ΑΤΕΙ ΗΠΕΙΡΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑ: «Υλοποίηση βάσης δεδομένων για τη Διαχείριση Αλυσίδας Σούπερ Μάρκετ»

ΦΟΙΤΗΤΗΣ
Λαθύρης Ιωάννης Α.Μ. 8320

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία πραγματοποιήθηκε για το Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπειρού, Τμήμα μηχανίκων Πληροφορικής. Η εργασία αυτή έχει ως στόχος να σχεδιάσει την διαδικασία λειτουργίας του λιανικού και χονδρικού εμπορίου μιας αλυσίδας σούπερ μάρκετ στον ελλαδικό χώρο.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητής μου, κ. Τάτση Γεώργιο, για την πολύτιμη βοήθεια, καθοδήγηση και συμπαράσταση καθ' όλη την διάρκεια πραγματοποίησης της πτυχιακής. .

Την οικογένεια μου που όλα τα χρόνια της φοίτησής μου ήταν δίπλα μου και με στήριζαν οικονομικά αλλά και ψυχολογικά.

ΕΙΣΑΓΩΓΗ

Η περίπτωση της Ελληνικής αλυσίδας Supermarket e-coop.

Στην παρούσα διπλωματική εργασία χρησιμοποιείται η εικονική αλυσίδα σούπερ μάρκετ e-coop η οποία εδρεύει στην Αθήνα και έχει άλλα δύο υποκαταστήματα τα οποία επίσης βρίσκονται εντός Αττικής.
Τα είδη των ερωτήσεων (queries) που μελετήθηκαν είναι γενικά και μπορούν να εφαρμοστούν σε οποιαδήποτε αλυσίδα από σούπερ μάρκετ ή καταστήματα τα οποία διατηρούν πληροφορίες σχετικά με τους πελάτες, τους υπαλλήλους και τα προϊόντα τους.

Σύντομη περιγραφή:

Το διοικητικό συμβούλιο της αλυσίδας σούπερ μάρκετ, με έχει προσλάβει ως σχεδιαστή βάσης δεδομένων και Developer ώστε να σχεδιάσω μία Βάση Δεδομένων η οποία θα βοηθά το διοικητικό συμβούλιο να γνωρίζει τα στατιστικά στοιχεία σχετικά με :

- Τις πωλήσεις οποιασδήποτε μάρκας προϊόντων:
  - Πόσοι πελάτες αγοράζουν το κάθε ένα από αυτά;
  - Σε ποιό κατάστημα;
  - Πόσο συχνά;
  - Ποια ημέρα της εβδομάδας;
  - Ποια ώρα της ημέρας;
  - Ποια είναι τα καταστήματα που έχουν τις περισσότερες πωλήσεις;
  - Ποια είναι τα αποθέματα σε οποιαδήποτε χρονική στιγμή;
  - Ποιες μάρκες έχουν πουληθεί σε όλα τα καταστήματα;
  - Ποιες μάρκες έχουν αγοραστεί από όλους τους εργαζομένους στα καταστήματα;

- Τα καταστήματα και τους πελάτες τους:
  - Πόσους πελάτες έχει κάθε κατάστημα;
  - Ποιες είναι οι ώρες λειτουργίας του;
  - Πόσοι πελάτες εξυπηρετούνται για κάθε ώρα της ημέρας και κάθε ημέρα της εβδομάδας;

- Τα καταστήματα και τους υπαλλήλους τους:
  - Πόσοι υπάλληλοι εργάζονται;
  - Πόσοι ταμίες δουλεύουν συνολικά μία οποιαδήποτε ημέρα;

- Πόσες ταμιακές μηχανές θέτονται σε λειτουργία οποιαδήποτε ώρα και μέρα σε κάθε κατάστημα;
- Πόσοι συνολικά υπάλληλοι δεν δουλεύουν ως ταμίες συγκεκριμένη μέρα;
- Ποιοι είναι οι υπάλληλοι που έχουν εργαστεί σε περισσότερα από ένα καταστήματα;
- Ποιοι είναι οι υπάλληλοι που έχουν εργαστεί σε ένα μόνο κατάστημα;

Επέλεξα να φτιάξω μια αλυσίδα σούπερ μάρκετ η οποία βασίζεται στους επιχειρηματικούς κανόνες που ακολουθούν:

- ✓ Κάθε προϊόν έχει συγκεκριμένο αναγνωριστικό κωδικό (barcode), αλλά και όλα τα τεμάχια που αντιστοιχούν στο ίδιο προϊόν έχουν το ίδιο αναγνωριστικό.
- ✓ Κάθε προϊόν που αγοράζεται σαρώνεται από το ταμείο.
- ✓ Κάθε αγορά συνοδεύεται από απόδειξή με καταγεγραμμένη ώρα.
- ✓ Κάθε ταμίας εισάγει τον κωδικό του κατά το «άνοιγμα» του ταμείου και γίνεται καταγραφή ημερομηνίας και ώρας.
- ✓ Κάθε ταμίας αναγνωρίζεται από την ταμειακή μηχανή σύμφωνα με τον κωδικό του και την ώρα.
- ✓ Κάθε πελάτης παρουσιάζει την κάρτα μέλους στο ταμίο πριν από την σάρωση του προϊόντος.
- ✓ Κάθε εργαζόμενος σαρώνει την κάρτα του κατά την άφιξη ή αναχώρησή του από το κατάστημα.

Στόχος μας είναι να δημιουργήσουμε μια Βάση Δεδομένων στη οποία θα αποθηκεύονται τα δεδομένα που αφορούν τα καταστήματα, τους υπαλλήλους, τους πελάτες, τους ιδιοκτήτες, τα προϊόντα και θα επιτρέπουν στο διοικητικό συμβούλιο να λαμβάνει αποφάσεις με βάση τα στατιστικά στοιχεία που αναφέρονται παρά πάνω. Για να το πραγματοποιήσουμε, θα σχεδιάσουμε τα ακόλουθα στάδια:

- ✓ Λεξικό δεδομένων.
- ✓ Διάγραμμα οντοτήτων συσχετίσεων.
- ✓ Μετατροπή του διαγράμματος οντοτήτων-συσχετίσεων σε ένα σύνολο σχέσεων.
- ✓ SQL κώδικα
- ✓ Κανονικοποίηση σχέσεων.
- ✓ Αντιπροσωπευτικά ερωτήματα (queries).

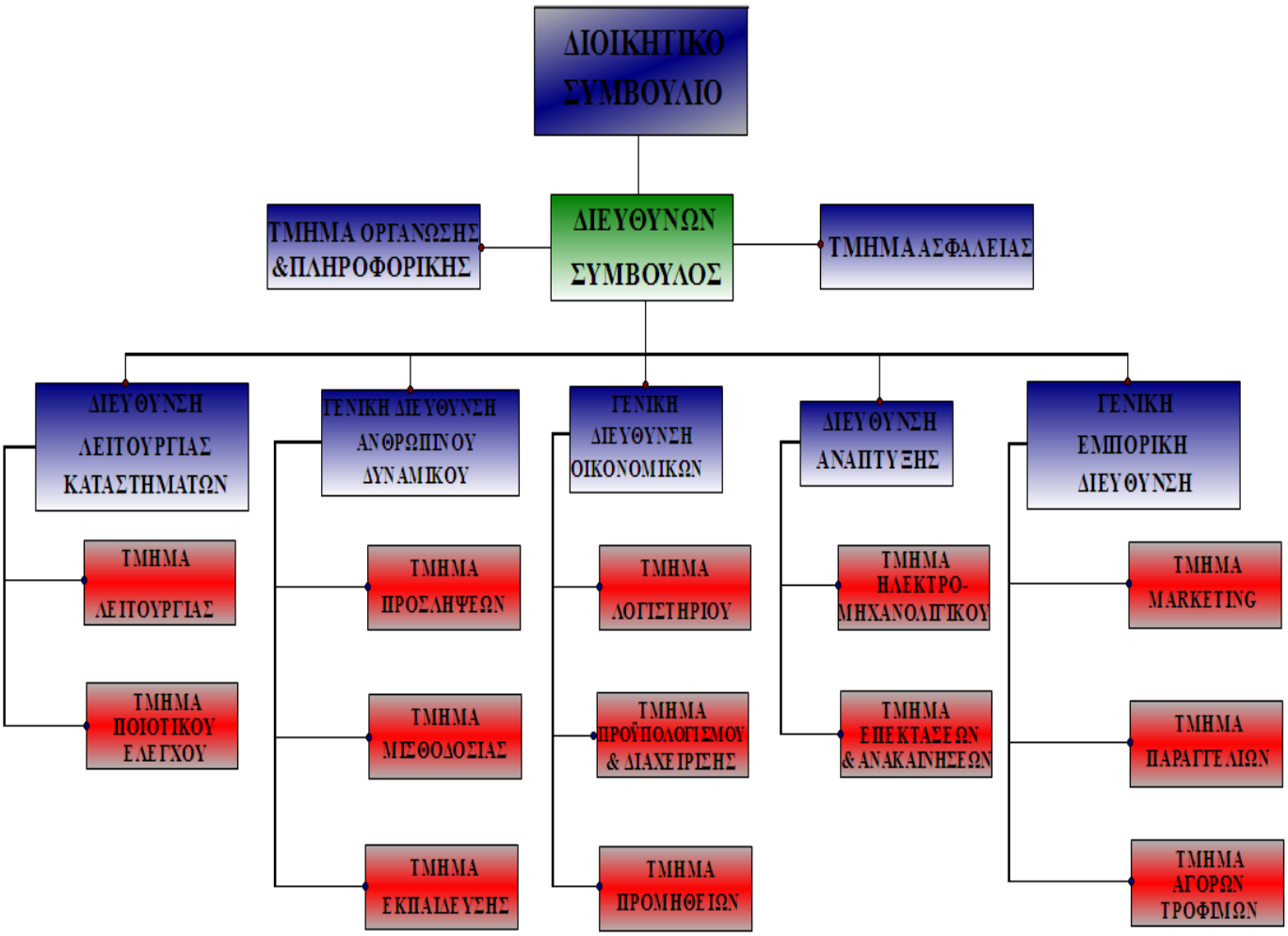
Ο όρος αλυσίδα σούπερ μάρκετ αναφέρεται στην ύπαρξη περισσοτέρων του ενός καταστημάτων του ίδιου ομίλου δηλαδή υποκαταστημάτων, τα οποία έχουν το ίδιο όνομα και προσφέρουν τα ίδια προϊόντα και υπηρεσίες στους πελάτες τους. Τα

καταστήματα αυτά μπορεί να βρίσκονται στην ίδια πόλη, σε διαφορετικές πόλεις ή ακόμα και σε διαφορετικές χώρες.

Στις περισσότερες περιπτώσεις αλυσίδων υπάρχει ένα κατάστημα, που θεωρείται ως το κεντρικό του ομίλου και στο οποίο είναι τοποθετημένα τα γραφεία του, δηλαδή το διοικητικό συμβούλιο της εταιρίας. Στις εγκαταστάσεις αυτές υπάρχουν οι διευθύνσεις και τα τμήματα τα οποία είναι απαραίτητα στο να λειτουργήσει μια αλυσίδα. Έτσι έχουμε τη **Γενική Διεύθυνση Ανθρώπινου Δυναμικού,** η οποία αποτελείται από το *Τμήμα προσλήψεων*, το οποίο είναι υπεύθυνο για τις προσλήψεις των κατάλληλων ανθρώπων στις κατάλληλες θέσεις. Το *Τμήμα μισθοδοσίας* το οποίο είναι υπεύθυνο για τη μισθοδοσία όλων των εργαζομένων. Τέλος έχει *το Τμήμα εκπαίδευσης* το οποίο είναι υπεύθυνο για την εκπαίδευση όχι μόνο όσων έχουν προσληφθεί τώρα στην εταιρία, αλλά και όσων δουλεύουν ήδη σε αυτή. Στη συνέχεια έχουμε τη **Γενική Διεύθυνση Οικονομικών** που αποτελείται από το *Τμήμα Λογιστηρίου*, που είναι υπεύθυνο για την μισθοδοσία και την τιμολόγηση των προϊόντων, το *Τμήμα Προμηθειών,* το οποίο είναι υπεύθυνο για τον έλεγχο των αποθεμάτων και τη σωστή παραγγελία των προϊόντων και το *Τμήμα Προϋπολογισμού και Διαχείρισης Διαθεσίμων*, το οποίο διαχειρίζεται τα θέματα των παραγγελιών, των νέων προϊόντων και εφοδιασμό των καταστημάτων. Έπειτα έχουμε τη **Διεύθυνση Ανάπτυξη**ς, που αποτελείται από το *Τμήμα επεκτάσεων και ανακαινίσεων*, και αφορά συνήθως τα νέα καταστήματα και το *Τμήμα Ηλεκτρομηχανολογικού εξοπλισμού*, το οποίο είναι υπεύθυνο για την εγκατάσταση και τον εξοπλισμό όλων των ηλεκτρικών μονάδων μέσα στα καταστήματα. Στη συνέχεια έχουμε τη **Γενική Εμπορική Διεύθυνση**, η οποία είναι πολύ σημαντική και αποτελείται από το *Τμήμα Παραγγελιών*, το οποίο ασχολείται με τους προμηθευτές και τις παραγγελίες που γίνονται ανάλογα με τη ζήτηση, Το *Τμήμα Marketing,* το οποίο ασχολείται με τον τομέα της προώθησης του κάθε καταστήματος αλλά και των φυλλαδίων με τις προσφορές και το *Τμήμα Αγορών Τροφίμων*, στο οποίο γίνεται η σωστή επιλογή τροφίμων με βάση κάποια κριτήρια, αλλά και στατιστικές που έχουν γίνει σε καταναλωτές. Μια άλλη πολύ σημαντική Διεύθυνση είναι αυτή της **Λειτουργίας των Καταστημάτων,** η οποία αποτελείται από το *Τμήμα Ποιοτικού Ελέγχου*, το οποίο έχει να κάνει με τον έλεγχο ποιότητας των προϊόντων, αν δηλαδή για παράδειγμα τα φρούτα από τον Χ προμηθευτή που έγινε η παραγγελία πληρούν τις προδιαγραφές και τις προτιμήσεις των πελατών και το δεύτερο τμήμα είναι αυτό της *Σωστής Λειτουργία*ς της αλυσίδας, το οποίο είναι υπεύθυνο στο να ελέγχει εβδομαδιαία τη σωστή λειτουργία των καταστημάτων, αν δηλαδή τα προϊόντα είναι σωστά τοποθετημένα στα ράφια, αν τα ταμεία είναι αρκετά, έτσι ώστε οι πελάτες να είναι ικανοποιημένοι και αν οι υπάλληλοι τηρούν σωστά το ωράριο εργασίας τους. Εν συνεχεία έχουμε την **Διεύθυνση Οργάνωσης και Πληροφορικής,** η οποία είναι υπεύθυνη έτσι ώστε να δουλεύει το λογιστήριο, τα ταμεία, οι παραγγελίες των προϊόντων και γενικά όλο το δίκτυο που συντηρεί τα καταστήματα της αλυσίδας. Στην συνέχεια υπάρχει το **Τμήμα Ασφάλειας**, το οποίο αποτελείται από τους φύλακες(security) και τις κάμερες, ούτως ώστε να αποφεύγονται οι κλοπές. Τέλος

έχουμε τον **Διευθύνοντα Σύμβουλο**, ο οποίος και ενημερώνεται καθημερινά για όλα τα θέματα που έχουν να κάνουν με την εταιρία.

