name));
    (β)

```

Ερώτηση 5.16 Παράδειγμα (α) exists υπό-ερώτησης και (β) not exists υπό-ερώτησης

Ο τελεστής unique (not unique). Ο τελεστής UNIQUE (NOT UNIQUE) εξετάζει αν το αποτέλεσμα της εσωτερικής ερώτησης έχει διπλότυπα ή όχι και επιστρέφει TRUE (FALSE) αν το αποτέλεσμα δεν έχει διπλότυπα και FALSE (TRUE) αλλιώς.

Η Ερώτηση 5.17(α) είναι ένα παράδειγμα χρήσης του UNIQUE και η Ερώτηση 5.17(β) ένα παράδειγμα χρήσης του NOT UNIQUE. Η Ερώτηση 5.17(α) επιστρέφει τα ονόματα των ηθοποιών που έπαιξαν ακριβώς σε μια ταινία. Για να το δείτε, σκεφτείτε ότι στην εσωτερική υπό-ερώτηση, το όνομα ενός ηθοποιού εμφανίζεται τόσες φορές όσες οι ταινίες στις οποίες έχει παίξει. Γενικά, μια UNIQUE/NOT UNIQUE υπό-ερώτηση μπορεί να χρησιμοποιηθεί για να ελέγξουμε αν το αποτέλεσμα μιας ερώτησης είναι ή όχι πολυσύνολο. Η Ερώτηση 5.17(β) επιστρέφει τις ταινίες που έχουν τουλάχιστον δύο διαφορετικά είδη.

```
SELECT P1.Name
FROM PLAYS AS P1
WHERE UNIQUE
  (SELECT P2.Name
   FROM PLAYS AS P2
   WHERE P2.Name = P1.Name);
```

(α)

```
SELECT DISTINCT G1Title, G1.Year
FROM MOVIE-GENRE AS G1
WHERE NOT UNIQUE
  (SELECT Genre
   FROM MOVIE-GENRE AS G2
   WHERE G1.Title = G2.Title AND
         G1.Year = G2.Year);
```

(β)

Ερώτηση 5.17 Παράδειγμα (α) unique υπό-ερώτησης και (β) not unique υπό-ερώτησης

Υπό-ερωτήσεις με ποσοδείκτες. Οι επόμενοι τελεστές επιτρέπουν τη σύγκριση μιας τιμής με τις τιμές που ανήκουν στο αποτέλεσμα της υπό-ερώτησης. Οι υπό-επερωτήσεις αυτές μπορούν να εμφανίζονται όπου εμφανίζεται μια τιμή.

Ο τελεστής *<op>* ALL, όπου *op* είναι ένας από τους τελεστές σύγκρισης {=, <=, <, >, >=, <> }, έχει τη σημασία του καθολικού ποσοδείκτη και επιστρέφει TRUE αν η σύγκριση ισχύει για όλες τις τιμές του πίνακα που επιστρέφει η υπό-ερώτηση.

Ο τελεστής *<op>* ANY, όπου *op* είναι ένας από τους τελεστές σύγκρισης {=, <=, <, >, >=, <> }, έχει τη σημασία του υπαρξιακού ποσοδείκτη και επιστρέφει TRUE αν η

σύγκριση ισχύει τουλάχιστον για μια από τις τιμές του πίνακα που επιστρέφει η υπό-ερώτηση.

Η Ερώτηση 5.18(α) είναι ένα παράδειγμα χρήσης του ALL και η Ερώτηση 5.18(β) ένα παράδειγμα χρήσης του ANY. Η Ερώτηση 5.18(α) επιστρέφει για κάθε ηθοποιό το έτος που έπαιξε τελευταία φορά σε κάποια ταινία. Η Ερώτηση 5.18(β) επιστρέφει τις ταινίες που έχουν τουλάχιστον ένα είδος ίδιο με τα είδη των ταινιών που γυρίστηκαν το 2000.

```
SELECT DISTINCT P1.Name, P1.Year
FROM PLAYS AS P1
WHERE P1.Year >= ALL
  (SELECT P2.Year
   FROM PLAYS AS P2
   WHERE P2.Name = P1.Name);
```

(α)

```
SELECT DISTINCT G1.Title, G1.Year
FROM MOVIE-GENRE AS G1
WHERE G1.Genre = ANY
  (SELECT Genre
   FROM MOVIE-GENRE
   WHERE Year = 2000);
```

(β)

Ερώτηση 5.18 Παράδειγμα υπό-ερώτησης με (α) any και (β) all

Βαθμωτές υπό-ερωτήσεις. Ειδικά, οι βαθμωτές ερωτήσεις, δηλαδή οι ερωτήσεις που επιστρέφουν μόνο μια πλειάδα ως αποτέλεσμα, μπορεί να χρησιμοποιηθούν σε μια άλλη SQL ερώτηση όπως μια πλειάδα. Η εσωτερική ερώτηση στην Ερώτηση 5.19 είναι ένα τέτοια παράδειγμα: επιστρέφει την εθνικότητα του ηθοποιού George Clooney. Συνολικά, η Ερώτηση 5.19 επιστρέφει τα ονόματα των ηθοποιών που έχουν την ίδια εθνικότητα με τον ηθοποιό George Clooney. Αν μια ερώτηση που χρησιμοποιείται ως βαθμωτή επιστρέφει παραπάνω από μια τιμές, τότε προκύπτει λάθος εκτέλεσης, ενώ αν η ερώτηση δεν επιστρέφει καμία γραμμή, θεωρείται ότι επιστρέφει την τιμή null.

```
SELECT DISTINCT Name
FROM ACTOR
WHERE Nationality = (SELECT Nationality
                    FROM ACTOR
                    WHERE Name = 'George Clooney');
```

Ερώτηση 5.19 Παράδειγμα βαθμωτής ερώτησης

Άσκηση Αυτοαξιολόγησης 5.6: Εμφωλευμένες Ερωτήσεις

Θεωρείστε τη βάση δεδομένων για τις ταινίες που ορίσαμε στην Εικόνα 5.2. Δώστε ερωτήσεις SQL που δίνουν ως αποτέλεσμα τις πιο πρόσφατες ταινίες με είδος κωμωδία χρησιμοποιώντας (α) (not) exists, (β) (not) in, (γ) all και (δ) any.

5.2.4 Συναθροίσεις

Η SQL παρέχει ένα σύνολο από ενσωματωμένες συναρτήσεις για τον υπολογισμό συναθροιστικών τιμών. Συγκεκριμένα, η συνάρτηση COUNT() επιστρέφει το πλήθος των τιμών του γνωρίσματος που παίρνει ως όρισμα. Οι συναρτήσεις AVG() και SUM() εφαρμόζονται σε γνωρίσματα με αριθμητικό πεδίο ορισμού και επιστρέφουν αντίστοιχα το μέσο όρο και το άθροισμα των τιμών τους. Τέλος, οι συναρτήσεις MIN() και MAX() εφαρμόζονται σε γνωρίσματα των οποίων το πεδίο ορισμού είναι ολικώς διατεταγμένο και επιστρέφουν αντίστοιχα τη μικρότερη και μεγαλύτερη τιμή τους. Οι συναρτήσεις θεωρούν τα γνωρίσματα τους ως πολυσύνολα. Αν θέλουμε να γίνει απαλοιφή των διπλότυπων πριν τον υπολογισμό των συναρτήσεων, θα πρέπει να χρησιμοποιήσουμε το διακριτικό DISTINCT. Τέλος, η COUNT(*) επιστρέφει το πλήθος των γραμμών του αποτελέσματος.

Οι συναθροιστικές συναρτήσεις μπορούν να χρησιμοποιηθούν στο SELECT μιας ερώτησης. Όταν χρησιμοποιούμε συναθροιστικές συναρτήσεις δεν μπορούμε να έχουμε άλλα γνωρίσματα στο SELECT. Ως παραδείγματα χρήσης των συναθροιστικών συναρτήσεων, δείτε τις Ερωτήσεις 5.20(α)-(δ). Η Ερώτηση 5.20(α) επιστρέφει τη μέγιστη, ελάχιστη και μέση διάρκεια των ταινιών που γυρίστηκαν μετά το 2000. Για παράδειγμα, η Ερώτηση 5.20(α) αν εφαρμοστεί στο στιγμιότυπο της Εικόνας 3.5, θα έχει ως αποτέλεσμα τις τιμές 117, 91 και 106,5. Η Ερώτηση 5.20(β) επιστρέφει το πλήθος των ειδών που εμφανίζονται στον πίνακα MOVIE-GENRE, χωρίς απαλοιφή διπλότυπων. Για παράδειγμα, η Ερώτηση 5.20(β), αν εφαρμοστεί στο στιγμιότυπο της Εικόνας 3.5, επιστρέφει την τιμή 9. Τέλος, οι Ερωτήσεις 5.18(γ) και 5.18(δ) είναι ισοδύναμες και επιστρέφουν και οι δύο το πλήθος των διαφορετικών ειδών που εμφανίζονται στον πίνακα MOVIE-GENRE. Για παράδειγμα, την τιμή 7 για το στιγμιότυπο της Εικόνας 3.5.

```
SELECT MAX(Runtime), MIN(Runtime), AVG(Runtime)
FROM MOVIE
WHERE Year > 2000;
```

(α)

```
SELECT COUNT(GENRE) FROM MOVIE-GENRE;
```

(β)
SELECT COUNT(DISTINCT GENRE)
FROM MOVIE-GENRE;

SELECT COUNT(*)
FROM (SELECT DISTINCT GENRE
FROM MOVIE-GENRE);

(γ)

(δ)

Ερώτηση 5.20 Παραδείγματα χρήσης συναθροιστικών συναρτήσεων και της σημασίας των πολυσυνόλων

Τέλος, η Ερώτηση 5.21 μας δίνει τις πιο πρόσφατες κωμωδίες. Θυμηθείτε ότι όταν έχουμε συναθροιστικές συναρτήσεις, στην πρόταση SELECT δε μπορούμε να έχουμε άλλα γνωρίσματα. Για αυτό το λόγο, χρειαζόμαστε την εξωτερική ερώτηση για να βρούμε το τίτλο των πιο πρόσφατων ταινιών. Επειδή, η εσωτερική ερώτηση είναι βαθμωτή χρησιμοποιήσαμε τον τελεστή = για τη σύγκριση (δηλαδή, τη θεωρήσαμε ως τιμή). Εναλλακτικά, στη θέση του = θα μπορούσαμε επίσης να χρησιμοποιήσουμε τον τελεστή IN ή και >=ALL.