Οργανωτική Δομή Εταιρίας – Οργανόγραμμα:



Όπως αναφέραμε και πιο πάνω σε μία αλυσίδα, όλες οι διαδικασίες προσλήψεων, μισθοδοσίας, τιμολόγησης, διόρθωσης βλαβών, αναφοράς παραπόνων αλλά και οι λήψεις αποφάσεων για την καλύτερη λειτουργία της επιχείρησης, γίνονται μόνο από το κεντρικό κατάστημα που είναι και τα γραφεία του διευθύνοντα σύμβουλου και του διοικητικού συμβουλίου.

Σε μία αλυσίδα κανένα κατάστημα δε λειτουργεί ανεξάρτητα αλλά ισχύουν κάποιοι κανόνες που καθορίζονται από το διοικητικό συμβούλιο και τα υπεύθυνα τμήματα όπως:

- Όλες οι τιμολογήσεις των προϊόντων.
- Όλα τα προϊόντα που βγαίνουν σε προσφορά ανά εβδομάδα.
- Οι αλλαγές τιμών σε ορισμένα προϊόντα.
- Η επιλογή συνεργασίας με συγκεκριμένους προμηθευτές.
- Η ρύθμιση του ωραρίου κάθε καταστήματος.
- Η επιλογή συγκεκριμένου αριθμού υπαλλήλων σε κάθε κατάστημα.

Αυτό είναι ένα πρόβλημα που απασχόλησε πολύ την Διοίκηση: Η διοίκηση και τα στελέχη έπρεπε με κάποιο τρόπο να ενημερώνονται αυτόματα και καθημερινά για όλες τις κινήσεις των καταστημάτων της αλυσίδας. Όταν υπάρχει ένας μεγάλος αριθμός από καταστήματα, που το καθένα εξυπηρετεί διαφορετική πελατεία, αλλάζουν συνεχώς οι ανάγκες και οι απαιτήσεις του. Έτσι αποφασίστηκε από το διοικητικό συμβούλιο να δημιουργηθεί μία βάση δεδομένων στη οποία θα αποθηκεύονται όλα όσα έχουν να κάνουν με τα καταστήματα, τους υπαλλήλους, τους πελάτες, τους ιδιοκτήτες και τα προϊόντα και με αυτό τον τρόπο θα επιτρέπεται στην διοίκηση να πάρει αποφάσεις σχετικά με τη λειτουργία της αλυσίδας με βάση τα στατιστικά στοιχεία που θα έχει στα χέρια της.

Οι παράγοντες που θα ληφθούν υπόψη για τη δημιουργία αυτού του project είναι:

- Η θέση που βρίσκεται το κάθε κατάστημα.
- Τα προϊόντα που διαθέτει.
- Ο αριθμός των υπαλλήλων που δουλεύουν σε κάθε κατάστημα.
- Ο αριθμός των ταμείων που υπάρχουν κάθε δεδομένη στιγμή και
- Οι ώρες λειτουργίας του καταστήματος.

Αυτή τη στιγμή η αλυσίδα διαθέτει τρία καταστήματα στην Αττική, ένα στην Παλλήνη, και δύο στην Αθήνα Το κατάστημα της Παλλήνης που είναι και το μεγαλύτερο από τα άλλα δύο, είναι ανοιχτό από τις οκτώ και μισή το πρωί μέχρι τις εννιά το βράδυ, αποτελείται από εικοσιπέντε άτομα. Από αυτά τα εικοσιπέντε άτομα, δύο αποτελούν τα διευθυντικά στελέχη του, δηλαδή το ένα είναι ο διευθυντής και το άλλο ο υποδιευθυντής , δύο άτομα είναι οι υπεύθυνοι ταμείων, ένας για κάθε βάρδια, τέσσερις είναι οι αποθηκάριοι, δυο για κάθε βάρδια, οκτώ είναι οι ταμίες , δηλαδή τέσσερις για κάθε βάρδια και τα υπόλοιπα εννιά άτομα είναι οι πωλητές αυτοί δηλαδή που γεμίζουν και στήνουν τους διαδρόμους του καταστήματος, αδειάζουν τις παλέτες με τα προϊόντα και τα τοποθετούν στα ράφια, βλέπουν ποια προϊόντα είναι σε έλλειψη οπότε και τα σημειώνουν έτσι ώστε να δώσουν παραγγελία και να εφοδιαστούν οι διάδρομοι άμεσα. Μάλιστα κάποια από αυτά τα άτομα έχουν και ιδιότητα ταμία έτσι ώστε τις ημέρες αλλά και ώρες αιχμής να εξυπηρετούνται πιο γρήγορα οι πελάτες. Τα άλλα δύο καταστήματα είναι πιο μικρά από αυτό της Παλλήνης και οι ώρες λειτουργίας τους είναι από τις εννιά το πρωί μέχρι τις εννιά το βράδυ. Το προσωπικό τους αποτελείται από δεκαπέντε άτομα στο καθένα αντίστοιχα, τα δυο άτομα είναι ο διευθυντής και ο υποδιευθυντής, δύο είναι υπεύθυνοι των ταμείων ένας για κάθε βάρδια, δύο είναι οι αποθηκάριοι ένας για κάθε βάρδια επίσης, τέσσερις ταμίες δηλαδή δύο για κάθε βάρδια και οι υπόλοιποι πέντε είναι οι πωλητές. Σε κάποιες περιπτώσεις όμως όπως σε περίοδο προσφορών, γιορτών (Χριστούγεννα η Πάσχα), μεσολάβηση Αργίας ή ακόμα και κάθε Σάββατο η κίνηση είναι αυξημένη κατά πολύ στα καταστήματα, θα χρειαστεί λοιπόν κάποιοι από τους εργαζομένους να φεύγουν από το δικό τους κατάστημα το οποίο ενδεχομένως δεν έχει τόση δουλειά και να πηγαίνουν σε κάποιο άλλο.

Με αυτά τα δεδομένα συνθέσαμε ένα project που αποτελείται από τα παρακάτω:

- Λεξικό Δεδομένων.
- Διάγραμμα Οντοτήτων Συσχετίσεων.
- Μετασχηματισμό από Εννοιολογικό Μοντέλο σε Λογικό.
- Κανόνες Μετασχηματισμού.
- Μετατροπή του Εννοιολογικού Μοντέλου σε πίνακες και υλοποίηση σε mysql.

- Επαλήθευση πινάκων - Κανονικοποίηση από Εξαρτήσεις.
- Ερωτήματα Σχεσιακής Άλγεβρας.
- Εισαγωγή Εγγραφών στους Πίνακες.
- Ερωτήματα (queries).

ΚΕΦΑΛΑΙΟ 1: ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ & ΕΝΝΟΙΟΛΟΓΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

1. Λεξικό Δεδομένων

- ***Εργαζόμενος***: ένα άτομο το οποίο εργάζεται για ένα άλλο άνθρωπο ή για κάποια επιχείρηση.
    - Ένας εργαζόμενος έχει μοναδικό κωδικό αναγνώρισης , όνομα-επίθετο και μοναδικό αριθμό ασφάλισης.
    - Κάθε άτομο που εργάζεται στην επιχείρηση πρέπει να διαθέτει μια κάρτα με την οποία μέσω μιας διαδικασίας που ονομάζεται scan, ενημερώνει το σύστημα με τις ώρες που εισέρχεται στο κατάστημα και τις ώρες που τελειώνει η βάρδια του. (start_work timestamp, end_work timestamp)
    - Κάθε φορά που κάποιος εργαζόμενος ενημερώνει τη Βάση δεδομένων με τη άφιξη του, μια μεταβλητή του συστήματος αυξάνεται κατά μία μονάδα. (employee_id)
    - Κάθε ταμίας διαθέτει ένα κωδικό με τον οποίο μπορεί να θέσει σε λειτουργία το ταμείο στο οποίο εργάζεται.
    - Κάθε ταμίας αναγνωρίζεται από το ταμείο με βάση τον κωδικό του και τις ώρες εργασίας του.
    - Οι υπάλληλοι μπορεί να είναι και πελάτες.
    - Οι υπάλληλοι εργάζονται σε καταστήματα.
    - Οι υπάλληλοι μπορεί να εργάζονται σε διαφορετικά καταστήματα.
    - Μερικοί υπάλληλοι δουλεύουν στα ταμεία.

- ***Ταμείο***: είναι ένα μηχάνημα με πληκτρολόγιο που λειτουργεί με ένα πρόγραμμα. Το πρόγραμμα είναι ρυθμισμένο με τέτοιο τρόπο ώστε να εμφανίζει το συνολικό κόστος μιας αγοράς αλλά και το ποσό των εισπράξεων.
    - Κάθε ταμείο έχει ένα αριθμό και ένα μοναδικό κωδικό (cache_id) ο οποίος αυξάνει αυτόματα μια μεταβλητή του συστήματος όταν προστίθεται στη Βάση Δεδομένων ένα ταμείο που αρχίζει να λειτουργεί.
    - Κάθε αριθμός ταμείου είναι μοναδικός για το κατάστημα
    - Κάποιοι υπάλληλοι δουλεύουν στα ταμεία.
    - Κάθε ταμίας, για να ανοίξει ένα ταμείο, εισάγει το κωδικό του ενώ καταγράφεται η ημερομηνία και η ώρα.
    - Τα ταμεία υπάρχουν στα καταστήματα
    - Το ταμείο αναγνωρίζει ένα προϊόν και ενημερώνει το σύστημα μέσω της διαδικασίας scan.

- ***Κατάστημα***: μια εγκατάσταση όπου πωλείται εμπόρευμα σε λιανική τιμή συνήθως.
    - Κάθε κατάστημα έχει όνομα, τη πόλη στην οποία βρίσκεται, την ώρα που ξεκινά η λειτουργία του, την ώρα που κλείνει και ένα μοναδικό κωδικό(shop_id) ο οποίος αυξάνει μια μεταβλητή του συστήματος όταν ένα καινούριο κατάστημα προστίθεται στη Βάση Δεδομένων.
    - Κάθε κατάστημα έχει ταμεία που αναγνωρίζονται από τον κωδικό αριθμό τους(cache_id).
    - Οι εργαζόμενοι δουλεύουν σε καταστήματα.
    - ΟΙ πελάτες αγοράζουν στα καταστήματα.
    - Τα καταστήματα πωλούν προϊόντα.

- ***Πελάτης***: είναι ένα άτομο που αγοράζει αγαθά και υπηρεσίες από άλλον.
    - Ο πελάτης περιγράφεται από το όνομα, μια κάρτα που περιέχει ένα αριθμό και ένα μοναδικό κωδικό(client_id) ο οποίος αυξάνει κατά μία μονάδα μία μεταβλητή του συστήματος όταν προστίθεται ένας καινούριος πελάτης στη Βάση Δεδομένων.
    - Κάθε πελάτης, που είναι μέλος, και θέλει να αγοράσει κάποια προϊόντα, πρέπει να επιδείξει τη κάρτα του στο ταμείο σύμφωνα με την οποία αποδεικνύει ότι είναι μέλος του καταστήματος . Αυτό πρέπει να γίνει πριν αναγνωριστούν τα προϊόντα από το σύστημα και ενημερωθεί.
    - Οι πελάτες αγοράζουν στα καταστήματα.

- ***Προϊόν***: ένα σύνολο από αγαθά και υπηρεσίες τα οποία κάνει διαθέσιμα μια εταιρία προς τους καταναλωτές.
    - Το προϊόν έχει ένα κωδικό(Barcode) , μια τιμή και όνομα.
    - Κάθε προϊόν έχει ένα ξεχωριστό κωδικό (barcode) αλλά όλα τα τεμάχια που αντιστοιχούν στο ίδιο προϊόν, έχουν ίδιο κωδικό (barcode).
    - Κάθε προϊόν που αγοράζεται, αναγνωρίζεται από το σύστημα και ενημερώνεται από τον ταμία(scan).
    - Τα προϊόντα πωλούνται στα καταστήματα.

- ***Απόθεμα***: σύνολο των προϊόντων που παραμένουν σε ετοιμότητα ούτως ώστε να πωληθούν.

- Το απόθεμα για κάθε προϊόν έχει ημερομηνία, ώρα και ποσό.
- Τα αποθέματα περιλαμβάνουν προϊόντα.

- ***Αγορά***: Για την απόκτηση κάποιον αγαθών χρειάζεται η καταβολή χρημάτων
  - Η αγορά προϊόντων έχει ημερομηνία.
  - Κάθε αγορά συνοδεύεται με απόδειξη στην οποία καταγράφεται η ώρα που εκδόθηκε.
  - Η αγορά περιλαμβάνει ένα σύνολο προϊόντων.

- ***Μάρκα***: Εμπορική επωνυμία ή εμπορικό σήμα
  - Έχει μοναδικό κωδικό (Brand_id) και όνομα.
  - Κάθε επώνυμη εταιρία ή μη, κατασκευάζει προϊόντα.

- ***Εργασία*****:** Σωματική και πνευματική απασχόληση του ανθρώπου για παραγωγικό σκοπό.

- ***Πώληση***: Μεταφορά προϊόντων ή προσφορά υπηρεσιών σε άλλους με χρηματική συναλλαγή.

- ***Βιομηχανία***: Χειρωνακτική ή μηχανική παραγωγή προϊόντων σε μεγάλη κλίμακα.

- ***Αγορά*****:** Επίσκεψη σε καταστήματα για αγορά ή εξέταση αγαθών.

ΙΙ. Διάγραμμα Οντοτήτων Συσχετίσεων

ΚΕΦΑΛΑΙΟ 2: ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ ΑΠΟ ΕΝΝΟΙΟΛΟΓΙΚΟ ΜΟΝΤΕΛΟ ΣΕ ΛΟΓΙΚΟ

Ι. Κανόνες Μετασχηματισμού

Κάθε τύπος μίας οντότητας αναπαρίσταται από ένα σχήμα σχέδιο ή πίνακας. Το όνομα του σχήματος είναι ίδιο με το όνομα της οντότητας. Η σχέση του σχήματος καθορίζεται από το όνομα και το τύπο της κάθε ιδιότητας που έχει αλλά και τους πε ιο κλ να ξε ση το

Ει
Ιδ
Ει
Ν

Surname: varchar, not null
Code: integer, not null, unique

Cache:
Ιδιότητες, τύπος και περιορισμοί:
Cache id: integer, not null, unique, auto increment, primary key(κύριο κλειδί)
Cache number : integer, not null, unique

Client:
Idclient : integer, not null, unique, auto increment, primary key(κύριο κλειδί)
Card_number : integer, not null, unique
Name : varchar
Surname: varchar, not null

Shop:
Shop_id : integer, not null, unique, auto increment, primary key(κύριο κλειδί)
City : varchar
Name : varchar
Opening_time : time
Closing_time : time

Product:
Barcode : varchar, not null, unique, primary key(κύριο κλειδί)
Price : integer, not null, unsigned
Name : varchar, not null

Brand:
Brand_id : integer, not null, unique, primary key(κύριο κλειδί)
Name : varchar, not null

Stock:
Stock_timestamp : timestamp, not null, unique, primary key(κύριο κλειδί)
Amount : not null, unsigned

Purchase:
Purchase_id:int, not null, auto increment, primary key(κύριο κλειδί)
Purchase_timestamp: timestamp, not null

Για κάθε σχέση 1 προς 1, το ξένο κλειδί προστίθεται σε έναν μόνο από τους δύο πίνακες.