```
SELECT Title, Year  
FROM MOVIE-GENRE  
WHERE Genre = 'comedy' AND Year = (SELECT MAX(Year)  
FROM MOVIE-GENRE  
WHERE Genre = 'comedy');
```

Ερώτηση 5.21 Παράδειγμα χρήσης συναθροιστικών συναρτήσεων και ανάκτησης της πλειάδας με τη μικρότερη τιμή

Ομαδοποίηση. Συχνά, θα θέλαμε να εφαρμόσουμε τις συναθροιστικές συναρτήσεις, σε ομάδες πλειάδων μιας σχέσης με βάση τις τιμές κάποιων γνωρισμάτων της σχέσης. Για παράδειγμα, θα μας ενδιέφερε το πλήθος των ταινιών όχι μόνο συνολικά αλλά και ανά έτος. Τα γνωρίσματα με βάση την τιμή των οποίων γίνεται η ομαδοποίηση (δηλαδή, το έτος στο προηγούμενο παράδειγμα) καλούνται *γνωρίσματα ομαδοποίησης (grouping attributes)*.

Η SQL δίνει τη δυνατότητα να δημιουργηθούν ομάδες με χρήση της πρότασης GROYP BY ακολουθούμενης από τα γνωρίσματα ομαδοποίησης, όπως φαίνεται στις Ερωτήσεις 5.22(α)-(β). Όταν σε μία ερώτηση υπάρχει πρόταση GROYP BY, οι πλειάδες χωρίζονται σε ομάδες, έτσι ώστε οι πλειάδες σε κάθε ομάδα να έχουν την ίδια τιμή στα γνωρίσματα ομαδοποίησης και στη συνέχεια οι συναθροιστικές συναρτήσεις υπολογίζονται σε κάθε ομάδα ξεχωριστά. Τα γνωρίσματα ομαδοποίησης

εμφανίζονται και στην πρόταση SELECT, έτσι ώστε κάθε συναθροιστική τιμή να εμφανίζεται μαζί με την τιμή των γνωρισμάτων της ομάδας στην οποία αυτή αντιστοιχεί. Στο αποτέλεσμα της ερώτησης υπάρχει μια πλειάδα για κάθε ομάδα.

```
SELECT Year, COUNT(*)
FROM MOVIE
GROUP BY Year;
```

(α)

```
SELECT Title, Year, COUNT(*)
FROM PLAYS, ACTOR
WHERE ACTOR.Name = PLAYS.Name AND
ACTOR.Sex = 'Female'
GROUP BY Title, Year;
```

(β)

Ερώτηση 5.22 Παραδείγματα χρήσης συναθροιστικών συναρτήσεων με ομαδοποίηση

Για παράδειγμα, στην Ερώτηση 5.22(α), οι πλειάδες της σχέσης MOVIE χωρίζονται σε ομάδες με βάση το έτος και εμφανίζεται το πλήθος των ταινιών ανά έτος. Το αποτέλεσμα της ερώτησης στο στιγμιότυπο της Εικόνας 3.2, δίνεται στην Εικόνα 5.11(α). Το GROUP BY ακολουθεί την πρόταση WHERE (αν αυτή υπάρχει) και η ομαδοποίηση εφαρμόζεται στο αποτέλεσμα. Για παράδειγμα, στην Ερώτηση 5.22(β), συνενώνουμε τις ταινίες με τους ηθοποιούς, έτσι ώστε να έχουμε μόνο τις γυναίκες ηθοποιούς, στη συνέχεια δημιουργούμε ομάδες, μία ανά ταινία και τέλος εμφανίζουμε για κάθε ταινία με γυναίκες ηθοποιούς, το πλήθος των γυναικών ηθοποιών που παίζουν σε αυτές. Το αποτέλεσμα της ερώτησης στο στιγμιότυπο της Εικόνας 3.2, δίνεται στην Εικόνα 5.11(β).

Year	Count
2012	1
2013	2
2001	1
1960	1

(α)

Title	Year	Count
Gravity	2013	1
Amour	2012	1

(β)

Εικόνα 5.11 Αποτέλεσμα (α) της Ερώτησης 5.20(α) και (β) της Ερώτησης 5.20(β) στο στιγμιότυπο της Εικόνας 3.2

Υπάρχουν περιπτώσεις που θα θέλαμε να υπολογίσουμε τις συναρτήσεις μόνο σε ομάδες που ικανοποιούν συγκεκριμένες συνθήκες. Η SQL δίνει αυτή τη δυνατότητα με χρήση της πρότασης HAVING σε συνδυασμό με την GROUP BY. Η HAVING προσδιορίζει μια συνθήκη πάνω στις ομάδες, έτσι ώστε μόνο οι ομάδες που ικανοποιούν αυτή τη συνθήκη να εμφανίζονται στο αποτέλεσμα. Για παράδειγμα, στην Ερώτηση 5.23(α), η συνθήκη στο HAVING απαιτεί κάθε ομάδα να έχει παραπάνω από μια ταινία. Το αποτέλεσμα της Ερώτησης 5.23(α) στο στιγμιότυπο της Εικόνας 3.2 είναι (2013, 2), γιατί μόνο η ομάδα με τιμή 2013 ικανοποιεί τη συνθήκη. Προσέξτε ότι η συνθήκη του HAVING εφαρμόζεται στις ομάδες και όχι ατομικά σε κάθε πλειάδα.

<pre>SELECT Year, COUNT(*) FROM MOVIE GROUP BY Year HAVING Count(*) > 1;</pre>	<pre>SELECT ACTOR.Name, Year, Count() FROM ACTOR, PLAYS WHERE ACTOR.Name = PLAYS.Name AND Year > 2010 GROUP BY ACTOR.Name, Year HAVING COUNT(*) > 3;</pre>
(α)	(β)

Ερώτηση 5.23 Παραδείγματα χρήσης συναθροιστικών συναρτήσεων με ομαδοποίηση και προσδιορισμό συνθήκης στις ομάδες

Ας δούμε τώρα τη σειρά αποτίμησης μια SQL ερώτησης. Πρώτα εκτελείται το καρτεσιανό γινόμενο των σχέσεων στο FROM, και στη συνέχεια επιλέγονται οι πλειάδες που ικανοποιούν τη συνθήκη του WHERE. Οι πλειάδες αυτές ομαδοποιούνται με βάση τις τιμές τους στα γνωρίσματα ομαδοποίησης. Από τις ομάδες επιλέγονται αυτές που ικανοποιούν τη συνθήκη του HAVING. Τέλος, υπολογίζονται και προβάλλονται οι συναθροιστικές συναρτήσεις στις επιλεγμένες ομάδες. Για παράδειγμα, η Ερώτηση 5.23(β) μας δίνει ως αποτέλεσμα για κάθε ηθοποιό για κάθε έτος μετά το 2010 στο οποίο ο ηθοποιός έχει παίξει σε περισσότερες από 3 ταινίες, το πλήθος των ταινιών στις οποίες αυτός έχει παίξει.

Άσκηση Αυτοαξιολόγησης 5.7: Συναθροιστικές Συναρτήσεις

Δώστε το αποτέλεσμα των παρακάτω ερωτήσεων στο στιγμιότυπο της Εικόνας 5.4:

(α) SELECT SUM(B) FROM R;

(β) SELECT SUM(DISTINCT B) FROM R;

(γ) `SELECT D, SUM(A) FROM R, S WHERE B = C GROUP BY D HAVING COUNT(*) > 1;`

Άσκηση Αυτοαξιολόγησης 5.8: Ερωτήσεις SQL

Θεωρείστε τη σχεσιακή βάση δεδομένων για τις πίτσες που ορίσατε στην Άσκηση Αυτοαξιολόγησης 5.1. Διατυπώστε ερωτήσεις SQL για τα ερωτήματα (α)–(γ) παρακάτω.

(α) Το όνομα του μαγαζιού που σερβίρει την φτηνότερη πίτσα Special.

(β) Για κάθε πίτσα που περιέχει μανιτάρι και σερβίρεται σε τουλάχιστον δύο μαγαζιά τη μέση τιμή της.

(γ) Για κάθε φοιτητή, την πιτσαρία που σερβίρει την πιο φτηνή πίτσα ανάμεσα σε αυτές που περιέχουν τουλάχιστον ένα συστατικό που του αρέσει.

5.2.5 Η τιμή null

Όπως είδαμε και στο Κεφάλαιο 3.1, η τιμή NULL είναι μια ειδική τιμή που μπορεί να ανατεθεί σε ένα γνώρισμα, που μπορεί να σημαίνει ότι η τιμή του γνωρίσματος δεν είναι γνωστή ή διαθέσιμη ή ότι δεν ορίζεται. Για παράδειγμα, στην πλειάδα (Ryan Gosling, 1980-11-12, Male, NULL) της σχέσης ACTOR, η τιμή NULL στο γνώρισμα Nationality του ηθοποιού Ryan Gosling, σημαίνει ότι η τιμή για την εθνικότητα του συγκεκριμένου ηθοποιού δεν είναι γνωστή. Στην ενότητα αυτή, θα εξετάσουμε πως η SQL χειρίζεται την τιμή NULL.

Αριθμητικές πράξεις που περιλαμβάνουν την τιμή NULL έχουν ως αποτέλεσμα NULL, ενώ οι συναθροιστικές συναρτήσεις εκτός της COUNT(*) αγνοούν τις τιμές NULL. Η SQL χρησιμοποιεί λογική τριών τιμών: TRUE, FALSE και UNKNOWN. Στον Πίνακα 5.1, βλέπετε τα αποτελέσματα των λογικών τελεστών AND, OR και NOT, στην περίπτωση της λογικής τριών τιμών. Οι προτάσεις WHERE και HAVING επιλέγουν πλειάδες μόνο αν το αποτέλεσμα της συνθήκης είναι TRUE.

Όταν συγκρίνουμε μια τιμή με την τιμή NULL, το αποτέλεσμα της σύγκρισης είναι UNKNOWN. Αν θέλουμε να ελέγξουμε αν μια τιμή, είναι ίση ή όχι με την τιμή NULL, πρέπει να χρησιμοποιήσουμε IS NULL και IS NOT NULL αντίστοιχα.

Πίνακας 5. 1 Λογική τριών τιμών: FALSE (F), TRUE (T), UNKONWN (U)

A AND B		B		
		F	U	T
A	F	F	F	F
	U	F	U	U
	T	F	U	T

A OR B		B		
		F	U	T
A	F	F	U	T
	U	U	U	T
	T	T	T	T

A	NOT A
F	T
T	F
U	U

5.2.6 Συνενώσεις

Η SQL υποστηρίζει συνενώσεις πινάκων με χρήση του διακριτικού JOIN (ή INNER JOIN) στο FROM σε συνδυασμό με το διακριτικό ON ακολουθούμενο από τη συνθήκη συνένωσης. Η χρήση συνένωσης προσφέρει έναν εναλλακτικό τρόπο διατύπωσης ερωτημάτων. Για παράδειγμα, η Ερώτηση 5.24 αποτελεί αναδιατύπωση της Ερώτησης 5.5, όπου οι συνθήκες συνένωσης εμφανίζονται ως συνθήκες στο ON και όχι ως συνθήκες στο WHERE.