Employee-client :idclient προστίθεται ως ξένο κλειδί στο πίνακα Employee.

Για σχέση ένα προς πολλά, το ξένο κλειδί προστίθεται στη μεριά του πίνακα που αντιπροσωπεύει τη σχέση των πολλών.

Client-purchase : idclient προστίθεται ως ξένο κλειδί στον πίνακα purchase

Purchase-cache: cache_id προστίθεται ως ξένο κλειδί στον πίνακα purchase

Purchase-Shop: shop_id προστίθεται ως ξένο κλειδί στον πίνακα purchase

Shop-cache : shop_id προστίθεται ως ξένο κλειδί στον πίνακα cache

Για σχέσεις πολλά προς πολλά, δημιουργούμε ένα καινούριο πίνακα ο οποίος θα περιέχει τα κύρια κλειδιά από τις δύο οντότητες, ως ιδιότητες πλέον. Ο πίνακας μπορεί να περιέχει μερικές ακόμα ιδιότητες.

Employee-shop : works_in_shop
Primary key: employee_id, shop_id, start_work
Ιδιότητες, τύπος και περιορισμοί:
start_work, timestamp, not null
end_work, timestamp, not null

Employee-cache: works_in_cache
Primary key: employee_id, cache_id, start_timestamp
Ιδιότητες, τύπος και περιορισμοί:
start_timestamp, timestamp, not null
end_timestamp, timestamp, not null

Shop-client: shops_in
Primary key: idclient, shop_id

Shop-product: sells
Primary key: shop_id, barcode

Product-brand: manufactures
Primary key: barcode, brand_id

Product-stock : product_stock_is_included
Primary key: barcode, stock_timestamp

Product-purchase : product_purchase_is_included
Primary key: purchase_id, barcode
Ιδιότητες, τύπος και περιορισμοί:
amount, int, not null

ΙΙ. Μετατροπή του Εννοιολογικού Μοντέλου μέσω της MySQL.

Δημιουργία Βάσης Δεδομένων με όνομα Myb:

```
create database Myb;
```

Για να χρησιμοποιήσουμε τη Βάση Δεδομένων πληκτρολογούμε την εντολή:

```
use Myb;
```

Δημιουργία του πίνακα client στη Βάση Δεδομένων Myb:

```
create table client (idclient int not null AUTO_INCREMENT,
        card_number int unsigned not null, name varchar(45), surname
        varchar(45),
primary key(idclient);
```

Δημιουργία του πίνακα Employee στη Βάση Δεδομένων Myb:

```
create table Employee (employee_id  int  unsigned not null auto_increment,
        name varchar(45) not null, surname varchar(45) not null,
        code int unsigned not null, idclient int not null,
primary key(employee_id),
foreign key (idclient) references client(idclient));
```

Δημιουργία του πίνακα Shop στη Βάση Δεδομένων Myb:

```
create table Shop (shop_id  int  not null auto_increment,
        city  varchar(45) null, name varchar(45) null, opening_time time null,
        closing_time time null,
primary key(shop_id));
```

Δημιουργία του πίνακα Cache στη Βάση Δεδομένων Myb:

```
create table Cache (cache_id int  not null auto_increment ,
        Cache_number int unsigned not null , shop_id int not null ,
primary key (cache_id) ,
foreign key (shop_id) references Shop (shop_id));
```

Δημιουργία του πίνακα Product στη Βάση Δεδομένων Myb:

```
create table Product (barcode varchar (50) not null,
        price int unsigned not null, name  varchar (45) not null,
primary key(barcode));
```

Δημιουργία του πίνακα Brand στη Βάση Δεδομένων Myb:

```
create table Brand (brand_id  int  not null auto_increment ,
        name  varchar (45) not null,
primary key (brand_id));
```

Δημιουργία του πίνακα Purchase στη Βάση Δεδομένων Myb:

```
create table Purchase (purchase_id  int  not null auto_increment,
        idclient  int  not null, shop_id  int  not null, cache_id  int  not null,
        purchase_timestamp timestamp not null,
primary key (purchase_id) ,
foreign key (cache_id ) references Cache (cache_id ),
foreign key (idclient )  references client (idclient ),
foreign key (shop_id ) references Shop (shop_id ));
```

Δημιουργία του πίνακα Stock στη Βάση Δεδομένων Myb:

```
create table Stock (stock_timestamp timestamp not null,
        amount  int unsigned not null,
primary key(stock_timestamp) );
```

Δημιουργία του πίνακα Works_in_cache στη Βάση Δεδομένων Myb:

```
create table Works_in_cache (employee_id  int unsigned not null,
        cache_id  int null, start_timestamp timestamp not null,
        end_timestamp timestamp not null,
primary key (employee_id, cache_id, start_timestamp),
foreign key (employee_id ) references Employee (employee_id ),
foreign key (cache_id ) references Cache (cache_id));
```

Δημιουργία του πίνακα works_in_shop στη Βάση Δεδομένων Myb:

```
create table Works_in_shop (employee_id  int unsigned not null,
        shop_id int not null,start_work  timestamp not null,
        end_work  timestamp not null,
primary key (employee_id, shop_id, start_work) ,
foreign key (employee_id ) references Employee (employee_id ),
foreign key (shop_id ) references Shop (shop_id ));
```

Δημιουργία του πίνακα Product_Purchase_is_included στη Βάση Δεδομένων Myb:

```
create table Product_Purchase_is_included (purchase_id int not null,
        barcode varchar(50) not null , amount  int unsigned not null,
primary key (purchase_id, barcode) ,
foreign key(purchase_id ) references Purchase (purchase_id ),
foreign key (barcode ) references Product(barcode ));
```

Δημιουργία του πίνακα sells στη Βάση Δεδομένων Myb:

```
create table Sells (shop_id  int not null, barcode varchar (50) not null,
primary key (shop_id, barcode) ,
```

```
foreign key (shop_id ) references Shop (shop_id ),
foreign key (barcode ) references Product(barcode ));
```

Δημιουργία του πίνακα manufactures στη Βάση Δεδομένων Myb:

```
create table Manufactures (barcode varchar (50) not null ,
        brand_id  int not null,
primary key (barcode, brand_id),
foreign key (barcode ) references Product (barcode ),
foreign key (brand_id ) references Brand (brand_id ));
```

Δημιουργία του πίνακα Product_Stock_is_included στη Βάση Δεδομένων Myb:

```
create table Product_Stock_is_included (barcode  varchar (50) not null,
        stock_timestamp timestamp not null,
  primary key (barcode, stock_timestamp) ,
  foreign key (barcode ) references Product (barcode ),
  foreign key (stock_timestamp ) references Stock (stock_timestamp ));
```

Δημιουργία του πίνακα shops_in στη Βάση Δεδομένων Myb:

```
create table shops_in (idclient int not null, shop_id  int not null,
primary key (idclient, shop_id) ,
foreign key (idclient ) references client (idclient ),
foreign key (shop_id )  references Shop(shop_id));
```

ΚΕΦΑΛΑΙΟ 3: ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ ΚΑΙ ΣΥΝΑΡΤΗΣΙΑΚΕΣ ΕΞΑΡΤΗΣΕΙΣ

I. Relvars και Εξαρτήσεις

Στις σχεσιακές Βάσεις Δεδομένων, ο όρος relvar, αποτελείται από δυο έννοιες δηλαδή relation variable που σημαίνει σχεσιακή μεταβλητή και διαφοροποιείται μεταξύ της μεταβλητής η οποία περιέχει μια σχέση και της ίδιας της μεταβλητής.

Άλλες βάσεις δεδομένων χρησιμοποιούν τον όρο relation και για τη μεταβλητή αλλά και για τα στοιχεία που περιέχει.

Κάθε οντότητα μετασχηματίζεται σε relval.

Κάθε σχέση πολλά προς πολλά μετασχηματίζεται σε relvar, όπου το κύριο κλειδί κάθε οντότητας που υπάρχει στη σχέση, περιλαμβάνεται στο κύριο κλειδί που αντιπροσωπεύει τη relvar σχέση πολλά προς πολλά.

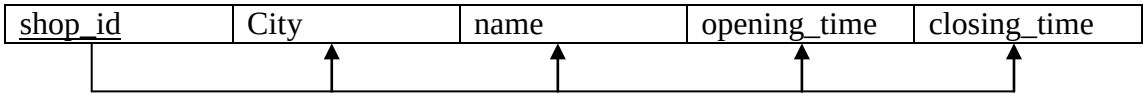
Κάθε σχέση ένα προς πολλά μετασχηματίζεται σε ξένο κλειδί στο πίνακα του ενός παραπέμποντας το κύριο κλειδί στην οντότητα των πολλών.

Κάθε σχέση ένα προς ένα μετασχηματίζεται σε ξένο κλειδί σε έναν από τους πίνακες που περιλαμβάνονται.

Εξαρτήσεις

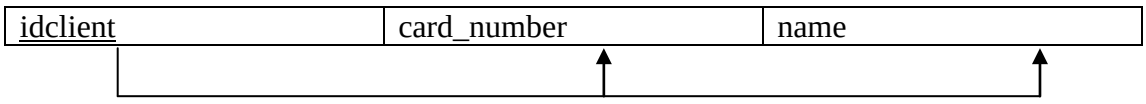
- Οι συναρτήσεις εξαρτήσεων είναι οι εξαρτήσεις από το κύριο κλειδί σε κάθε μία από τις υπόλοιπες ιδιότητες. Πιο συγκεκριμένα:

Shop



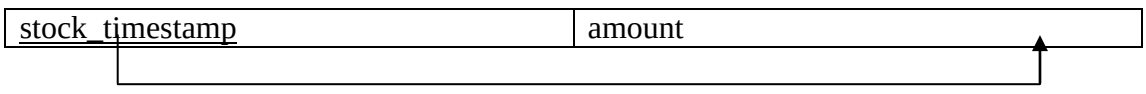
shop_id →{city, name, opening_time, closing_time}

Client



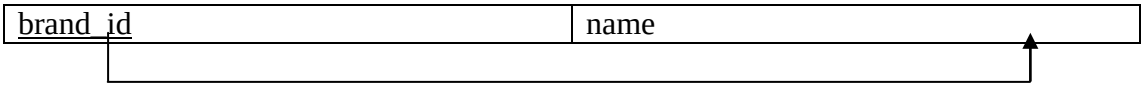
idclient → {card_number, name}

Stock



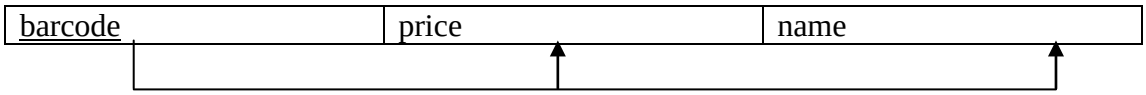
stock_timestamp → {amount}

Brand



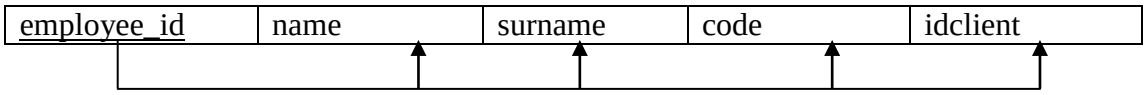
brand_id → {name}

Product



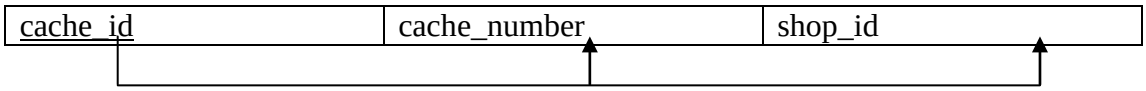
product_barcode → {price, name}

Employee



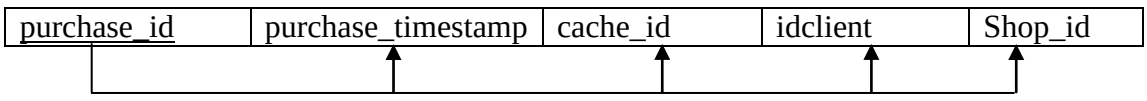
employee_id → {name, surname, code, idclient}

Cache



cache_id → {Cache_number, shop_id}

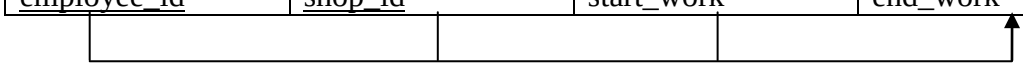
Purchase



purchase_id → {purchase_timestamp, shop_id, cache_id, idclient}

Works_in_shop

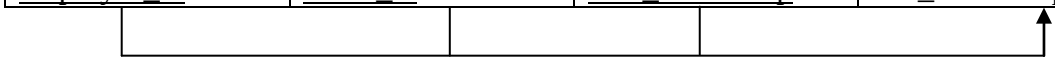
employee_id	shop_id	start_work	end_work



employee_id, shop_id, start_work → {end_work}

Works_in_cache

employee_id	cache_id	start_timestamp	end_timestamp



Employee_id, cache_id, start_timestamp → {end_timestamp}

Shops_in

Idclient	shop_id

idclient, Shop_id →

Sells

shop_id	Barcode

Shop_id, barcode →

Manufactures

barcode	brand_id

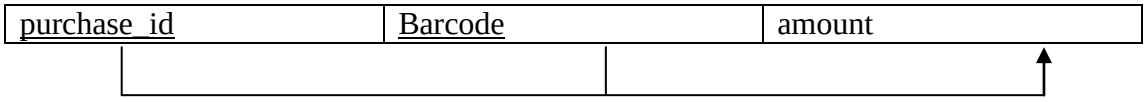
barcode, Brand_ id →

Product_stock_is_included

Barcode	stock_timestamp

barcode , Stock_ timestamp →

Product_purchase_is_included



Purchase_id, Product_barcode → {amount}

- Δεν υπάρχουν εξαρτήσεις με πολλαπλές τιμές.

II. Κανονικοποίηση

Στις σχεσιακές Βάσεις Δεδομένων, όλοι οι πίνακες θα πρέπει να είναι κανονικοποιημένοι , δηλαδή να μη περιλαμβάνουν επαναλαμβανόμενες ομάδες δεδομένων και να μην υπάρχουν πεδία που παίρνουν πολλαπλές ιδιότητες.

Οι πίνακες πρέπει να ελέγχονται σύμφωνα με κάποιους κανόνες.

Σκοπός της κανονικοποίησης είναι να μετασχηματίσουμε ένα σχεσιακό σχήμα σε μια νέα μορφή η οποία θα είναι απαλλαγμένη από ανωμαλίες εισαγωγής, διαγραφής και τροποποίησης εγγραφών.

Με την έννοια πολλαπλές ιδιότητες εννοούμε τα γνωρίσματα που έχουν καμία, μία ή πολλές ιδιότητες.

Υπάρχουν 5 κανονικές μορφές(1NF-5NF) στις οποίες μπορεί να βρεθεί η Βάση Δεδομένων. Οι τρείς πρώτες ορίστηκαν από τον Edgar Frank "Ted" Codd ενώ οι άλλες ορίστηκαν από τον Ronald Fagin.