```
SELECT DISTINCT Genre
FROM PLAYS JOIN MOVIE-GENRE ON PLAYS.Title = MOVIE-GENRE.Title
AND PLAYS.Year = MOVIE-GENRE.Year;
WHERE Name = 'George Clooney';
```

Ερώτηση 5.24 Παράδειγμα συνένωσης

Η SQL υποστηρίζει επίσης τη φυσική συνένωση (δείτε και την αντίστοιχη Ενότητα 3.3.5 της σχεσιακής άλγεβρας) με χρήση του διακριτικού NATURAL JOIN. Στην περίπτωση αυτή, δε χρησιμοποιείται το ON, αλλά υπονοείται μια συνθήκη ισότητας στα γνωρίσματα των δύο πινάκων που έχουν το ίδιο όνομα. Η Ερώτηση 5.25 αποτελεί αναδιατύπωση της Ερώτησης 5.24 με χρήση φυσικής συνένωσης.

```
SELECT DISTINCT Genre
FROM PLAYS NATURAL JOIN MOVIE-GENRE
WHERE Name = 'George Clooney';
```

Ερώτηση 5.25 Παράδειγμα φυσικής συνένωσης

Τέλος, υπάρχει η δυνατότητα χρήσης εξωτερικής συνένωσης (δείτε και την αντίστοιχη Ενότητα 3.3.6 της σχεσιακής άλγεβρας) προσδιορίζοντας τα διακριτικά LEFT OUTER JOIN, RIGHT OUTER JOIN και FULL OUTER JOIN για την αριστερή, δεξιά και πλήρη εξωτερική συνένωση αντίστοιχα. Η χρήση της λέξης-κλειδί OUTER είναι και στις τρεις περιπτώσεις προαιρετική. Για παράδειγμα, η Ερώτηση 5.26 δίνει για κάθε ηθοποιό που έπαιξε σε μια ταινία το 2000, τον τίτλο της ταινίας

και τον ρόλο του σε αυτήν. Οι ηθοποιοί που δεν έπαιξαν σε κάποια ταινία του 2000 εμφανίζονται στο αποτέλεσμα έχοντας null τιμές στα άλλα δύο γνωρίσματα.

```
SELECT ACTOR.Name, PLAYS.Title, PLAYS.Role,  
FROM ACTOR LEFT JOIN PLAYS ON ACTOR.Name = PLAYS.Name  
WHERE Year = 2000;
```

Ερώτηση 5.26 Παράδειγμα εξωτερικής συνένωσης

Άσκηση Αυτοαξιολόγησης 5.9: Τιμές null

Έστω ένας πίνακας R(A, B) με αριθμητικά γνωρίσματα. Δώστε ένα παράδειγμα στιγμιότυπου αυτού του πίνακα όπου:

(α) η ερώτηση `SELECT * FROM R` δίνει διαφορετικό αποτέλεσμα από την ερώτηση `SELECT * FROM R WHERE A = A`,

(β) η ερώτηση `SELECT AVG(A) FROM R` δίνει διαφορετικό αποτέλεσμα από την ερώτηση `SELECT SUM(A)/COUNT(*) FROM R`.

Άσκηση Αυτοαξιολόγησης 5.10: Συνενώσεις.

Θεωρείστε τη σχεσιακή βάση δεδομένων για τις πιτσαρίες που ορίσατε στην Άσκηση Αυτοαξιολόγησης 5.1. Διατυπώστε ερωτήσεις SQL που να χρησιμοποιούν το κατάλληλο τύπο συνένωσης για να ανακτήσουν:

(α) τα μαγαζιά που σερβίρουν τουλάχιστον μια πίτσα που έχει κάποιο συστατικό που να αρέσει στον φοιτητή Δημήτρη.

(β) για κάθε φοιτητή με όνομα Φ, ένα ζεύγος Φ, Π, για κάθε πίτσα Π που περιέχει κάποιο συστατικό που αρέσει στο φοιτητή Φ. Για τους φοιτητές, για τους οποίους δεν υπάρχει καμία πίτσα που να έχει κάποιο συστατικό που τους αρέσει, θα πρέπει να υπάρχει ένα ζεύγος με τιμή null στο όνομα πίτσας.

Άσκηση Αυτοαξιολόγησης 5.11: Ερωτήσεις SQL

Θεωρείστε τη σχεσιακή βάση δεδομένων για τις πιτσαρίες που ορίσατε στην Άσκηση Αυτοαξιολόγησης 5.1. Διατυπώστε ερωτήσεις SQL που να επιστρέφουν ως αποτέλεσμα τα παρακάτω:

(α) Για κάθε πίτσα, τον αριθμό συστατικών της που αρέσουν στο φοιτητή Δημήτρη και την τιμή 0 αν δε του αρέσει κανένα από τα συστατικά της.

(β) τα συστατικά πίτσας που αρέσουν και στον Δημήτρη και στη Μαρία χρησιμοποιώντας (i) EXISTS, (ii) IN (iii) πράξη συνόλου και (iii) χωρίς χρήση κάποιου από τα προηγούμενα.

5.3 Εισαγωγή, Διαγραφή και Ενημέρωση

Στην προηγούμενη ενότητα, μελετήσαμε τις εντολές της SQL που αφορούν στη διατύπωση ερωτημάτων για την ανάκτηση πληροφορίας από μια σχεσιακή βάση δεδομένων. Στην ενότητα αυτή, θα εξετάσουμε τις εντολές της SQL που αφορούν στη διαχείριση του στιγμιότυπου της βάσης δεδομένων. Η SQL παρέχει τις εντολές: (α) INSERT για την εισαγωγή πλειάδων σε ένα πίνακα, (β) DELETE για τη διαγραφή πλειάδων από ένα πίνακα και (γ) UPDATE για την τροποποίηση των τιμών κάποιων γνωρισμάτων των πλειάδων ενός πίνακα. Στη συνέχεια, παρουσιάζουμε αυτές τις εντολές αναλυτικά.

Σημειώνουμε ότι κατά την υποβολή μιας εντολής εισαγωγής, διαγραφής ή ενημέρωσης, γίνεται έλεγχος αν η εντολή ικανοποιεί τους περιορισμούς ακεραιότητας που έχουν προσδιοριστεί κατά τον ορισμό του σχήματος. Η εντολή γίνεται δεκτή μόνο αν δεν παραβιάζει τους περιορισμούς.

5.3.1 Εισαγωγή Δεδομένων

Μπορούμε να χρησιμοποιήσουμε την εντολή INSERT για να εισάγουμε σε ένα πίνακα είτε μια μόνο πλειάδα είτε το αποτέλεσμα μιας SFW ερώτησης.

Για να εισάγουμε μια πλειάδα σε έναν πίνακα προσδιορίζουμε αρχικά το όνομα του πίνακα και στη συνέχεια την πρόταση VALUES ακολουθούμενη από μια λίστα τιμών, μια για κάθε γνώρισμα, ακολουθώντας τη σειρά που τα γνωρίσματα ορίζονται στον αντίστοιχο CREATE TABLE ορισμό, όπως φαίνεται και στην Εντολή 5.1(α). Αν δε θέλουμε να προσδιορίσουμε μια τιμή για όλα τα γνωρίσματα ή δε θέλουμε να ακολουθήσουμε τη σειρά προσδιορισμού των γνωρισμάτων στον ορισμό του πίνακα, μπορούμε αντί να προσδιορίσουμε μόνο το όνομα του πίνακα, να παραθέσουμε και τα γνωρίσματα των οποίων τις τιμές θέλουμε να εισάγουμε. Ένα σχετικό παράδειγμα είναι η Εντολή 5.1(β). Στα γνωρίσματα που παραλείπονται ανατίθεται αν υπάρχει η προκαθορισμένη τιμή τους ή αλλιώς η τιμή null.

```
INSERT INTO ACTOR VALUES ('Meryl Streep', '1949-06-22', 'Female', 'American');  
(α)  
INSERT INTO MOVIE(Title, Year) VALUES ('Blue Valentine', 2010);
```

(β)

Εντολή 5.1 Παραδείγματα εισαγωγής πλειάδων σε πίνακα με προσδιορισμό των τιμών (α) όλων και (β) μερικών γνωρισμάτων

Εναλλακτικά, μπορούμε να εισάγουμε σε ένα πίνακα το αποτέλεσμα μιας ερώτησης αντικαθιστώντας την VALUES με την ερώτηση. Η Εντολή 5.2 είναι ένα τέτοιο παράδειγμα, όπου για κάθε ταινία που γυρίστηκε το 2010, εισάγουμε μια ταινία με τον ίδιο τίτλο και την ίδια διάρκεια, αλλά με τιμή για το έτος 2015 (αν αυτή η ταινία δεν υπάρχει ήδη). Προσέξτε ότι πρώτα εκτελείται η ερώτηση και μετά εισάγεται το αποτέλεσμα της συνολικά στον πίνακα, δηλαδή, η υπό-ερώτηση NOT IN εκτελείται στον παλιό πίνακα.

```
INSERT INTO MOVIE
  (SELECT Title, 2015, Runtime
   FROM MOVIE
   WHERE Year = 2010 and (Title, 2015) NOT IN (SELECT Title, Year FROM MOVIE));
```

Εντολή 5.2 Παράδειγμα εισαγωγής σε πίνακα των αποτελεσμάτων μιας ερώτησης

5.3.2 Διαγραφή Δεδομένων

Με την εντολή DELETE μπορούμε να διαγράψουμε τις πλειάδες ενός πίνακα που ικανοποιούν μια συνθήκη του WHERE. Για παράδειγμα η Εντολή 5.3(α) διαγράφει από τον πίνακα MOVIE-GENRE τις πλειάδες με είδος δράμα και έτος 2010, ενώ η Εντολή 5.3(β) διαγράφει από τον πίνακα MOVIES τις ταινίες χωρίς ηθοποιούς. Παρατηρείστε ότι ενώ μπορούμε να διαγράψουμε πλειάδες από ένα μόνο πίνακα, στη συνθήκη του WHERE μπορούμε να έχουμε παραπάνω από έναν πίνακα. Το ίδιο ισχύει και για την εντολή UPDATE και για την ερώτηση (αν υπάρχει) της εντολής INSERT.

```
DELETE FROM MOVIE-GENRE
  WHERE Genre = 'drama' and Year = 2010;
  (α)

DELETE FROM MOVIE
  WHERE (Title, Year) NOT IN (SELECT Title, Year FROM PLAYS);
  (β)
```

Εντολή 5.3 Παραδείγματα διαγραφής πλειάδων

Αν δεν υπάρχει WHERE, διαγράφονται όλες οι πλειάδες του πίνακα. Για παράδειγμα, η εντολή DELETE FROM PLAYS, διαγράφει όλες τις πλειάδες του πίνακα PLAYS. Προσέξτε τη διαφορά από την εντολή DROP PLAYS. Η εντολή DROP διαγράφει όχι μόνο το περιεχόμενο του πίνακα PLAYS (όπως η εντολή DELETE) αλλά και τον ορισμό του πίνακα.

5.3.3 Τροποποίηση Δεδομένων

Με την εντολή UPDATE μπορούμε να τροποποιήσουμε την τιμή ενός γνωρίσματος ενός πίνακα για όλες τις πλειάδες του πίνακα που ικανοποιούν μια συνθήκη. Για παράδειγμα η Εντολή 5.4(α) αλλάζει στον πίνακα ACTOR την τιμή του γνωρίσματος Nationality σε Canadian για τον ηθοποιό Ryan Gosling, ενώ η Εντολή 5.4(β) αλλάζει στον πίνακα MOVIE-GENRE την τιμή του γνωρίσματος GENRE από `science-fiction` σε `Sci-Fi`. Αν λείπει το WHERE τότε ενημερώνεται η τιμή του γνωρίσματος σε όλες τις πλειάδες.

```
UPDATE ACTOR
SET Nationality ='Canadian'
WHERE Name = 'Ryan Gosling';
```

(α)

```
UPDATE MOVIE-GENRE
SET GENRE ='Sci-Fi'
WHERE GENRE ='science-fiction';
```

(β)

Εντολή 5.4 Παραδείγματα ενημέρωσης πλειάδων

Άσκηση Αυτοαξιολόγησης 5.12: Εντολές Ενημέρωσης και Περιορισμοί Ακεραιότητας

Θεωρείστε τον ορισμό του σχήματος για τη βάση δεδομένων για τις κινηματογραφικές ταινίες της Εικόνας 5.2. Για κάθε μια από τις εντολές INSERT, DELETE και UPDATE, εξετάστε ποιοι από τους περιορισμούς ακεραιότητας μπορεί να παραβιάζονται και δώστε αντίστοιχα παραδείγματα εντολών που μπορεί να οδηγήσουν σε παραβίαση τους.

Άσκηση Αυτοαξιολόγησης 5.13: Εντολές SQL

Θεωρείστε τη σχεσιακή βάση δεδομένων για τις πισταρίες που ορίσατε στην Άσκηση Αυτοαξιολόγησης 5.1. Δώστε εντολές που να έχουν ως αποτέλεσμα τα παρακάτω:

(α) εισαγωγή στον πίνακα PIZZA μιας νέας πίτσας με όνομα Dimitris-Pizza που να έχει ως συστατικά όλα τα συστατικά που αρέσουν στο φοιτητή Δημήτρη

(β) διαγραφή του συστατικού μανιτάρι (mushroom) από όλες τις πίτσες που περιέχουν μανιτάρι και ανανά (pineapple).

(γ) μείωση της τιμής κατά 1 για όλες τις πίτσες που περιέχουν μανιτάρι και έχουν τιμή μεγαλύτερη του 12.

Δραστηριότητα 5.1: Προγραμματισμός σε SQL

Εγκαταστήστε κάποιο σχεσιακό ΣΔΒΔ, για παράδειγμα τη MySQL.¹ Ορίστε το σχήμα για τη βάση δεδομένων της Εικόνας 3.5 χρησιμοποιώντας τις εντολές της Εικόνας 5.2. Στη συνέχεια, χρησιμοποιήστε την εντολή INSERT για να εισάγετε τουλάχιστον 5 πλειάδες σε κάθε έναν από τους πίνακες. Δοκιμάστε να εκτελέσετε τα σχετικά παραδείγματα ερωτήσεων και εντολών που παρουσιάζονται σε αυτό το κεφάλαιο. Προσοχή, αν και η SQL είναι η γλώσσα όλων των σχεσιακών ΣΔΒΔ, υπάρχουν μικρές διαφοροποιήσεις στη σύνταξη καθώς και επιπρόσθετοι περιορισμοί στα χαρακτηριστικά της που υποστηρίζει κάθε ΣΔΒΔ. Για παράδειγμα, η MySQL δεν υποστηρίζει ανάμεσα σε άλλα την έκφραση EXCEPT.

Εκτιμώμενος χρόνος ολοκλήρωσης: 60 λεπτά

Στόχος: Εξάσκηση στον προγραμματισμό σε SQL

Δραστηριότητα 5.2: Ο Διερμενέας RA της Σχεσιακής Άλγεβρας

Η σχεσιακή άλγεβρα είναι μια θεωρητική γλώσσα για τη διατύπωση ερωτήσεων σε μια σχεσιακή βάση δεδομένων. Για εκπαιδευτικούς λόγους, έχει αναπτυχθεί ο διερμενέας RA², ο οποίος εκτελεί ερωτήματα διατυπωμένα σε σχεσιακή άλγεβρα σε μια σχεσιακή βάση δεδομένων μεταφράζοντάς τα σε SQL. Σε αυτήν τη δραστηριότητα, θα χρησιμοποιήσετε τον RA, για να ελέγξετε τη γνώση σας στη σχεσιακή άλγεβρα. Η δραστηριότητα αυτή δίνεται σε αυτό το κεφάλαιο και όχι στο Κεφάλαιο 3, γιατί για να διατυπώσετε ερωτήματα στον RA πρέπει πρώτα να ορίσετε τη βάση δεδομένων σε ένα ΣΔΒΔ και να εισάγετε δεδομένα σε αυτήν χρησιμοποιώντας SQL.

Εγκαταστήστε το διερμηνέα RA και μελετήστε τις σύντομες οδηγίες για τη χρήση του. Ορίστε το σχήμα για τη βάση δεδομένων της Εικόνας 3.5 και εισάγετε δεδομένα όπως στη

¹ <http://www.mysql.com/>

² <http://www.cs.duke.edu/~junyang/ra/>

Δραστηριότητα 5.1. Στη συνέχεια, διατυπώστε τα αντίστοιχα παραδείγματα σχεσιακής άλγεβρας του Κεφαλαίου 3.

Εκτιμώμενος χρόνος ολοκλήρωσης: 60 λεπτά

Στόχος: Επανάληψη της σχεσιακής άλγεβρας

Σύνοψη

Στο κεφάλαιο αυτό, παρουσιάστηκε η γλώσσα SQL που είναι η βασική γλώσσα που υποστηρίζεται από τα σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων. Η SQL χρησιμοποιείται ως γλώσσα ορισμού του σχήματος μιας βάσης δεδομένων και ως γλώσσα χειρισμού των δεδομένων της. Ως γλώσσα ορισμού, η SQL παρέχει τις βασικές εντολές για τον ορισμό του σχήματος μιας σχεσιακής βάσης δεδομένων και των περιορισμών ακεραιότητας που ισχύουν σε αυτό.

Ως γλώσσα χειρισμού των δεδομένων, η SQL παρέχει εντολές για την διατύπωση ερωτήσεων και για τη διαχείριση του στιγμιότυπου της βάσης δεδομένων. Η γλώσσα ερωτήσεων επιτρέπει τη διατύπωση ερωτημάτων για την ανάκτηση πληροφορίας από τη βάση δεδομένων. Η βασική ερώτηση SQL αποτελείται από μια SELECT-FROM-WHERE πρόταση που είναι αντίστοιχη μιας select-project-join έκφρασης της σχεσιακής άλγεβρας. Τέλος, η SQL επιτρέπει την τροποποίηση του στιγμιότυπου της βάσης δεδομένων με την εισαγωγή, διαγραφή και ενημέρωση πλειάδων.

Λίστα Ελέγχου Γνώσεων

Έχοντας ολοκληρώσει τη μελέτη του κεφαλαίου που εισάγει τη γλώσσα βάσεων δεδομένων SQL, αξιολογείστε το βαθμό κατανόησης των εννοιών που αναπτύχθηκαν, επιστρέφοντας στα αρχικά προσδοκώμενα αποτελέσματα και ελέγχοντας κατά πόσο μπορείτε:

- να ορίσετε, χρησιμοποιώντας κατάλληλες εντολές CREATE, τους πίνακες μιας σχεσιακής βάσης δεδομένων, των γνωρισμάτων κάθε πίνακα και των τύπων αυτών των γνωρισμάτων καθώς και τους πιθανούς περιορισμούς ακεραιότητας,

- να εισάγετε, διαγράφετε και τροποποιείτε πλειάδες στους πίνακες μιας σχεσιακής βάσης δεδομένων χρησιμοποιώντας αντίστοιχα εντολές INSERT, DELETE και UPDATE,
- να διατυπώνετε κατάλληλες SFW ερωτήσεις ώστε να ανακτάτε από μια σχεσιακή βάση δεδομένων την πληροφορία που σας ενδιαφέρει.

Ευρετήριο όρων

SFW ερώτηση, 117, 123, 124, 129, 130, 144, 150, 151, 152
 SQL
 λογική τριών τιμών, 141
 πολυσύνολα, 137
 συναθροιστικές συναρτήσεις, 137, 138, 140, 141, 155
 υπό-ερωτήσεις, 132, 136

Γλωσσάρι

SQL (Structured Query Language): η πιο διαδεδομένη γλώσσα για τη διαχείριση σχεσιακών βάσεων δεδομένων.

SFW ερώτηση ή SPJ ερώτηση: η βασική δομή μιας ερώτησης SQL που αποτελείται από ένα SELECT-FROM-WHERE αίτημα

Βιβλιογραφία

Elmasri, R. & Navathe, S.B. (2003). *Fundamentals of Database Systems* (4th edition). Reading, MA: Addison Wesley.

Ramakrishnan, R. & Gehrke, J. (2002). *Database management systems* (3rd. edition). Boston: McGraw-Hill.

Οδηγός για περαιτέρω μελέτη

Στο κεφάλαιο αυτό περιγράφηκαν τα βασικά χαρακτηριστικά της SQL. Η SQL και παραλλαγές της υποστηρίζεται από πολλά εμπορικά συστήματα

συμπεριλαμβανομένων της ORACLE και της DB2 αλλά και από συστήματα ανοικτού κώδικα όπως η MySQL και η PostgreSQL. Για παραπάνω πληροφορίες μπορείτε να ανατρέξετε στα αντίστοιχα εγχειρίδια χρήσης.

Ο προγραμματισμός σε SQL καλύπτεται επίσης από όλα τα διδακτικά βιβλία των βάσεων δεδομένων. Ενδεικτικά, για να δείτε περισσότερα παραδείγματα προγραμματισμού σε SQL, μπορείτε να ανατρέξετε στο Κεφάλαιο 5 του βιβλίου των Ramakrishnan, R. & Gehrke, J. (2002)., και στα Κεφάλαια 8 και 9 του βιβλίου των Elmasri, R. & Navathe, S.B. (2003).

ΑΠΑΝΤΗΣΕΙΣ ΑΣΚΗΣΕΩΝ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ

Απάντηση Άσκησης Αυτοαξιολόγησης 5.1

(α) Στην Εικόνα 5.12 βλέπετε τον ορισμό των τριών πινάκων PIZZA, LIKES και SERVES για τη βάση δεδομένων με τις πιτσαρίες. Στον Πίνακα SERVES ορίσαμε ως ξένο κλειδί το Pizza-Name το οποίο αναφέρεται στον πίνακα SERVES καθώς και τις δράσεις ON DELETE CASCADE και ON DELETE UPDATE.