Δεν υπάρχουν πεδία με πολλαπλές ιδιότητες ή φωλιασμένες σχέσεις άρα και οι σχέσεις στους πίνακες είναι στη μορφή 1NF.

Σε σχέσεις όπου το πρωτεύον κλειδί αποτελείται από πολλές ιδιότητες δεν υπάρχει συναρτησιακή εξάρτηση μεταξύ μέρους του κλειδιού και των υπόλοιπων ιδιοτήτων .Οι πίνακες είναι σε 1NF και κανένα πεδίο δεν είναι μερικώς εξαρτώμενο από το υποψήφιο κύριο κλειδί, υπάρχει πλήρη εξάρτηση. Άρα. οι πίνακες είναι σε 2NF.

Δεν υπάρχει σχέση στην οποία μια ιδιότητα, που δεν είναι το κύριο κλειδί, να προσδιορίζει συναρτησιακά ένα άλλο πεδίο, το οποίο επίσης δεν αφορά το κλειδί. Άρα, όλοι οι πίνακες είναι στην μορφή 3NF.

Κάθε σχέση στην BCNF είναι επίσης και στην 3NF, αλλά δεν ισχύει πάντα το αντίστροφο. Όλες οι συναρτησιακές εξαρτήσεις είναι εξαρτήσεις με το μοναδικό υποψήφιο κλειδί, άρα η Βάση Δεδομένων είναι σε μορφή BCNF.

Δεν υπάρχουν εξαρτήσεις πολλαπλών τιμών. Άρα οι πίνακες στη Βάση Δεδομένων είναι στη μορφή 4NF.

Το σχήμα είναι επίσης στην μορφή 5NF.

ΚΕΦΑΛΑΙΟ 4: ΣΧΗΜΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

I. Πίνακες της Βάσης Δεδομένων

Shop

shop_id	City	name	opening_time	closing_time

Shop_id: INT
City: VARCHAR(45)
Name: varchar(45)
Opening_time: time
Closing_time: time

Brand

brand_id	Name

Brand_id: int
Name: varchar(45)

Stock

stock_timestamp	Amount

Stock_timestamp: timestamp
Amount: int

Product

Barcode	Price	name

Barcode: varchar(45)
Price: int
Name: varchar(45)

Client

Idclient	card_number	Name	surname

Idclient: int
Card_number: int
Name: varchar(45)
surname: varchar(45)

Employee

employee_id	Name	surname	code	idclient

Employee_id: int
Name: varchar(45)
surname: varchar(45)
Code: int
Idclient: int

Cache

cache_id	cache_number	shop_id

Cache_id: int
Cache_number: int
Shop_id: int

Purchase

Purchase_id	purchase_timestamp	cache_id	idclient	Shop_id

Purchase_id: int
Purchase_timestamp: timestamp
Cache_id: int
Idclient: int
Shop_id: int

Works_in_shop

employee_id	shop_id	start_work	end_work

Employee_id: int
Shop_id: int

Start_work: timestamp
End_work: timestamp

Works_in_cache

employee_id	cache_id	start_timestamp	end_timestamp

Employee_id: int
Cache_id: int
Start_timestamp: timestamp
End_timestamp: timestamp

Shops_in

Idclient	shop_id

Idclient: int
Shop_id: int

Sells

shop_id	Barcode

Shop_id: int
Barcode: int

Manufactures

Barcode	brand_id

Barcode: varchar(45)
Brand_id: int

Product_stock_is_included

Barcode	stock_timestamp

Barcode: varchar(45)

Stock_timestamp: timestamp

Product_purchase_is_included

Purchase_id	Barcode	amount

Purchase_id: timestamp
Barcode: varchar(45)
Amount: int

ΙΙ. Ερωτήματα Σχεσιακής Άλγεβρας

Υποθέτουμε ότι οι ώρες που πρόκειται να αναφερθούν στη συνέχεια είναι της μορφής:
[έτος, μέρα, ώρα, λεπτά και δευτερόλεπτα]

Οι πωλήσεις από οποιαδήποτε μάρκα προϊόντος:

- Ποια μέρα της εβδομάδας;
  T <- Purchase * Product_purchase_is_included * Product
  T1 <- client **F** (T)
  R_{day} <- π client, name, day

- Ποια ώρα της ημέρας;
  T2 <- client,day **F** (T)
  R_{hour} <- π client, name, day, time

- Ποια είναι τα αποθέματα οποιαδήποτε χρονική στιγμή;
  S <- Stock * Product_stock_is_included * Product
  R_{stock} <- π stock_timestamp, name, amount (S)

Τα καταστήματα και οι πελάτες:

- Πόσους πελάτες έχει;
  $R_{clients}$ <- shop_id **F** COUNT idclient (shops_in)

- Ποιες είναι οι ώρες λειτουργίας;

R_{hours} <- π shop_id, name, opening_time, closing_time (Shop)

- Πόσοι πελάτες εξυπηρετούνται συγκεκριμένη χρονική στιγμή κάθε ημέρα της εβδομάδας;
  Clients between time 00-01. Similar for the rest hours of the day.
  R_{temp} <- σ (hour>00 AND hour<01) (Purchase)
  $R_{clients\text{-}hour}$ <- day **F** COUNT clientid (R_{temp})

Τα καταστήματα και οι υπάλληλοι:

- Πόσοι υπάλληλοι εργάζονται ;
  $R_{employees}$<- shop_id **F** COUNT employee_id (works_in_shop)

- Πόσοι ταμίες εργάζονται σε μία συγκεκριμένη ώρα και μέρα;
  R1 <- σ (TIME=given_time AND DAY=given_day)(works_in_cache)
  $R_{cashiers}$<-F COUNT employee_id (R1)

- Πόσα ταμεία λειτουργούν σε μία συγκεκριμένη ώρα και μέρα;
  R2 <- σ (time>start_time AND time<end_time)(works_in_cache)
  R_{cashes} <- **F** COUNT cacheid(R2)

- Πόσοι υπάλληλοι δουλεύουν ως ταμίες σε μία συγκεκριμένη ώρα και μέρα;
  R3 <- π employee_id (R2)
  R4 <- π employee_id (Employee)
  $R_{emp\_not\_assignes}$ <- R4 – R3

- Ποιοι είναι οι υπάλληλοι που έχουν δουλέψει σε περισσότερα του ενός καταστήματα;
  R5 <- Employee_id **F** COUNT shop_id (works_in_shop)
  ρ_{R5} (Employee_id, No_of_shops) (R5)
  $R_{shops>1}$ <- σ No_of_shops > 1 (R5)

- Ποιοι είναι οι υπάλληλοι που έχουν δουλέψει σε ένα μόνο κατάστημα;
  R <- σ No_of_shops=1 (R5)

ΚΕΦΑΛΑΙΟ 5: ΕΡΩΤΗΜΑΤΑ (QUERIES)

Ι. Εισαγωγή Στοιχείων

Εφόσον η Βάση Δεδομένων με τους πίνακες έχει ολοκληρωθεί, σειρά έχει να εισάγουμε εγγραφές στους πίνακες.

```
insert into client values (1, 11, 'AGAMEMNWNAS', 'ATHANASIOY ');
insert into client values (2, 22, 'BIKTWRIA', 'BASILATOY');
insert into client values (3, 33, 'ERMHS', 'TRIANTAFYLLIDHS');
insert into client values (4, 44, 'DHMHTRA', 'DASKALOPOYLOY');
insert into client values (5, 55, 'ERATOSTHENHS', 'DASKALOPOULOS');
insert into client values (6, 66, 'THEMISTOKLHS', ' KLAPAS ');
insert into client values (7, 77, 'IKAROS', 'KAMMENOS');
insert into client values (8, 88, 'KALUPSW', ' MOUSTOKSIDH ');
insert into client values (9, 99, 'KIRKH', 'KYPRIANOU');
insert into client values (10, 10, 'OMHROS', ' HLIADHS ');


insert into employee values(1, 'BIKTWRIA',  'BASILATOY', 123, 2);
insert into employee values(1, 'DHMHTRA',  'DASKALOPOYLOY', 124, 4);
insert into employee values(1, 'ERATOSTHENHS',  'DASKALOPOULOS', 125, 5);
insert into employee values(1, 'IKAROS',  'KAMMENOS', 126, 7);
insert into employee values(1, 'KALUPSW',  ' MOUSTOKSIDH ', 127, 8);


insert into shop values (1, 'PALHNH', 'PALHNH MARKET',  '08:30:00 21:00:00');
insert into shop values (2, 'ATHINA',  'CENTRAL MARKET',  '09:00:00 21:00:00');
insert into shop values (3, 'ATHINA',  'ATHINA MARKET',  '08:00:00 21:00:00');


insert into cache values(1, 1, 1);
insert into cache values(2, 2, 1);
insert into cache values(3, 1, 2);
insert into cache values(4, 1, 3);
insert into cache values(5, 2, 3);


insert into product values('111111', 2, 'PATATOES');
insert into product values('123456', 12, 'ARIEL');
insert into product values('222222', 3, 'GREEN TEA');
insert into product values('333333', 5, 'CHIKEN');
insert into product values('444444', 2, 'MILK');


insert into brand values(1, 'MARKET');
insert into brand values(2, 'THESS');
insert into brand values(3, 'GOLD');
```

```
insert into purchase values(1, 1, 1, 2, '2013-02-05 15:01:00');
insert into purchase values(2, 3, 3, 1, '2013-02-05 16:50:00');
insert into purchase values(3, 4, 1, 1, '2013-02-08 12:45:00');
insert into purchase values(4, 4, 1, 1, '2013-02-09 10:07:59');
insert into purchase values(5, 5, 1, 1, '2013-02-09 10:07:59');
insert into purchase values(6, 8, 2, 3, '2013-02-15 09:45:00');
insert into purchase values(7, 9, 2, 3, '2013-02-20 13:34:00');


insert into stock values('2013-02-04 08:30:00', 10);
insert into stock values('2013-02-06 10:30:00', 12);
insert into stock values('2013-02-09 12:00:00', 20);


insert into work_in_cache values(1, 1, '2013-01-03 09:01:00', '2013-01-03 12:00:00');
insert into work_in_cache values(1, 1, '2013-01-03 09:03:00', '2013-01-03 12:00:00');
insert into work_in_cache values(1, 2, '2013-01-04 12:01:00', '2013-01-03 15:00:00');
insert into work_in_cache values(2, 1, '2013-01-06 12:01:00', '2013-01-06 15:00:00');
insert into work_in_cache values(3, 1, '2013-01-06 16:01:00', '2013-01-06 20:00:00');
insert into work_in_cache values(4, 1, '2013-01-09 09:01:00', '2013-02-09 13:00:00');
insert into work_in_cache values(4, 3, '2013-01-09 12:01:00', '2013-02-09 15:00:00');
insert into work_in_cache values(5, 3, '2013-01-09 09:01:00', '2013-02-09 13:00:00');


insert into works_in_shop values(1, 1, '2013-02-05 08:35:00', '2013-02-05 12:00:00');
insert into works_in_shop values(1, 1, '2013-02-06 08:35:00', '2013-02-05 12:00:00');
insert into works_in_shop values(2, 3, '2013-02-09 09:00:00', '2013-02-09 13:00:00');
insert into works_in_shop values(2, 3, '2013-02-15 12:00:00', '2013-02-09 20:00:00');
insert into works_in_shop values(3, 2, '2013-01-03 08:00:00', '2013-01-03 16:00:00');
insert into works_in_shop values(4, 2, '2013-01-04 08:00:00', '2013-01-04 16:00:00');
insert into works_in_shop values(5, 3, '2013-01-06 12:00:00', '2013-01-06 20:00:00');


insert into product_purchase_is_included values(1, '111111', 1);
insert into product_purchase_is_included values(1, '2222222', 4);
insert into product_purchase_is_included values(2, '333333', 1);
insert into product_purchase_is_included values(3, '222222', 4);
insert into product_purchase_is_included values(4, '123456', 3);
insert into product_purchase_is_included values(4, '444444', 10);
insert into product_purchase_is_included values(5, '111111', 8);
insert into product_purchase_is_included values(5, '123456', 5);
insert into product_purchase_is_included values(6, '444444', 5);
insert into product_purchase_is_included values(7, '222222', 9);


insert into sells values(1, '111111');
insert into sells values(1, '123456');
insert into sells values(3, '123456');
insert into sells values(2, '222222');
insert into sells values(3, '222222');
```

```
insert into sells values(1, '333333');
insert into sells values(2, '444444');
insert into sells values(3, '444444');


insert into manufactures('111111', 1);
insert into manufactures('444444', 1);
insert into manufactures('123456', 2);
insert into manufactures('444444', 2);
insert into manufactures('123456', 3);
insert into manufactures('222222', 3);
insert into manufactures('333333', 3);


insert into product_stock_is_included values('111111' '2013-02-04 08:30:00');
insert into product_stock_is_included values('123456' '2013-02-06 10:30:00');
insert into product_stock_is_included values('444444' '2013-02-09 12:30:00');


insert into shops_in values(1, 1);
insert into shops_in values(2, 1);
insert into shops_in values(3, 1);
insert into shops_in values(6, 1);
insert into shops_in values(1, 2);
insert into shops_in values(5, 2);
insert into shops_in values(8, 2);
insert into shops_in values(2, 3);
```

ΙΙ. Ερωτήματα MYSQL

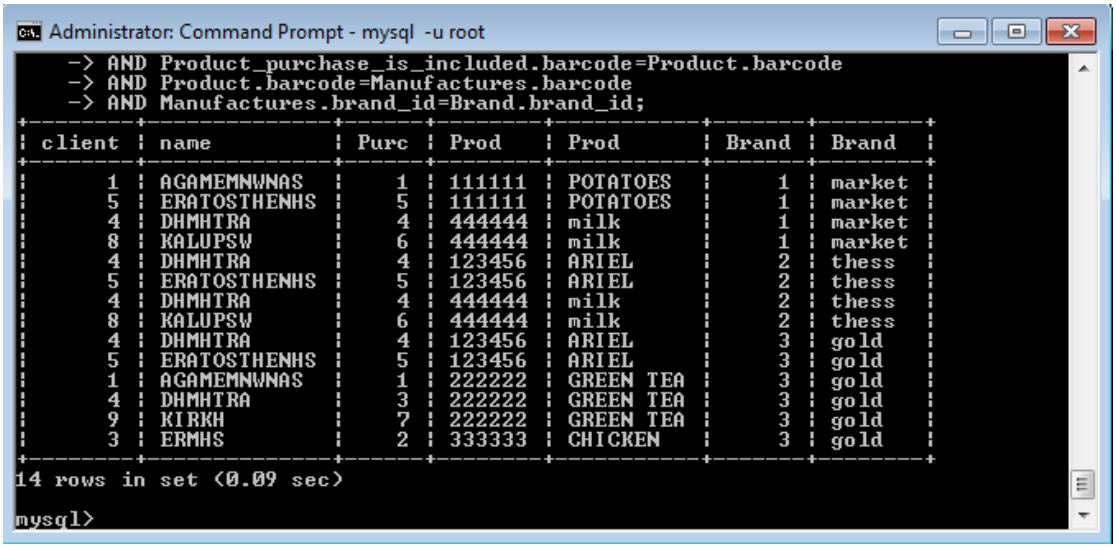
Τα Queries ή ερωτήματα, είναι ένας τρόπος που μας επιτρέπει να επιλέξουμε πληροφορίες για πεδία ή εγγραφές από ένα ή περισσότερους πίνακες. Με τα ερωτήματα μπορούμε επίσης να διαγράψουμε πεδία ή εγγραφές, να εκτελέσουμε πράξεις, να ομαδοποιήσουμε αλλά και να ταξινομήσουμε τα δεδομένα μας. Μια άλλη

δυνατότητα είναι η υποβολή κριτηρίων, δηλ, είναι περιορισμοί προκειμένου να φιλτραριστούν και να απομονωθούν τα αποτελέσματα που θέλουμε. Οι νέες πληροφορίες που προκύπτουν εμφανίζονται στην οθόνη με μορφή πίνακα χωρίς να αλλάξουν οι αρχικοί πίνακες. Εξαιρούνται βέβαια οι πίνακες που υπόκεινται σε διαγραφή εγγραφών.