```
CREATE TABLE PIZZA
(Name VARCHAR(30),
Ingredient VARCHAR(30),
PRIMARY KEY (Name, Ingredient));
```

```
CREATE TABLE LIKES
(Student VARCHAR(30),
Ingredient VARCHAR(30);
PRIMARY KEY (Student, Ingredient));
```

```
CREATE TABLE SERVES
(Place VARCHAR(30),
Pizza-Name VARCHAR(30),
Price SMALLINT,
PRIMARY KEY (Place, Pizza-Name),
FOREIGN KEY (Pizza-Name)
REFERENCES PIZZA(Name)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

Εικόνα 5.12 Ορισμός σε SQL του σχήματος της βάσης δεδομένων για τις πιτσαρίες

(β) Στην Εικόνα 5.13 βλέπετε τον ορισμό των πινάκων USER, MESSAGE, USER-INTEREST, REPOSTS και FOLLOWS της βάσης δεδομένων για τα μηνύματα. Για

παράδειγμα, το User-id έχει οριστεί ως ξένο κλειδί και στον πίνακα MESSAGE και στον πίνακα REPOST με αντίστοιχες CASCADE δράσεις που σημαίνει ότι αν ο χρήστης διαγραφεί θα διαγραφούν αντίστοιχα τα μηνύματα (MESSAGE) που έχει στείλει ο ίδιος (και το περιεχόμενό τους) και η ένδειξη (REPOSTS) προώθησης για τα μηνύματα που έχει προωθήσει.

```
CREATE TABLE USER
(User-id INT PRIMARY KEY,
Address VARCHAR(30));
```

```
CREATE TABLE MESSAGE
(Message-id INT PRIMARY KEY,
Content VARCHAR(140) NOT NULL,
Post-time DATETIME NOT NULL,
User-sender-id VARCHAR(20),
FOREIGN KEY (User-Sender-id)
REFERENCES USER(User-id)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE USER-INTEREST
(User-id INT,
Interest VARCHAR(30),
PRIMARY KEY (User-id, Interest)
FOREIGN KEY (User-id)
REFERENCES USER(User-id)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE REPOSTS
(User-id INT,
Message-id INT,
PRIMARY KEY (User-id, Message-id)
FOREIGN KEY (User-id)
REFERENCES USER(User-id)
ON DELETE CASCADE
ON UPDATE CASCADE
FOREIGN KEY (Message-id)
REFERENCES MESSAGE(Message-id)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE FOLLOWS
(Follower INT,
Followee INT,
PRIMARY KEY (Follower, Followee)
FOREIGN KEY (Follower)
REFERENCES USER(User-id)
ON DELETE CASCADE
ON UPDATE CASCADE);
FOREIGN KEY (Followee)
REFERENCES USER(User-id)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

Εικόνα 5.13 Ορισμός σε SQL του σχήματος της βάσης δεδομένων για τα μηνύματα

Απάντηση Άσκησης Αυτοαξιολόγησης 5.2

(α) Η διαγραφή της πλειάδας με $A = 6$ επηρεάζει την πλειάδα (3, 6) του πίνακα S που αναφέρεται σε αυτήν. Στην περίπτωση (i) που έχουμε ορίσει ON DELETE CASCADE, η πλειάδα θα διαγραφεί (Εικόνα 5.14(α)), στην περίπτωση (ii) που

έχουμε ορίσει ON DELETE SET NULL, η τιμή 6 θα αντικατασταθεί με NULL (Εικόνα 5.14(β)), στην περίπτωση (iii) που έχουμε ορίσει ON DELETE SET DEFAULT. η τιμή 6 θα αντικατασταθεί με την DEFAULT τιμή του D δηλαδή με 1 (Εικόνα 5.14(γ)). Τέλος, στην περίπτωση (iv) στην οποία δεν έχει προσδιοριστεί κάποια δράση, η διαγραφή της πλειάδας της σχέσης R, δε γίνεται αποδεκτή γιατί παραβιάζεται ένας περιορισμός ακεραιότητας και άρα ο πίνακας S δε μεταβάλλεται (Εικόνα 5.14(δ)).

S	
C	D
2	1
5	1
4	9
7	9

S	
C	D
2	1
3	NULL
5	1
4	9
7	9

S	
C	D
2	1
3	1
5	1
4	9
7	9

S	
C	D
2	1
3	6
5	1
4	9
7	9

S	
C	D
2	1
3	10
5	1
4	9
7	9

(α) (β) (γ) (δ) (ε)

Εικόνα 5.14 (α) ON DELETE CASCADE, (β) ON DELETE SET NULL, (γ) ON DELETE SET DEFAULT, (δ) καμία δράση, (ε) ON UPDATE CASCADE

(β) Αντίστοιχα με το ερώτημα (α) η αλλαγή της τιμής του A από 6 σε 10 επηρεάζει την πλειάδα (3, 6) του πίνακα S που αναφέρεται σε αυτήν. Στις περιπτώσεις (ii), (iii) και (iv), το αποτέλεσμα είναι το ίδιο με αυτό του ερωτήματος (α). Αντίθετα στην περίπτωση (i) που έχουμε ορίσει ON UPDATE CASCADE, η αλλαγή της τιμής του A μεταδίδεται στην αναφερόμενη πλειάδα όπως φαίνεται στην Εικόνα 5.14(ε).

Απάντηση Άσκησης Αυτοαξιολόγησης 5.3

(α) Η πληροφορία για τα ονόματα των πιτσών που περιέχουν μανιτάρι υπάρχει στον πίνακα PIZZA και η Ερώτηση 5.27(α) μας δίνει το ζητούμενο αποτέλεσμα.

```
SELECT Name
FROM PIZZA
WHERE Ingredient= `mushroom`;
(α)
```

```
SELECT Ingredient
FROM PIZZA
WHERE Pizza = `Special`;
(β)
```

Ερώτηση 5.27 Απλές ερωτήσεις SFW

(β) Αντίστοιχα, η Ερώτηση 5.27(β) ανακτά τα συστατικά της πίτσας Special.

(γ) Για τα συστατικά που αρέσουν σε έναν τουλάχιστον φοιτητή θα χρησιμοποιήσουμε τον πίνακα LIKES όπως στην Ερώτηση 5.28.

```
SELECT DISTINCT Ingredient
FROM LIKES;
```

Ερώτηση 5.28 Απλή ερώτηση SFW με χρήση DISTINCT

Παρατηρείστε ότι στην Ερώτηση 5.28 χρησιμοποιήσαμε το διακριτικό DISTINCT ώστε ένα συστατικό να εμφανίζεται μόνο μια φορά στο αποτέλεσμα. Σε αντίθετη περίπτωση, κάθε συστατικό θα εμφανιζόταν στο αποτέλεσμα, τόσες φορές όσοι οι διαφορετικοί φοιτητές στους οποίους αρέσει. Στα προηγούμενα ερωτήματα, στο αποτέλεσμα δε μπορούσαν να υπάρχουν διπλότυπα, οπότε το DISTINCT ήταν περιττό. Στην Εικόνα 5.15(α) βλέπετε το αποτέλεσμα της Ερώτησης 5.28 χωρίς DISTINCT και στην Εικόνα 5.15(β) το αποτέλεσμα με DISTINCT για το στιγμιότυπο της Εικόνας 3.12.

Ingredient
mushroom
ham
olive
mushroom
ham
bacon
pineapple

(α)

Ingredient
mushroom
ham
olive
bacon
pineapple

(β)

Εικόνα 5.15 Αποτέλεσμα της Ερώτησης 5.28 (α) χωρίς τη χρήση DISTINCT, (β) με χρήση DISTINCT

(δ) Για να βρούμε τα μαγαζιά που σερβίρουν πίτσα με συστατικό που αρέσει τουλάχιστον σε ένα φοιτητή πρέπει να συνενώσουμε και τους τρεις πίνακες της βάσης δεδομένων, όπως φαίνεται στην Ερώτηση 5.29 στην οποία χρησιμοποιήσαμε μεταβλητές πλειάδας για ευκολία στη διατύπωση.

```
SELECT DISTINCT C.Place
FROM LIKES L, PIZZA P, PLACE C
WHERE L.Ingredient = P.Ingredient and P.Name = C.Pizza-Name;
```

Ερώτηση 5.29 Ερώτηση SFW

(ε) Για να βρούμε τα μαγαζιά που σερβίρουν τουλάχιστον μια πίτσα με κάποιο συστατικό που αρέσει στον Δημήτρη αρκεί να προσθέσουμε στην Ερώτηση 5.29 ακόμα μια συνθήκη για το όνομα του φοιτητή, όπως φαίνεται στην Ερώτηση 5.30

```
SELECT DISTINCT C.Place
FROM LIKES L, PIZZA P, PLACE C
WHERE L.Ingredient = P.Ingredient and P.Name = C.Pizza-Name
AND L.Student = 'Dimitris';
```

Ερώτηση 5.30 Ερώτηση SFW

(στ) Για να βρούμε τις πίτσες που έχουν τουλάχιστον δυο διαφορετικά συστατικά θα χρειαστεί μια συνένωση του πίνακα PIZZA με τον εαυτό του. Η πράξη αυτή καλείται και αυτοσυνένωση (self-join). Συγκεκριμένα, θα συνενώσουμε τον πίνακα PIZZA με τον εαυτό του, όπως φαίνεται στην Ερώτηση 5.31 στην οποία χρησιμοποιούμε μεταβλητές πλειάδας για διάκριση των διπλών εμφανίσεων των γνωρισμάτων του πίνακα.

```
SELECT DISTINCT P1.Name
FROM, PIZZA P1, PIZZA P2
WHERE P1.Name = P2.Name and P1.Ingredient <> P2.Ingredient;
```

Ερώτηση 5.31 Ερώτηση SFW με αυτοσυνένωση

Απάντηση Άσκησης Αυτοαξιολόγησης 5.4

(α) Για τον υπολογισμό των πλειάδων που δεν περιέχουν κάποιο συστατικό, θα χρησιμοποιήσουμε την πράξη της διαφοράς όπως και στη σχεσιακή άλγεβρα. Η Ερώτηση 5.32 δίνει την απάντηση.

```
(SELECT Name FROM, PIZZA)
EXCEPT
(SELECT Name FROM PIZZA WHERE Ingredient = `mushroom`);
```

Ερώτηση 5.32 Ερώτηση με αφαίρεση

(β) Όπως και στη σχεσιακή άλγεβρα, η ερώτηση μπορεί να διατυπωθεί με δύο τρόπους: (i) με χρήση OR στη συνθήκη της επιλογής (Ερώτηση 5.33(α)) ή με ένωση

(Ερώτηση 5.33(β)). Προσέξτε ότι το διακριτικό DISTINCT είναι απαραίτητο στην Ερώτηση 5.33(α). Στην αντίθετη περίπτωση, μια πίτσα που περιέχει και τα δύο συστατικά θα εμφανίζεται δύο φορές στο αποτέλεσμα.

<pre>SELECT DISTINCT Name FROM PIZZA WHERE Ingredient = 'mushroom' OR Ingredient = 'ham';</pre>	<pre>(SELECT Name FROM, PIZZA WHERE Ingredient = 'mushroom') UNION (SELECT Name FROM PIZZA WHERE Ingredient = 'ham');</pre>
(α)	(β)

Ερώτηση 5.33 Ερώτηση με (α) OR και (β) ένωση

(γ) Θα χρησιμοποιήσουμε την πράξη της τομής. Η ερώτηση που δίνει την απάντηση στο ερώτημα είναι ίδια με την Ερώτηση 5.33(β) με μόνη διαφορά την αλλαγή του UNION με INTERSECT.

(δ) Θα χρησιμοποιήσουμε την πράξη της διαφοράς. Η ερώτηση που δίνει την απάντηση στο ερώτημα είναι ίδια με την Ερώτηση 5.33(β) με μόνη διαφορά την αλλαγή του UNION με EXCEPT.

Απάντηση Άσκησης Αυτοαξιολόγησης 5.5

Η ερώτηση μας δίνει ως αποτέλεσμα τους χρήστες που ο αριθμός των χρηστών που ακολουθούν είναι μεγαλύτερος από τον αριθμό των χρηστών που τους ακολουθεί. Για να το δείτε, θεωρίστε ότι ένας χρήστης ακολουθεί m_1 χρήστες και τον ακολουθούν m_2 χρήστες. Αυτός ο χρήστης θα εμφανίζεται m_1 φορές στο αποτέλεσμα του πρώτου select και m_2 φορές στο αποτέλεσμα του δεύτερου select. Ένας χρήστης εμφανίζεται στο αποτέλεσμα, αν και μόνο αν, $m_1 > m_2$.

Ας δούμε το αποτέλεσμα του ίδιου ερωτήματος αν αντί για EXCEPT ALL έχουμε EXCEPT. Τότε ένας χρήστης θα εμφανίζεται στο αποτέλεσμα μόνο αν εμφανίζεται στο αποτέλεσμα του πρώτου select και όχι στο αποτέλεσμα του δεύτερου select. Δηλαδή, το αποτέλεσμα είναι οι χρήστες που ακολουθούν τουλάχιστον ένα χρήστη και δεν τους ακολουθεί κανείς.

Απάντηση Άσκησης Αυτοαξιολόγησης 5.6

Η άσκηση αυτή ζητά να βρεθεί η πλειάδα με τη μεγαλύτερη τιμή στο γνώρισμα YEAR. Στην Ενότητα 5.2.4, θα δούμε τον υπολογισμό του ίδιου ερωτήματος με χρήση συναθροιστικής συνάρτησης.

(α) Στην Ερώτηση 5.34, η υπό-ερώτηση NOT EXISTS αποκλείει από το αποτέλεσμα τις κωμωδίες για τις οποίες υπάρχει μια κωμωδία πιο πρόσφατη από αυτές.

```
SELECT G1.Title, G1.Year
FROM MOVIE-GENRE AS G1
WHERE G1.Genre = 'comedy' AND
      NOT EXISTS (SELECT *
                  FROM MOVIE- GENRE AS G2
                  WHERE G2.Genre = 'comedy'
                  AND G2.Year > G1.Year);
```

Ερώτηση 5.34 Ερώτηση υπολογισμού μέγιστου με not exists

(β) Το ερώτημα αυτό είναι πιο δύσκολο. Θα χρειαστεί μια συνένωση του πίνακα MOVIE-GENRE με τον εαυτό του όπως φαίνεται στην Ερώτηση 5.35. Για να δείτε γιατί η Ερώτηση 5.35 μας δίνει το σωστό αποτέλεσμα, σκεφτείτε ότι το μόνο έτος που δεν εμφανίζεται στο αποτέλεσμα της NOT IN υπό-ερώτησης είναι το μεγαλύτερο. Η Ερώτηση 5.35 μας δίνει ένα γενικό τρόπο για να υπολογίζουμε τη μέγιστη (ή ελάχιστη) τιμή ενός συνόλου.

```
SELECT Title, Year
FROM MOVIE-GENRE
WHERE Genre = 'comedy' AND
      NOT IN (SELECT G1.Year
             FROM MOVIE- GENRE AS G1, MOVIE-GENRE AS G2
             WHERE G1.Genre = 'comedy' AND G2.Genre = 'comedy'
             AND G1.Year < G2.Gear);
```

Ερώτηση 5.35 Ερώτηση υπολογισμού μέγιστου με συνενώσεις

(γ) Η Ερώτηση 5.36 επιστρέφει τις κωμωδίες για τις οποίες το έτος είναι μεγαλύτερο ή ίσο με το έτος όλων των κωμωδιών.

```
SELECT G1.Title, G1.Year
FROM MOVIE-GENRE AS G1
WHERE G1.Genre = 'comedy' AND
      G1.Year >= ALL (SELECT G2.Year
                    FROM MOVIE-GENRE as G2
                    WHERE G2.Genre = 'comedy');
```


Ερώτηση 5.36 Ερώτηση υπολογισμού μέγιστου με all

(δ) Η Ερώτηση 5.37 επιστρέφει τις κωμωδίες για τις οποίες δεν υπάρχει άλλη κωμωδία με μικρότερο έτος.