ΕΡΩΤΗΜΑ 1*: Πόσοι πελάτες αγοράζουν κάθε προϊόν του καταστήματος;*

➢ Για να το πετύχουμε όμως αυτό θα πρέπει πρώτα να δούμε μερικές λεπτομέρειες όσον αφορά τα προϊόντα αλλά και τις μάρκες των προϊόντων που προτιμούν να αγοράζουν οι πελάτες, αναλυτικά.

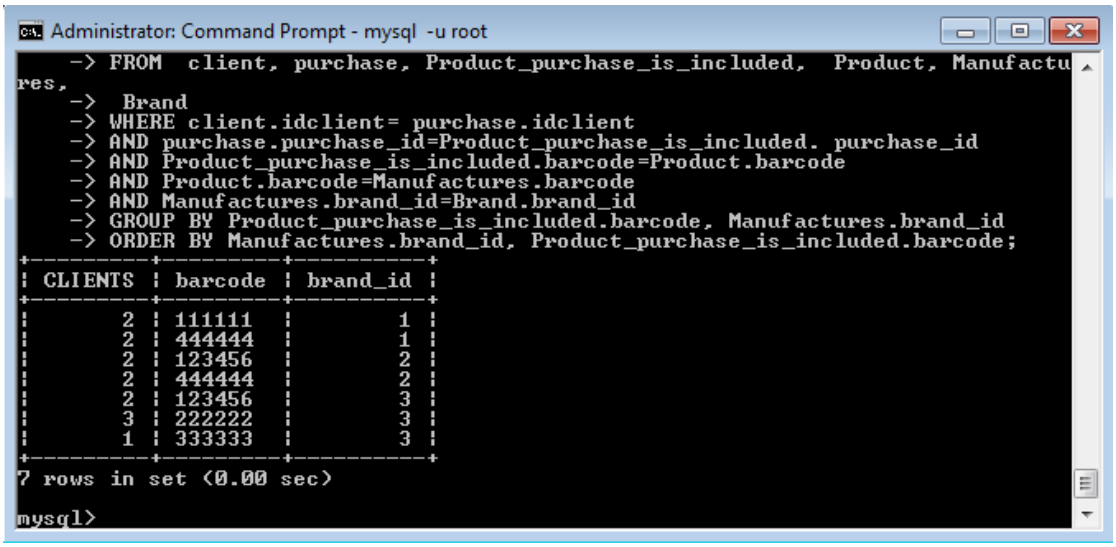
```
SELECT purchase.idclient "client",
        client.name, purchase.purchase_id "Purc",
        Product_purchase_is_included.barcode "Prod",
        Product.name "Prod",
       Manufactures.brand_id "Brand",
        Brand.name "Brand"
FROM client, purchase, Product_purchase_is_included, Product, Manufactures,
         Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id;
```



➢ Εφόσον πήραμε τα αποτελέσματα που μας ενδιαφέρουν, το επόμενο βήμα είναι να προσθέσουμε τους πελάτες ανάλογα με το κωδικό του προϊόντος και της μάρκας.

```
SELECT COUNT(client.idclient) CLIENTS,
        Product_purchase_is_included.barcode,
```

```
        Manufactures.brand_id
FROM  client, purchase, Product_purchase_is_included,  Product, Manufactures,
        Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
GROUP BY Product_purchase_is_included.barcode, Manufactures.brand_id
ORDER BY Manufactures.brand_id, Product_purchase_is_included.barcode;
```



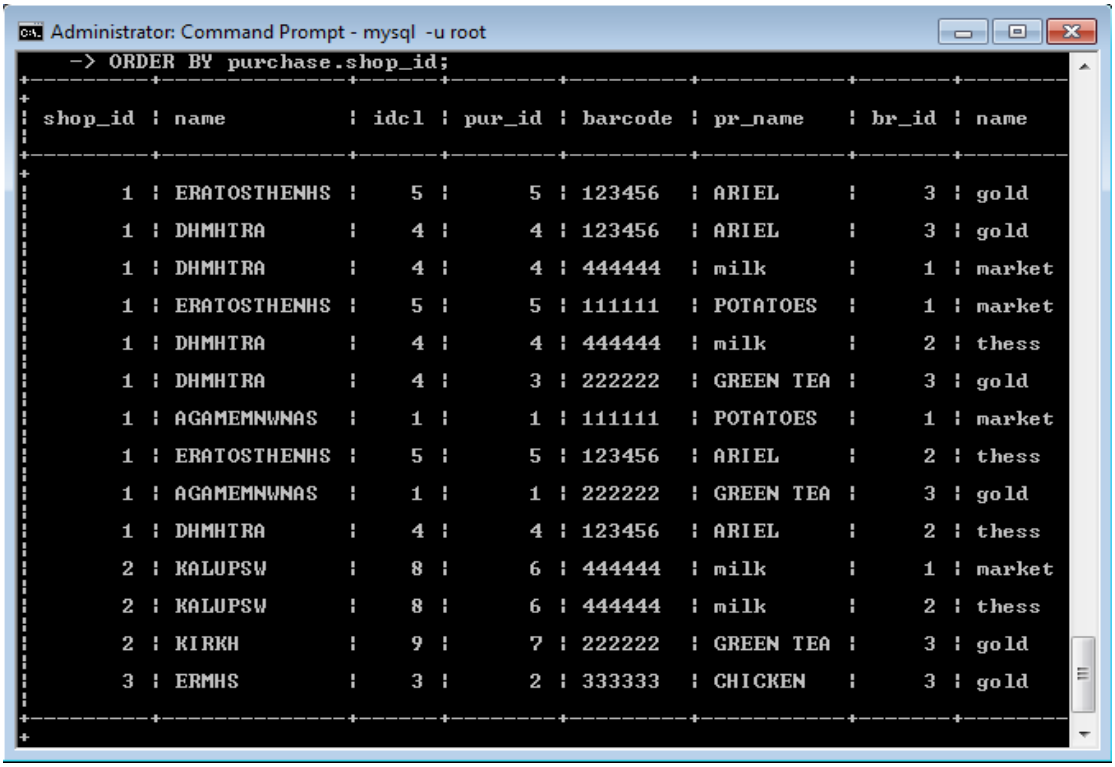
Σύμφωνα με το παράδειγμά μας βλέπουμε ότι 2 πελάτες συνολικά αγόρασαν προϊόν με κωδικό 44444444 και κωδικό μάρκας 1 ενώ 3 πελάτες συνολικά αγόρασαν προϊόν με κωδικό 222222 και κωδικό μάρκας 3.

ΕΡΩΤΗΜΑ 2: *Ποιο κατάστημα έχει τους περισσότερους πελάτες και τι αγοράζουν συνήθως;*

- Για να δούμε ποιο κατάστημα έχει τους περισσότερους πελάτες θα πρέπει πρώτα να δούμε μερικές λεπτομέρειες όπως είναι ο κωδικός του καταστήματος που αγοράζουν οι πελάτες καθώς επίσης και τις μάρκες των προϊόντων.

```
SELECT purchase.shop_id shop_id,
        client.name name , purchase.idclient idcl,
        purchase.purchase_id pur_id,
        Product_purchase_is_included.barcode barcode ,
        Product.name pr_name,
       Manufactures.brand_id br_id, Brand.name
FROM client, purchase, Product_purchase_is_included,  Product, Manufactures,
        Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
```

AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
ORDER BY purchase.shop_id;

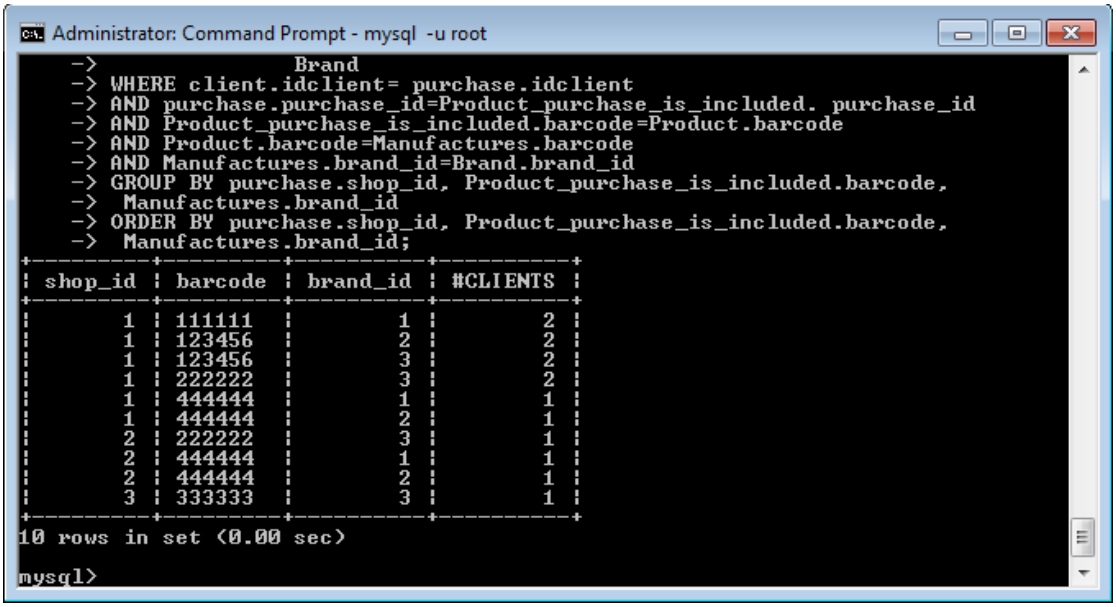


Στον πίνακα βλέπουμε ποια προϊόντα αγοράστηκαν από κάθε πελάτη αλλά και τη μάρκα που επέλεξε.

- Αφού δούμε τις λεπτομέρειες που μας ενδιαφέρουν, σειρά έχει να πάρουμε το τελικό αποτέλεσμα ώστε να δούμε ποια προϊόντα πωλούνται σε κάθε κατάστημα και πόσοι πελάτες τα αγοράζουν.

SELECT purchase.shop_id,
Product_purchase_is_included.barcode "barcode",
Manufactures.brand_id,
COUNT(client.idclient) "#CLIENTS"
FROM client, purchase, Product_purchase_is_included, Product, Manufactures,
Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
GROUP BY purchase.shop_id, Product_purchase_is_included.barcode,
Manufactures.brand_id
ORDER BY purchase.shop_id, Product_purchase_is_included.barcode,

Manufactures.brand_id;

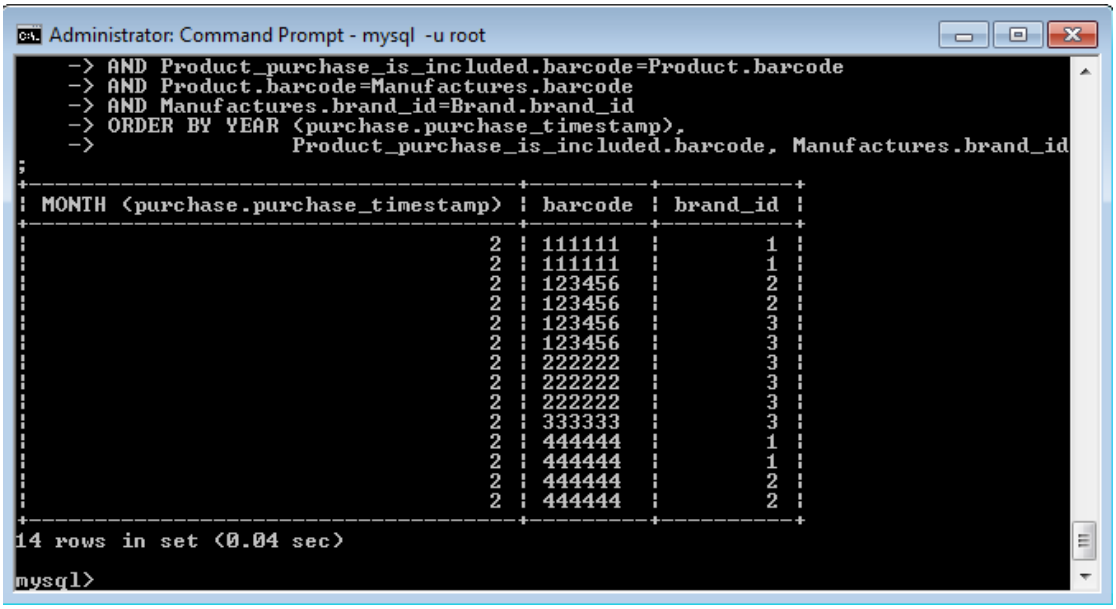


Βλέπουμε λοιπόν ότι το κατάστημα με κωδικό αριθμό 1 διαθέτει προϊόντα που έχουν κωδικό 111111,123456,222222,444444 με ανάλογο κωδικό μάρκας.
Επίσης βλέπουμε ότι 2 πελάτες έχουν αγοράσει από το κατάστημα 1 το προϊόν με κωδικό 111111 και κωδικό μάρκας 1.

ΕΡΩΤΗΜΑ 3: *Πόσο συχνά ψωνίζουν οι πελάτες;*

- Για να δούμε πόσο συχνά ψωνίζουν οι πελάτες πρέπει να δούμε πρώτα ποια προϊόντα πωλήθηκαν, σε ποιους μήνες αλλά και το κωδικό της μάρκας των προϊόντων.

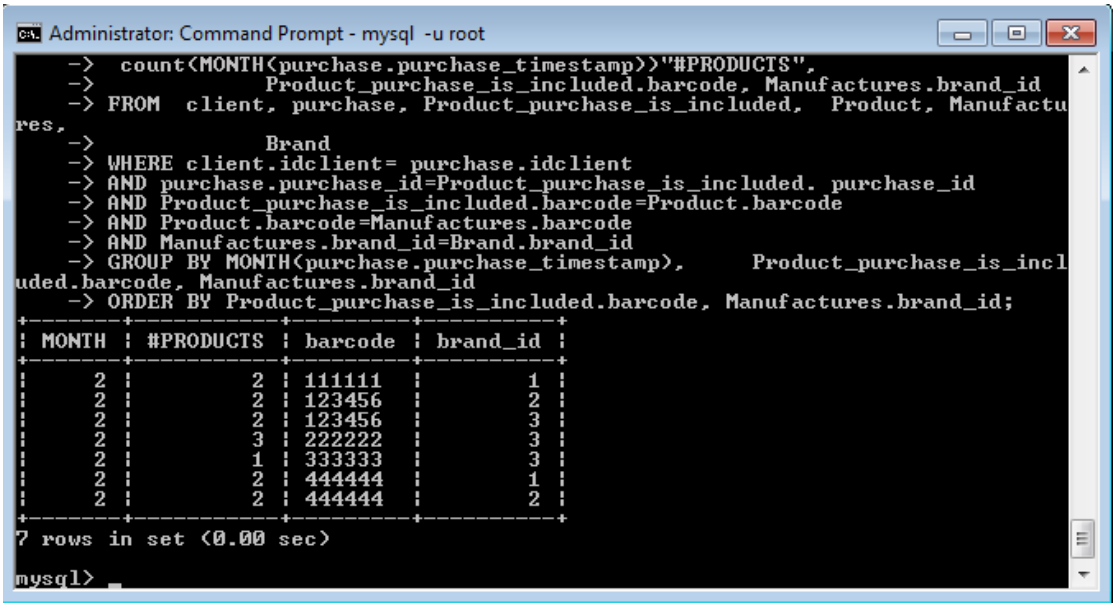
```
SELECT MONTH (purchase.purchase_timestamp),
        Product_purchase_is_included.barcode,
       Manufactures.brand_id
FROM client, purchase, Product_purchase_is_included, Product,
       Manufactures, Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
ORDER BY YEAR (purchase.purchase_timestamp),
       Product_purchase_is_included.barcode, Manufactures.brand_id;
```



Στο παράδειγμά μας βλέπουμε ποια προϊόντα πουλήθηκαν σε ποιους μήνες.

- Εφόσον πήραμε τα αποτελέσματα από το προηγούμενο ερώτημα, σειρά έχει να υπολογίσουμε αθροιστικά πόσα προϊόντα πουλήθηκαν ανά κωδικό, κάθε μήνα.

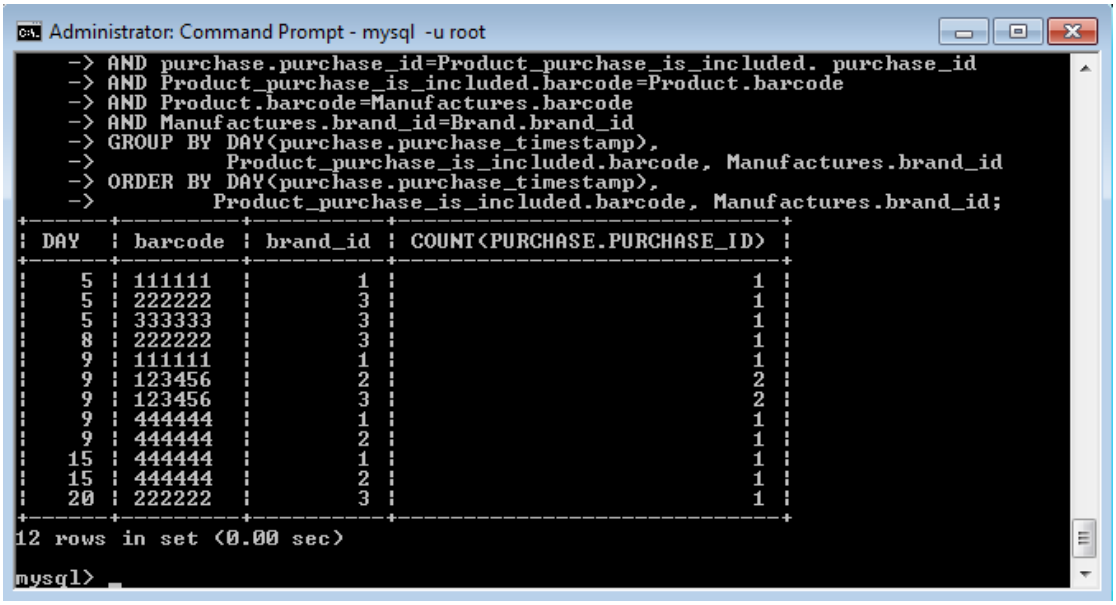
```
SELECT MONTH (purchase.purchase_timestamp) "MONTH",
       count(MONTH(purchase.purchase_timestamp))"#PRODUCTS",
       Product_purchase_is_included.barcode, Manufactures.brand_id
FROM  client, purchase, Product_purchase_is_included,  Product, Manufactures,
       Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
GROUP BY MONTH(purchase.purchase_timestamp),
         Product_purchase_is_included.barcode, Manufactures.brand_id
ORDER BY Product_purchase_is_included.barcode, Manufactures.brand_id;
```



Σύμφωνα με το παράδειγμά μας συμπεραίνουμε ότι κατά τον δεύτερο μήνα για παράδειγμα, πωλήθηκαν 2 προϊόντα με τον κωδικό 111111.

ΕΡΩΤΗΜΑ 4: *Ποιες μέρες ψώνισαν οι πελάτες;*
Για να βρούμε ποιες μέρες ψώνισαν οι πελάτες, θα πρέπει να δούμε το σύνολο των αγορών κάθε προϊόντος ανά ημέρα.

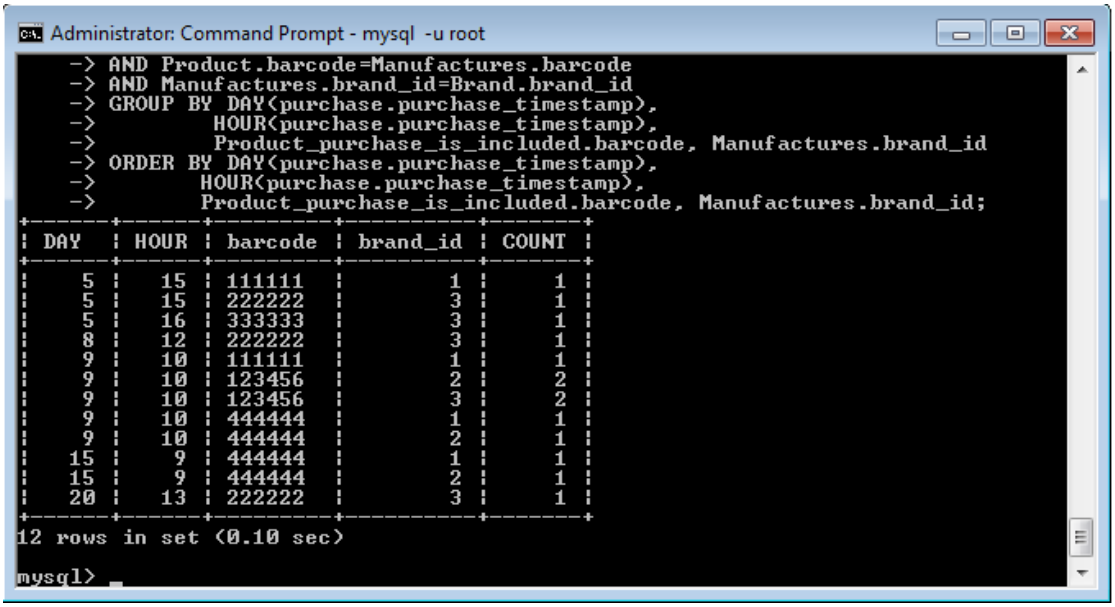
```
SELECT DAY(purchase.purchase_timestamp) DAY,
        Product_purchase_is_included.barcode,
       Manufactures.brand_id,
COUNT(PURCHASE.PURCHASE_ID)
FROM  client, purchase, Product_purchase_is_included,  Product, Manufactures,
       Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
GROUP BY DAY(purchase.purchase_timestamp),
       Product_purchase_is_included.barcode, Manufactures.brand_id
ORDER BY DAY(purchase.purchase_timestamp),
      Product_purchase_is_included.barcode, Manufactures.brand_id;
```



Στον πίνακα βλέπουμε ότι την 5η μέρα για παράδειγμα πωλήθηκε 1 τεμάχιο κάθε προϊόντος με κωδικό 111111,222222,333333.

ΕΡΩΤΗΜΑ 5: *Ποια ώρα της ημέρας πωλήθηκαν προϊόντα;*
Για να βρούμε ποια ώρα της ημέρας, θα πρέπει να βρούμε το σύνολο των αγορών κάθε προϊόντος ανά ώρα και μέρα.

```
SELECT DAY(purchase.purchase_timestamp)"DAY",
      HOUR(purchase.purchase_timestamp)"HOUR",
      Product_purchase_is_included.barcode, Manufactures.brand_id,
COUNT(PURCHASE.PURCHASE_ID)"COUNT"
FROM  client, purchase, Product_purchase_is_included,  Product, Manufactures,
      Brand
WHERE client.idclient= purchase.idclient
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
AND Product_purchase_is_included.barcode=Product.barcode
AND Product.barcode=Manufactures.barcode
AND Manufactures.brand_id=Brand.brand_id
GROUP BY DAY(purchase.purchase_timestamp),
      HOUR(purchase.purchase_timestamp),
      Product_purchase_is_included.barcode, Manufactures.brand_id
ORDER BY DAY(purchase.purchase_timestamp),
     HOUR(purchase.purchase_timestamp),
     Product_purchase_is_included.barcode, Manufactures.brand_id;
```

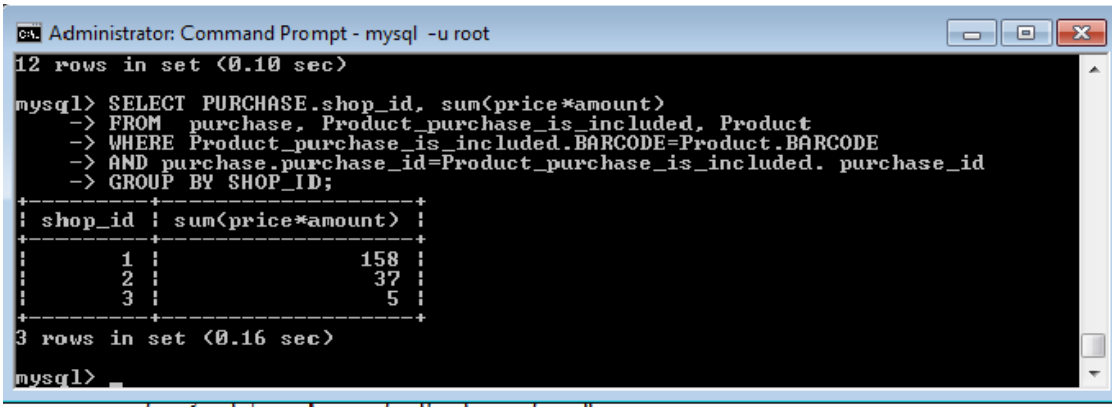


Από το παράδειγμά μας βλέπουμε ότι την 5η ημέρα στις 03:00μμ πωλήθηκε 1 τεμάχιο με κωδικό(barcode)111111 και 1 τεμάχιο με κωδικό(barcode)222222!

ΕΡΩΤΗΜΑ 6: *Ποιο κατάστημα είναι πρώτο σε πωλήσεις;*

- Για να το βρούμε ποιο κατάστημα είναι πρώτο σε πωλήσεις, θα πρέπει πρώτα να βρούμε τις πωλήσεις από κάθε κατάστημα.

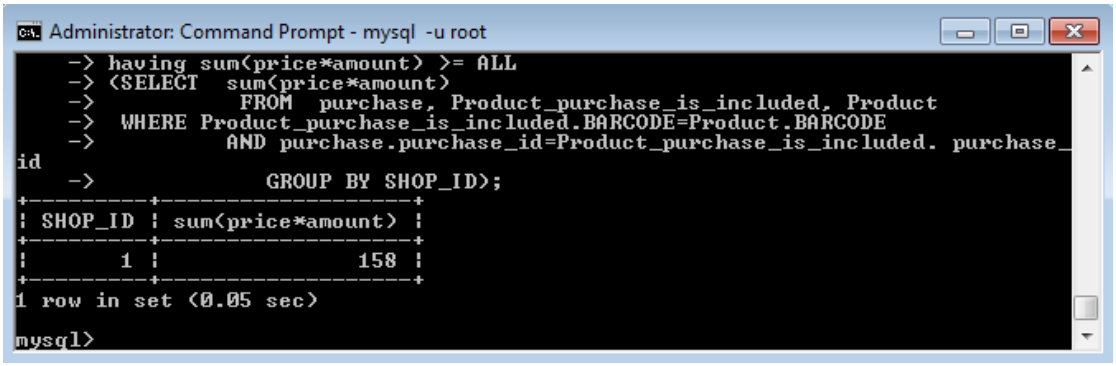
```
SELECT PURCHASE.shop_id, sum(price*amount)
FROM  purchase, Product_purchase_is_included, Product
WHERE Product_purchase_is_included.BARCODE=Product.BARCODE
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
GROUP BY SHOP_ID;
```



Βλέπουμε τις πωλήσεις του κάθε καταστήματος και συμπεραίνουμε ότι το πρώτο κατάστημα με κωδικό 1, έχει τις μεγαλύτερες πωλήσεις.

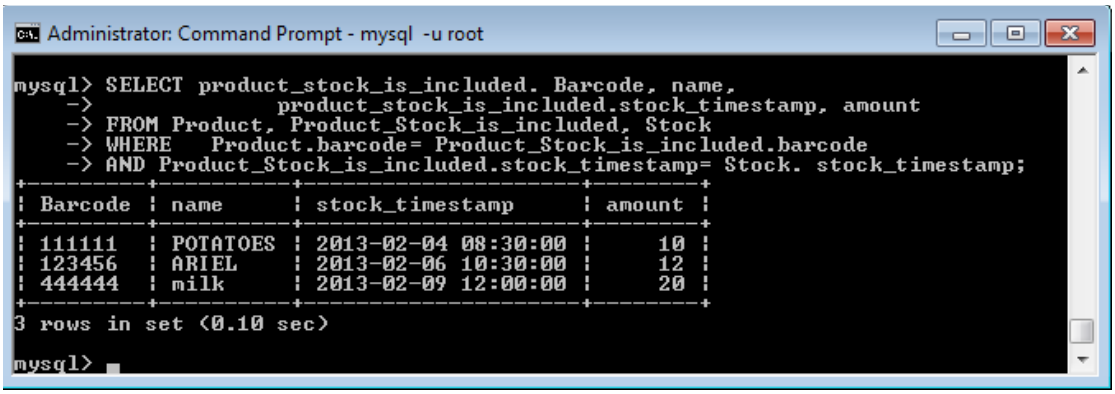
➢ Επόμενο βήμα είναι να εμφανίσουμε μόνο το κατάστημα που είναι πρώτο σε πωλήσεις .

```
SELECT  SHOP_ID, sum(price*amount)
FROM  purchase, Product_purchase_is_included, Product
WHERE Product_purchase_is_included.BARCODE=Product.BARCODE
AND purchase.purchase_id=Product_purchase_is_included. purchase_id
GROUP BY SHOP_ID
having sum(price*amount) >= ALL
        (SELECT  sum(price*amount)
         FROM  purchase, Product_purchase_is_included, Product
         WHERE Product_purchase_is_included.BARCODE=Product.BARCODE
        AND purchase.purchase_id=Product_purchase_is_included. purchase_id
        GROUP BY SHOP_ID);
```



ΕΡΩΤΗΜΑ 7: *Ποια είναι τα αποθέματα των προϊόντων κάθε στιγμή;*

```
SELECT product_stock_is_included. Barcode, name,
        product_stock_is_included.stock_timestamp, amount
FROM Product, Product_Stock_is_included, Stock
WHERE   Product.barcode= Product_Stock_is_included.barcode
AND Product_Stock_is_included.stock_timestamp= Stock. stock_timestamp;
```

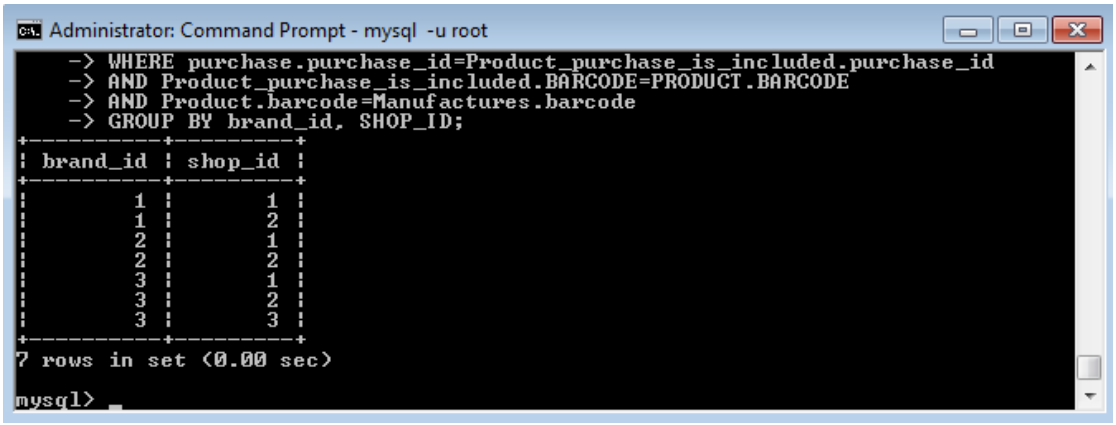


Τα αποθέματα των πατατών με κωδικό(barcode) 111111 είναι 10 την 04/02/2013 και ώρα 08:30 πμ.