```
SELECT G1.Title, G1.Year
FROM MOVIE-GENRE AS G1
WHERE G1.Genre = `comedy` AND
NOT (G1.Year < ANY (SELECT G2.Year
                    FROM MOVIE-GENRE as G2
                    WHERE G2.Genre = `comedy`));
```

Ερώτηση 5.37 Ερώτηση υπολογισμού μέγιστου με any

Τέλος, να σημειώσουμε ότι το ερώτημα (γ) μας δίνει ένα τρόπο να υπολογίσουμε τη μέγιστη τιμή και στην σχεσιακή άλγεβρα, αρκεί αντί για NOT IN να χρησιμοποιήσουμε αφαίρεση.

Απάντηση Άσκησης Αυτοαξιολόγησης 5.7

(α) Το αποτέλεσμα είναι το άθροισμα των τιμών του γνωρίσματος B, δηλαδή 34, όπως φαίνεται στην Εικόνα 5.16(α).

(β) Επειδή υπάρχει το DISTINCT γίνεται απαλοιφή διπλότυπων και υπολογίζεται το άθροισμα των διακριτών τιμών του B, που είναι ίσο με 24, όπως φαίνεται στην Εικόνα 5.16(β).

(γ) Αρχικά υπολογίζεται το καρτεσιανό γινόμενο και στη συνέχεια εφαρμόζεται η συνθήκη του WHERE. Οι πλειάδες που ικανοποιούν τη συνθήκη φαίνονται στην Εικόνα 5.16(γ) ομαδοποιημένες με βάση την τιμή του γνωρίσματος ομαδοποίησης D. Από αυτές, η ομάδα που αντιστοιχεί στο D = 6, δεν ικανοποιεί τη συνθήκη του HAVING και άρα οι συναθροιστικές συναρτήσεις δεν υπολογίζονται σε αυτήν. Το αποτέλεσμα φαίνεται στην Εικόνα 5.16(δ).

Sum_B
34

(α)

Sum_B
24

(β)

A	B	C	D
1	2	2	1
3	2	2	1
2	4	4	9
5	7	7	9
7	3	3	6

(γ)

D	SUM_A
1	4
9	7

(δ)

Εικόνα 5.16 Αποτελέσματα συναθροιστικών συναρτήσεων

Απάντηση Άσκησης Αυτοαξιολόγησης 5.8

(α) Η Ερώτηση 5.38 μας δίνει το μαγαζί που σερβίρει τη φθηνότερη πίτσα Special. Θυμηθείτε ότι όταν έχουμε συναθροιστικές συναρτήσεις χωρίς GROUP BY στην πρόταση SELECT δε μπορούμε να έχουμε άλλα γνωρίσματα. Για αυτό το λόγο, χρειαζόμαστε την εξωτερική ερώτηση για να βρούμε το όνομα του μαγαζιού.

```
SELECT S.Place
FROM SERVES AS S
WHERE S.Pizza-Name = 'Special' and S.Price = (SELECT MIN(Price)
                                           FROM SERVES
                                           WHERE Pizza-Name = 'Special');
```

Ερώτηση 5.38 Ανάκτηση πλειάδας με ελάχιστη τιμή

(β) Η ερώτηση θα πρέπει πρώτα να επιλέγει τις πίτσες που έχουν μανιτάρι και στη συνέχεια να τις ομαδοποιεί με βάση το όνομα τους. Επειδή, θέλουμε τη μέση τιμή μόνο για τις πίτσες που σερβίρονται σε τουλάχιστον δύο μαγαζιά, θα πρέπει να διατυπώσουμε μια συνθήκη στις ομάδες χρησιμοποιώντας το HAVING που να διατυπώνει αυτόν τον περιορισμό. Η Ερώτηση 5.39 είναι το ζητούμενο.

```
SELECT P.Pizza-Name, AVG(P.Price)
FROM SERVES AS S, PIZZA as P
WHERE P.Pizza-Name = S.Pizza-Name AND P.Ingredient = 'mushroom'
GROUP BY P.Pizza-Name
HAVING COUNT >=2;
```

Ερώτηση 5.39 Ερώτηση με HAVING

(γ) Το ερώτημα αυτό είναι σχετικά πιο περίπλοκο. Η Ερώτηση 5.40 χρησιμοποιεί μια συσχετισμένη υπό-ερώτηση, η οποία υπολογίζει για κάθε φοιτητή της εξωτερικής ερώτησης την φθηνότερη πίτσα που έχει ένα συστατικό που του αρέσει.

```
SELECT DISTINCT L1.Student, S1.Place
FROM SERVES AS S1, PIZZA AS P1, LIKES AS L1
WHERE S1.Pizza-Name = P1.Name AND L1.Ingredient = P1.Ingredient AND
      S1.Price = (SELECT MIN(PRICE)
                 FROM SERVES AS S2, PIZZA AS P2, LIKES AS L2
                 WHERE S2.Pizza-Name = P2.Name AND
                       L2.Ingredient = P2.Ingredient AND
                       L2.Student = L1.Student);
```

Ερώτηση 5.40 Συναθροιστική συνάρτηση με αυτοσυσχετιζόμενη υπό-ερώτηση

Απάντηση Άσκησης Αυτοαξιολόγησης 5.9

Ο λόγος που μπορεί να προκύψουν διαφορετικά αποτελέσματα είναι η ύπαρξη τιμών null στο γνώρισμα A. Έστω το στιγμιότυπο της Εικόνας 5.17(α).

A	B	A	A	AVG_A	SUM/COUNT_A
1	1	1	1	3	2
5	2	5	5		
NULL	3	NULL			

(α) (β) (γ) (δ) (ε)

Εικόνα 5. 17 Στιγμιότυπο για την Άσκηση Αυτοαξιολόγησης 5.9: οι τιμές null μπορεί να οδηγήσουν σε μη αναμενόμενα αποτελέσματα

(α) Στην Εικόνα 5.17(β) βλέπετε το αποτέλεσμα της πρώτης ερώτησης και στην Εικόνα 5.17(γ) της δεύτερης ερώτησης. Η δεύτερη ερώτηση έχει αυτό το αποτέλεσμα, γιατί η σύγκριση με την τιμή null ακόμα και με μια άλλη τιμή null έχει ως αποτέλεσμα την τιμή UNKNOWN και άρα η τελευταία πλειάδα της R (με τιμή null στο A) δεν εμφανίζεται στο αποτέλεσμα.

(β) Στην Εικόνα 5.17(δ) βλέπετε το αποτέλεσμα της πρώτης ερώτησης και στην Εικόνα 5.17(ε) της δεύτερης ερώτησης. Οι συναρτήσεις AVG και SUM αγνοούν την τιμή null και υπολογίζουν αντίστοιχα το μέσο όρο και το άθροισμα των τιμών 1 και 5, που είναι 3 και 6. Η COUNT(*) όμως δεν αγνοεί την τιμή null και υπολογίζει το πλήθος των πλειάδων της R ίσο με 3. Άρα η πρώτη ερώτηση έχει ως αποτέλεσμα 3 και η δεύτερη $6/3 = 2$.

Απάντηση Άσκησης Αυτοαξιολόγησης 5.10

(α) Για τον υπολογισμό της απάντησης χρειαζόμαστε δεδομένα και από τους τρεις πίνακες της βάσης δεδομένων και άρα απαιτούνται δύο συνενώσεις. Για τους πίνακες LIKES και PIZZA των οποίων το γνώρισμα της συνένωσης (Ingredient) είναι το ίδιο θα χρησιμοποιήσουμε φυσική συνένωση. Η απάντηση δίνεται στην Ερώτηση 5.41.

```
SELECT DISTINCT Place
FROM (LIKES NATURAL JOIN PIZZA) JOIN SERVES ON Name = Pizza-Name
WHERE Student = 'Dimitris'
```

Ερώτηση 5.41 Ερώτηση με φυσική συνένωση

(β) Επειδή θέλουμε κάθε φοιτητής να εμφανίζεται στο αποτέλεσμα ακόμα και αν δε ικανοποιεί τη συνθήκη της συνένωσης θα χρησιμοποιήσουμε εξωτερική συνένωση και συγκεκριμένα αριστερή εξωτερική συνένωση με αριστερή σχέση τη σχέση LIKES, όπως φαίνεται στην Ερώτηση 5.42.

```
SELECT DISTINCT Student, Name
FROM LIKES LEFT OUTER JOIN PIZZA ON LIKES.Ingredient = PIZZA.Ingredient;
```

Ερώτηση 5.42 Ερώτηση με εξωτερική συνένωση

Απάντηση Άσκησης Αυτοαξιολόγησης 5.11

(α) Για τον υπολογισμό της απάντησης, θα ομαδοποιήσουμε τα συστατικά ανά πίτσα και θα χρησιμοποιήσουμε τη συνάρτηση COUNT για να μετρήσουμε το πλήθος τους. Επειδή με αυτόν τον τρόπο δε δημιουργούνται ομάδες για τις πίτσες που δεν έχουν κάποιο συστατικό, θα προσθέσουμε στο αποτέλεσμα αυτές τις πίτσες χρησιμοποιώντας UNION. Η Ερώτηση 5.43 υλοποιεί τα παραπάνω.

```
(SELECT Name, Count(Ingredient)
FROM LIKES NATURAL JOIN PIZZA
WHERE Student = 'Dimitris'
GROUP BY Name)
UNION
(SELECT P.Name, 0
FROM PIZZA as P
WHERE NOT EXISTS (SELECT *
FROM LIKES NATURAL JOIN PIZZA
WHERE Student = 'Dimitris' AND P.Name = PIZZA.Name):
```

Ερώτηση 5.43 Περίπλοκη ερώτηση SQL

(β)

(i) Η Ερώτηση 5.44 υπολογίζει το αποτέλεσμα χρησιμοποιώντας EXISTS για να εξετάσει για κάθε συστατικό που αρέσει στον Δημήτρη αν υπάρχει αντίστοιχη πλειάδα στη LIKES για αυτό το συστατικό και τη Μαρία.

```

SELECT Ingredient
FROM LIKES as L
WHERE Student = 'Dimitris' AND
      EXISTS (SELECT *
              FROM LIKES
              WHERE Student = 'Maria' AND Ingredient = L.Ingredient):

```

Ερώτηση 5.44 Ερώτηση με exists

(ii) Η Ερώτηση 5.45 υπολογίζει το αποτέλεσμα χρησιμοποιώντας IN για να εξετάσει αν το συστατικό που αρέσει στον Δημήτρη ανήκει στα συστατικά που αρέσουν στη Μαρία.

```

SELECT Ingredient
FROM LIKES as L
WHERE Student = 'Dimitris' AND Ingredient IN (SELECT Ingredient
                                             FROM LIKES
                                             WHERE Student = 'Maria');

```

Ερώτηση 5.45 Ερώτηση με in

(iii) Η Ερώτηση 5.46 υπολογίζει το αποτέλεσμα ως την τομή (INTERSECT) των συστατικών που αρέσουν στο Δημήτρη και των συστατικών που αρέσουν στη Μαρία.