ΕΡΩΤΗΜΑ 8: Ποιες είναι οι μάρκες των προϊόντων που έχουν πωληθεί σε όλα τα κατάστημα;

- Για να βρούμε τις μάρκες των προϊόντων που έχουν πωληθεί σε κάθε κατάστημα θα πρέπει να πρώτα να δούμε αρχικά ποια προϊόντα πουλά κάθε κατάστημα.

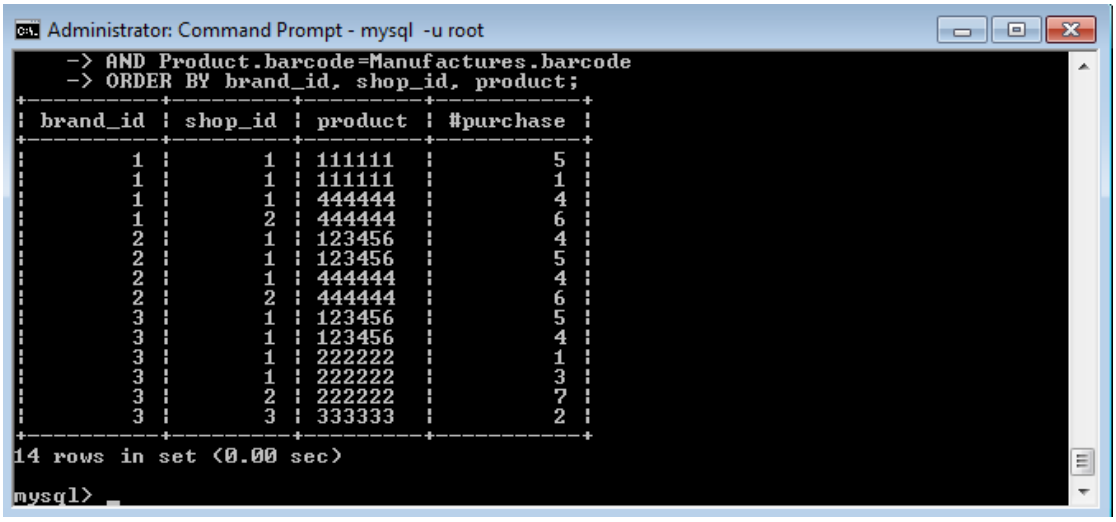
```
SELECT brand_id, shop_id
FROM  purchase, Product_purchase_is_included, Product, Manufactures
WHERE purchase.purchase_id=Product_purchase_is_included.purchase_id
AND Product_purchase_is_included.BARCODE=PRODUCT.BARCODE
AND Product.barcode=Manufactures.barcode
GROUP BY brand_id, SHOP_ID;
```



Βλέπουμε ότι το κατάστημα με κωδικό 1 πουλά προϊόντα με κωδικό 1,2 και 3.

- Αφού δούμε τι προϊόντα πουλάει το κάθε κατάστημα, επόμενο βήμα είναι να δούμε ποια προϊόντα έχουν πωληθεί και σε ποιο κατάστημα.

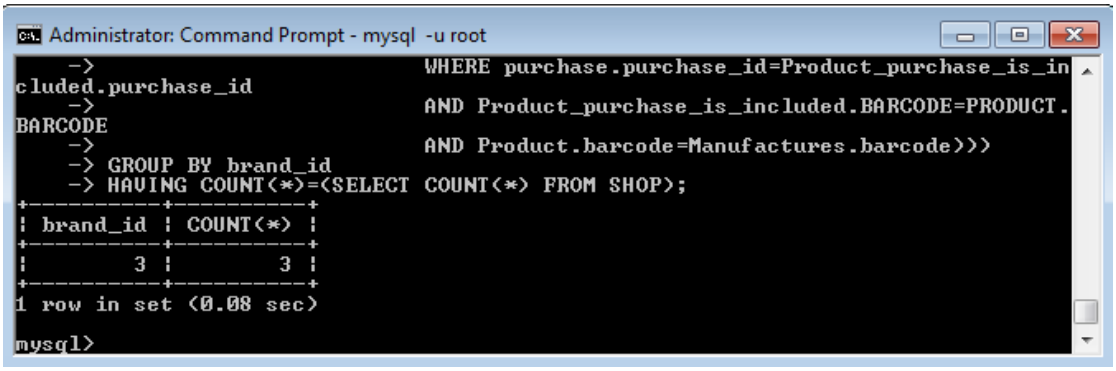
```
SELECT brand_id, shop_id, Product_purchase_is_included.BARCODE "product",
    Product_purchase_is_included.purchase_id "#purchase"
FROM  purchase, Product_purchase_is_included, Product, Manufactures
WHERE purchase.purchase_id=Product_purchase_is_included.purchase_id
AND Product_purchase_is_included.BARCODE=PRODUCT.BARCODE
AND Product.barcode=Manufactures.barcode
ORDER BY brand_id, shop_id, product;
```



Βλέπουμε ότι το κατάστημα με κωδικό 2 έχει πωλήσει 3 προϊόντα:
Το προϊόν με κωδικό(barcode) 444444, κωδικό μάρκας 1 και κωδικό αγοράς 6.
Το προϊόν με κωδικό (barcode) 444444, κωδικό μάρκας 2 και κωδικό αγοράς 6.
Το προϊόν με κωδικό(barcode)222222, κωδικό μάρκας 3 και κωδικό αγοράς 7.

➢ Εφόσον ολοκληρώσουμε το προηγούμενο ερώτημα, σειρά έχει να το εμπλουτίσουμε ώστε να πάρουμε τα αποτελέσματα που μας ενδιαφέρουν, δηλαδή ποιο είναι το προϊόν που πωλήθηκε σε όλα τα καταστήματα.

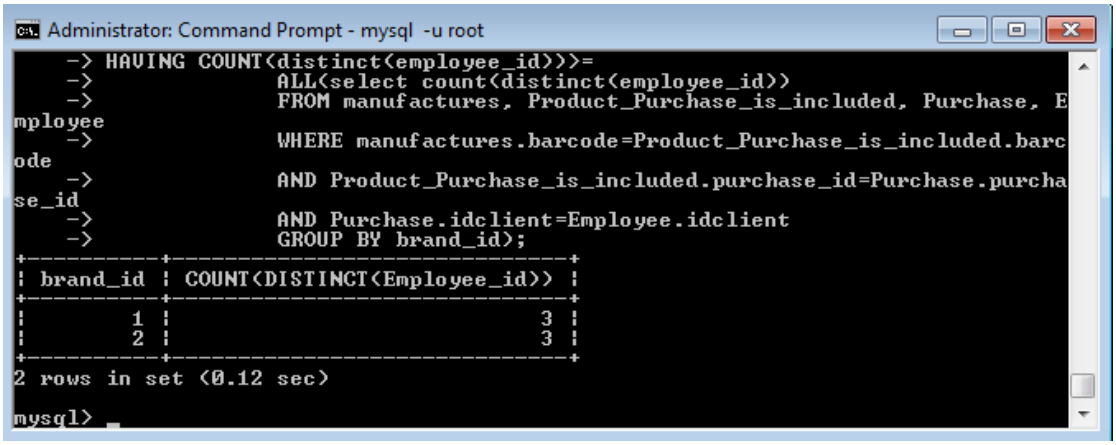
```
SELECT brand_id, COUNT(*)
FROM manufactures
WHERE BARCODE IN
    (SELECT PRODUCT_PURCHASE_IS_INCLUDED.BARCODE
     FROM manufactures,PRODUCT_PURCHASE_IS_INCLUDED
     WHERE
            manufactures.barcode=PRODUCT_PURCHASE_IS_INCLUDED.barcod
            e
      AND purchase_id IN
          (SELECT purchase_id
           FROM purchase
           WHERE shop_id in
                (SELECT  shop_id
                FROM  purchase, Product_purchase_is_included, Product,
Manufactures
                WHERE
purchase.purchase_id=Product_purchase_is_included.purchase_id
                AND
Product_purchase_is_included.BARCODE=PRODUCT.BARCODE
                AND Product.barcode=Manufactures.barcode)))
GROUP BY brand_id
HAVING COUNT(*)=(SELECT COUNT(*) FROM SHOP);
```



Άρα το προϊόν με κωδικό μάρκας 3έχει πωληθεί σε όλα τα καταστήματα.

ΕΡΩΤΗΜΑ 9: *Ποια είναι η μάρκα των προϊόντων την οποία έχουν αγοράσει όλοι οι υπάλληλοι των καταστημάτων;*

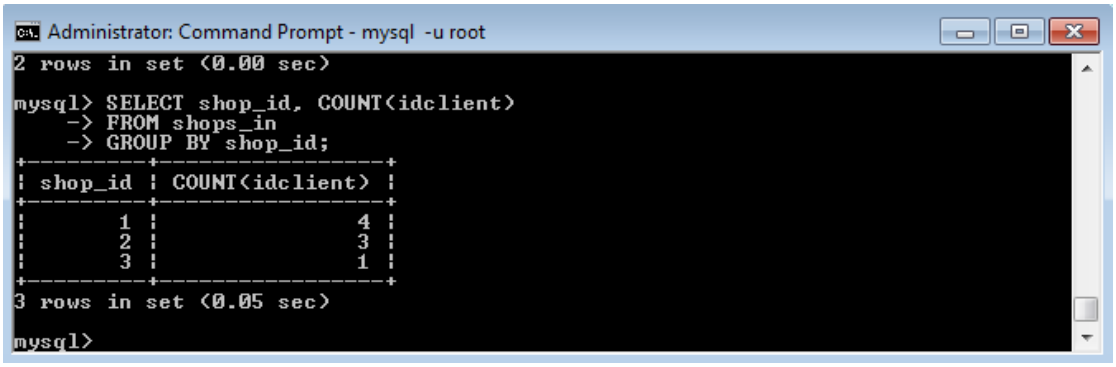
```
SELECT brand_id, COUNT(DISTINCT(Employee_id))
FROM manufactures, Product_Purchase_is_included, Purchase, Employee
WHERE manufactures.barcode=Product_Purchase_is_included.barcode
AND Product_Purchase_is_included.purchase_id=Purchase.purchase_id
AND Purchase.idclient=Employee.idclient
GROUP BY brand_id
HAVING COUNT(distinct(employee_id))>=
        ALL(select count(distinct(employee_id))
        FROM manufactures, Product_Purchase_is_included, Purchase, Employee
        WHERE manufactures.barcode=Product_Purchase_is_included.barcode
        AND Product_Purchase_is_included.purchase_id=Purchase.purchase_id
        AND Purchase.idclient=Employee.idclient
        GROUP BY brand_id);
```



Από τον πίνακα βλέπουμε ότι 3 υπάλληλοι έχουν προτιμήσει τη μάρκα με κωδικό 1 και 3 υπάλληλοι έχουν προτιμήσει τη μάρκα με κωδικό 2.

ΕΡΩΤΗΜΑ 10: *Πόσους πελάτες έχει συνολικά κάθε κατάστημα;*

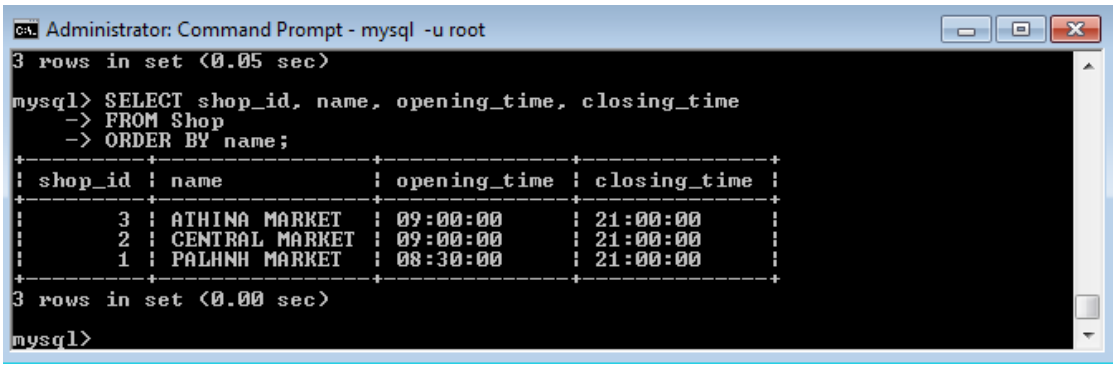
```
SELECT shop_id,
COUNT(idclient)
FROM shops_in
GROUP BY shop_id;
```



Σύμφωνα με το πίνακα βλέπουμε για παράδειγμα ότι το κατάστημα με κωδικό 1, έχει 4 πελάτες.