```

(SELECT Ingredient FROM LIKES as L WHERE Student = 'Dimitris')
INTERSECT
(SELECT Ingredient FROM LIKES as L WHERE Student = 'Maria');

```

Ερώτηση 5.46 Ερώτηση με τομή

(iv) Η Ερώτηση 5.47 υπολογίζει το αποτέλεσμα συνενώνοντας τη LIKES με τον εαυτό της (self-join).

```

SELECT L1.Ingredient
FROM LIKES as L1, LIKES AS L2
WHERE L1.Ingredient = L2.Ingredient and L1.Student = 'Dimitris' AND
      L2.Student = 'Maria');

```

Ερώτηση 5.47 Ερώτηση με αυτοσυσχέτιση

Απάντηση Άσκησης Αυτοαξιολόγησης 5.12

Ας εξετάσουμε τους διάφορους περιορισμούς ακεραιότητας. Αρχικά ο περιορισμός του πεδίου ορισμού, παραβιάζεται όταν σε μια εντολή INSERT η τιμή για κάποιο γνώρισμα δεν ανήκει στο πεδίο ορισμού αυτού του γνωρίσματος. Για παράδειγμα, η εντολή INSERT INTO MOVIES (15, 2013, 60) παραβιάζει τον περιορισμό πεδίου ορισμού για το γνώρισμα Title αφού η τιμή 15 δεν είναι τύπου VARCHAR.

Ο περιορισμός του κλειδιού παραβιάζεται όταν μια εντολή INSERT ή UPDATE αναθέτει στα γνωρίσματα του κλειδιού τιμές που ήδη υπάρχουν. Για παράδειγμα, η εντολή INSERT INTO MOVIES ('Amour', 2012, 100) στο στιγμιότυπο της Εικόνας 3.2 παραβιάζει τον περιορισμό κλειδιού του πίνακα MOVIES.

Ο περιορισμός ξένου κλειδιού παραβιάζεται από μια εντολή INSERT όταν γίνεται εισαγωγή στον αναφέροντα πίνακα μια πλειάδας με τιμή στα γνωρίσματα του ξένου κλειδιού που δεν υπάρχουν στον αναφερόμενο πίνακα. Για παράδειγμα, η εντολή INSERT INTO PLAYS ('Julia Roberts', 'Ocean's Eleven', 2011, 'Tess Ocean') στο στιγμιότυπο της Εικόνας 3.2 παραβιάζει τον περιορισμό ξένου κλειδιού για τον πίνακα PLAYS γιατί δεν υπάρχει στον πίνακα ACTOR η ηθοποιός Julia Roberts.

Ο περιορισμός ξένου κλειδιού παραβιάζεται από μια εντολή DELETE ή UPDATE όταν η διαγραφή ή η ενημέρωση οδηγεί στη διαγραφή μιας πλειάδας στην οποία αναφέρονται άλλες πλειάδες. Για παράδειγμα, η εντολή DELETE FROM ACTORS WHERE Name = 'George Clooney' στο στιγμιότυπο της Εικόνας 3.2, παραβιάζει τον περιορισμό ξένου κλειδιού για τον πίνακα PLAYS γιατί οι δύο πρώτες πλειάδες του αναφέρονται τώρα σε ηθοποιό που δεν υπάρχει. Επειδή δεν έχει οριστεί σχετική ON DELETE δράση στην Εικόνα 5.2, η προκαθορισμένη δράση είναι η εντολή DELETE να μη γίνει δεκτή.

Τέλος, άλλοι περιορισμοί είναι ο περιορισμός NOT NULL και οι σημασιολογικοί περιορισμοί που ορίζονται μέσω του CHECK. Οι περιορισμοί αυτοί παραβιάζονται από εντολές INSERT και UPDATE που αντίστοιχα εισάγουν τιμές NULL ή τιμές που δεν ικανοποιούν τη συνθήκη του CHECK.

Απάντηση Άσκησης Αυτοαξιολόγησης 5.13

(α) Θα χρησιμοποιήσουμε την εντολή INSERT όπως φαίνεται στην Εντολή 5.5.

```
INSERT INTO PIZZA  
(SELECT 'Dimitris-Pizza', Ingredient FROM LIKES WHERE Student = 'Dimitris');
```

Εντολή 5.5 Εισαγωγή πλειάδων

(β) Η Εντολή 5.6 χρησιμοποιεί την εντολή DELETE για να διαγράψει το συστατικό μανιτάρι από μια πίτσα. Η εσωτερική ερώτηση ελέγχει ότι η πίτσα περιέχει και ανανά.

```
DELETE FROM PIZZA as P
WHERE P.Ingredient = 'mushroom' AND
      P.Name IN (SELECT Name
                FROM Pizza
                WHERE Ingredient = 'pineapple');
```

Εντολή 5.6 Διαγραφή πλειάδων

(γ) Η Εντολή 5.7 χρησιμοποιεί την εντολή UPDATE για να μειώσει κατά 1 την τιμή για όλες τις πίτσες που ικανοποιούν τις συνθήκες της εκφώνησης.

```
UPDATE FROM SERVES
SET Price = Price - 1
WHERE Price > 12 AND Pizza-Name IN (SELECT Name
                                     FROM Pizza
                                     WHERE Ingredient = 'mushroom');
```

Εντολή 5.7 Ενημέρωση πλειάδων

ΑΠΑΝΤΗΣΕΙΣ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Απάντηση Δραστηριότητας 5.1

Σκοπός της δραστηριότητας είναι η εξάσκηση σε προγραμματισμό σε SQL. Μπορεί να χρειαστεί κάποιος επιπρόσθετος χρόνος για την εγκατάσταση του λογισμικού. Επίσης, καλό θα ήταν να πειραματιστείτε και με παραλλαγές των ερωτήσεων που παρουσιάζονται σε αυτό το κεφάλαιο και στις ασκήσεις αυτοαξιολόγησης.

Απάντηση Δραστηριότητας 5.2

Όπως και στην προηγούμενη δραστηριότητα, μπορεί να χρειαστεί κάποιος επιπρόσθετος χρόνος για την εγκατάσταση του λογισμικού. Επίσης, προσέξτε να έχετε ήδη εγκαταστήσει κάποιο ΣΔΒΔ (για παράδειγμα, τη MySQL).

Κατάλογος Εικόνων

Εικόνα 5. 1 Εντολές ορισμού ενός νέου σχήματος σχέσης με χρήση του συμβολισμού BNF

Εικόνα 5. 2 Ορισμός σε SQL του σχήματος της βάσης δεδομένων για κινηματογραφικές ταινίες

Εικόνα 5. 3 Εντολές αλλαγής σχήματος

Εικόνα 5. 4 Στιγμιότυπο για την Άσκηση Αξιολόγησης 5.2

Εικόνα 5. 5 Η βασική SFW δομής μιας ερώτησης SQL

Εικόνα 5. 6 Αποτέλεσμα της Ερώτησης 5.3 για το είδος ταινίας (α) χωρίς της χρήση DISTINCT, (β) με χρήση DISTINCT, ισοδύναμο με το $\pi_{\text{Genre}}(\text{MOVIE-GENRE})$

Εικόνα 5. 7 Πράξεις συνόλου στην SQL

Εικόνα 5. 8 Αποτελέσματα ερωτήσεων συνόλου στο στιγμιότυπο της Εικόνας 3.5: (α) αποτέλεσμα του πρώτου SFW των Ερωτήσεων 5.11, (β) αποτέλεσμα του δεύτερου SFW των Ερωτήσεων 5.11, (γ) αποτέλεσμα του UNION (Ερώτηση 5.11(α)), (δ) αποτέλεσμα του INTERSECT (Ερώτηση 5.11(β)), (ε) αποτέλεσμα του EXCEPT (Ερώτηση 5.11(γ)), (στ) αποτέλεσμα αν θεωρήσουμε UNION ALL αντί UNION στην Ερώτηση 5.11(α), (ζ) αποτέλεσμα αν θεωρήσουμε INTERSECT ALL αντί INTERSECT στην Ερώτηση 5.11(β), (η) αποτέλεσμα αν θεωρήσουμε EXCEPT ALL αντί EXCEPT στην Ερώτηση 5.11(γ)

Εικόνα 5. 9 (α) Ο πίνακας BAG1 (β) ο πίνακας BAG2 και (γ)-(ε) αποτελέσματα των Ερωτήσεων 5.12(α)-(γ) αντίστοιχα, δηλαδή, (γ) UNION ALL, (δ) INTERSECT ALL, (ε) EXCEPT ALL

Εικόνα 5. 10 Γενική μορφή μιας ερώτησης εμφωλευμένης στην πρόταση WHERE μιας άλλης ερώτησης

Εικόνα 5. 11 Αποτέλεσμα (α) της Ερώτησης 5.20(α) και (β) της Ερώτησης 5.20(β) στο στιγμιότυπο της Εικόνας 3.2

Εικόνα 5. 12 Ορισμός σε SQL του σχήματος της βάσης δεδομένων για τις πιτσαρίες

Εικόνα 5. 13 Ορισμός σε SQL του σχήματος της βάσης δεδομένων για τα μηνύματα

Εικόνα 5. 14 (α) ON DELETE CASCADE, (β) ON DELETE SET NULL, (γ) ON DELETE SET DEFAULT, (δ) καμία δράση, (ε) ON UPDATE CASCADE

Εικόνα 5. 15 Αποτέλεσμα της Ερώτησης 5.28 (α) χωρίς τη χρήση DISTINCT, (β) με χρήση DISTINCT

Εικόνα 5. 16 Αποτελέσματα συναθροιστικών συναρτήσεων

Εικόνα 5. 17 Στιγμιότυπο για την Άσκηση Αυτοαξιολόγησης 5.9: οι τιμές null μπορεί να οδηγήσουν σε μη αναμενόμενα αποτελέσματα

Κατάλογος Πινάκων

Πίνακας 5. 1 Λογική τριών τιμών: FALSE (F), TRUE (T), UNKNOWN (U)