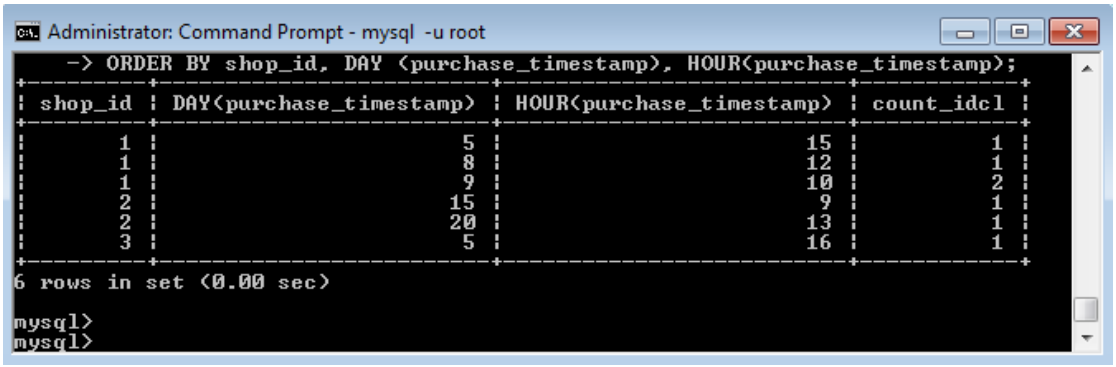
ΕΡΩΤΗΜΑ 11: *Ποιες είναι οι ώρες λειτουργίας κάθε καταστήματος;*

```
SELECT shop_id, name, opening_time, closing_time
FROM Shop
ORDER BY name;
```



ΕΡΩΤΗΜΑ 12: *Πόσοι πελάτες εξυπηρετούνται κάθε χρονική στιγμή και κάθε ημέρα της εβδομάδας;*
Θέλουμε να βρούμε δηλαδή ανά κατάστημα πόσοι πελάτες εξυπηρετήθηκαν σύμφωνα με την μέρα και την ώρα.

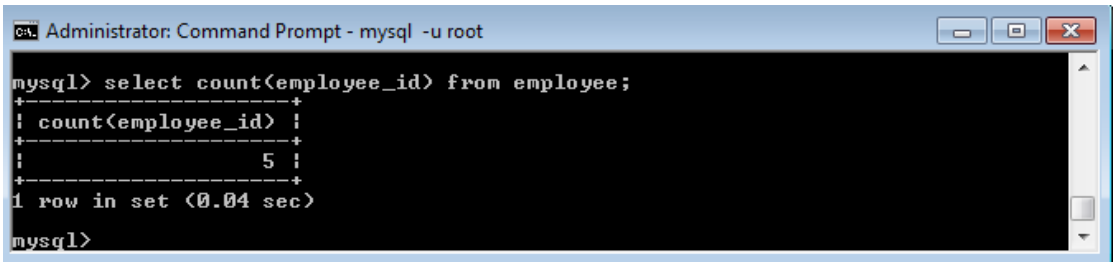
SELECT shop_id, DAY(purchase_timestamp), HOUR(purchase_timestamp),
COUNT(idclient) count_idcl
FROM Purchase
GROUP BY (purchase_timestamp), HOUR(purchase_timestamp)
ORDER BY shop_id, DAY (purchase_timestamp), HOUR(purchase_timestamp);



Και όντως βλέπουμε ότι το 3ο κατάστημα για παράδειγμα, την 5η ημέρα, στις 04:00 μμ εξυπηρέτησε 1 πελάτη συνολικά.

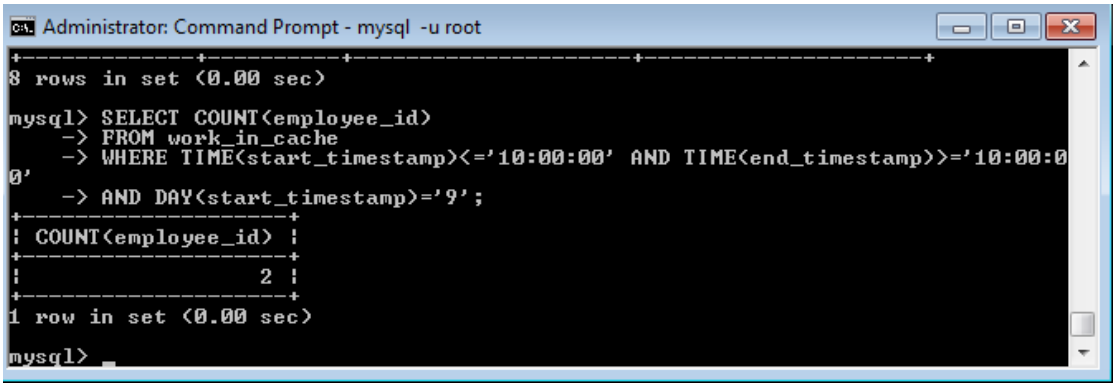
ΕΡΩΤΗΜΑ 13: *Πόσοι υπάλληλοι εργάζονται;*

SELECT COUNT(employee_id)
FROM employee;



ΕΡΩΤΗΜΑ 14: *Πόσοι ταμίες δουλεύουν συνολικά μια οποιαδήποτε μέρα;*
Στο ερώτημα που ακολουθεί, επιλέξαμε να δούμε πόσοι ταμίες δουλεύουν την 9η μέρα.

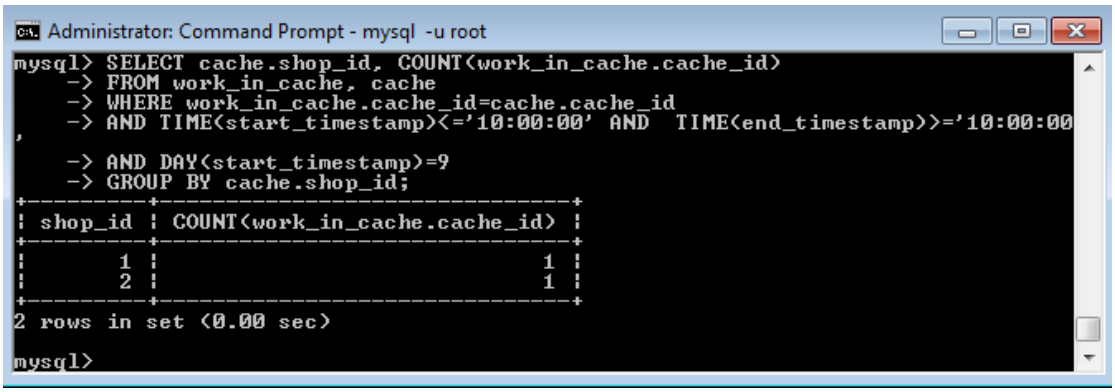
SELECT COUNT(employee_id)
FROM work_in_cache
WHERE TIME(start_timestamp)<='10:00:00' AND
TIME(end_timestamp)>='10:00:00'
AND DAY(start_timestamp)='9';



Άρα, την 9η ημέρα 2 υπάλληλοι ξεκίνησαν την δουλειά τους πριν τις 10:00πμ.

ΕΡΩΤΗΜΑ 15: *Πόσες ταμιακές μηχανές θέτονται σε λειτουργία οποιαδήποτε ώρα και μέρα σε κάθε κατάστημα;*
Στο ερώτημα που ακολουθεί, επιλέξαμε να δούμε πόσες ταμιακές μηχανές λειτουργούν την 9η μέρα.

```
SELECT cache.shop_id, COUNT(work_in_cache.cache_id)
FROM work_in_cache, cache
WHERE work_in_cache.cache_id=cache.cache_id
AND TIME(start_timestamp)<='10:00:00' AND  TIME(end_timestamp)>='10:00:00'
AND DAY(start_timestamp)=9
GROUP BY cache.shop_id;
```

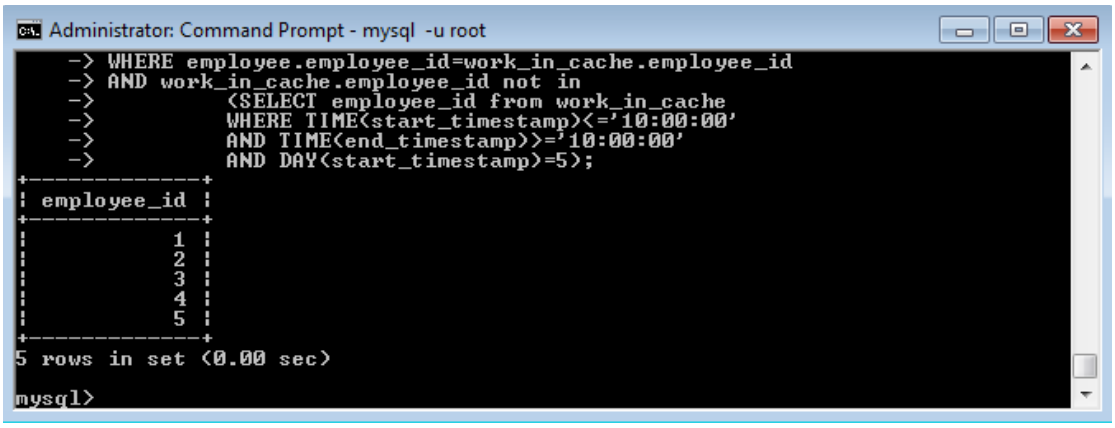


Από το παράδειγμά μας συμπεραίνουμε για παράδειγμα ότι την 9η μέρα το κατάστημα με κωδικό 1 λειτούργησε 1 ταμιακή μηχανή.

ΕΡΩΤΗΜΑ 16: *Πόσοι συνολικά υπάλληλοι δε δουλεύουν ως ταμίες οποιαδήποτε μέρα;*

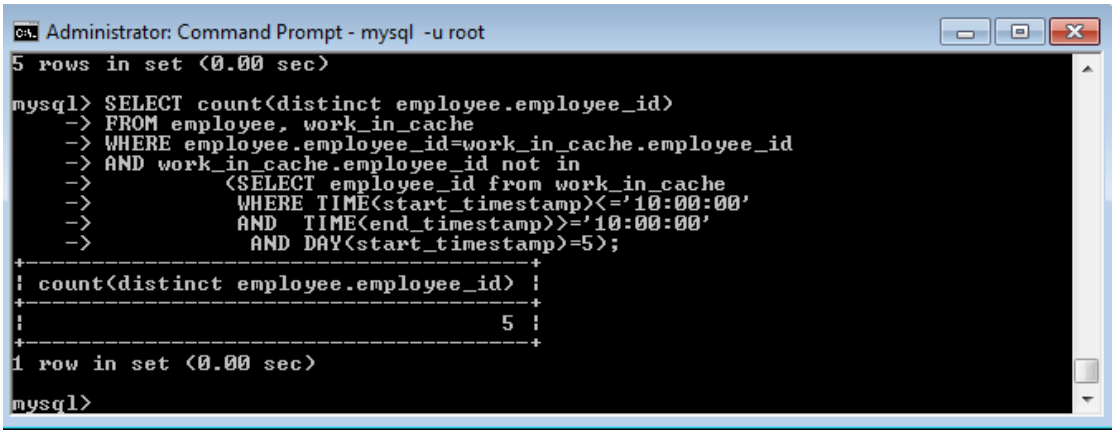
- Θέλουμε να βρούμε δηλαδή τους υπαλλήλους ταμίες συμφωνά με το κωδικό τους, οι οποίοι δε δουλεύουν την 5η μέρα.

```
SELECT Distinct employee.employee_id
FROM employee, work_in_cache
WHERE employee.employee_id=work_in_cache.employee_id
AND work_in_cache.employee_id not in
        (SELECT employee_id from work_in_cache
        WHERE TIME(start_timestamp)<='10:00:00'
        AND TIME(end_timestamp)>='10:00:00'
        AND DAY(start_timestamp)=5);
```



- Με βάση το προηγούμενο ερώτημα θα βρούμε συνολικά τους υπαλλήλους που δεν δούλευαν ως ταμίες.

```
SELECT count(distinct employee.employee_id)
FROM employee, work_in_cache
WHERE employee.employee_id=work_in_cache.employee_id
AND work_in_cache.employee_id not in
        (SELECT employee_id from work_in_cache
         WHERE TIME(start_timestamp)<='10:00:00'
         AND  TIME(end_timestamp)>='10:00:00'
          AND DAY(start_timestamp)=5);
```

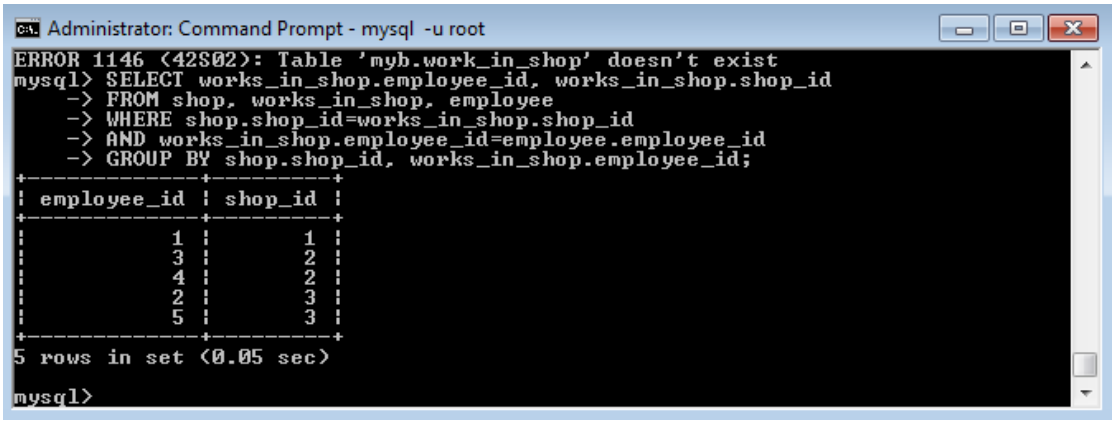


Συνολικά 5 υπάλληλοι δε δούλεψαν την 5η ημέρα!

ΕΡΩΤΗΜΑ 17: *Ποιοι είναι οι υπάλληλοι που έχουν δουλέψει σε περισσότερα από ένα καταστήματα;*

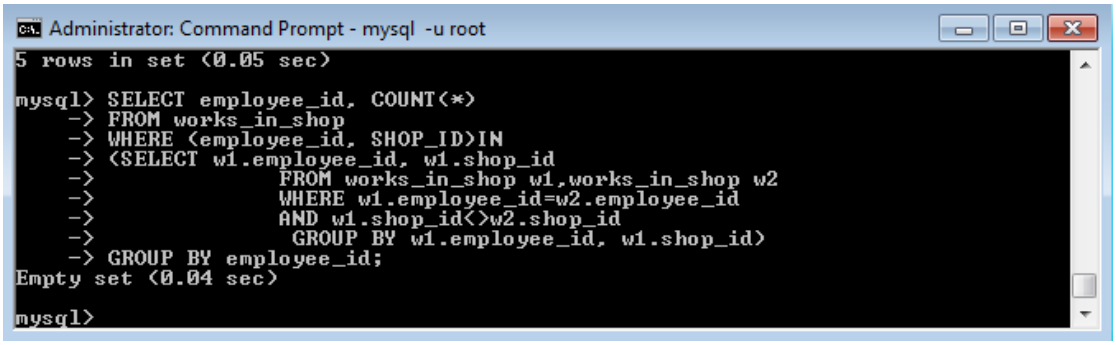
➢ Πρώτα θα πρέπει να δούμε σε ποια καταστήματα έχουν δουλέψει οι υπάλληλοι, αναλυτικά.

```
SELECT works_in_shop.employee_id, works_in_shop.shop_id
FROM shop, works_in_shop, employee
WHERE shop.shop_id=works_in_shop.shop_id
AND works_in_shop.employee_id=employee.employee_id
GROUP BY shop.shop_id, works_in_shop.employee_id;
```



➢ Οι υπάλληλοι που έχουν δουλέψει σε περισσότερα του ενός καταστήματα

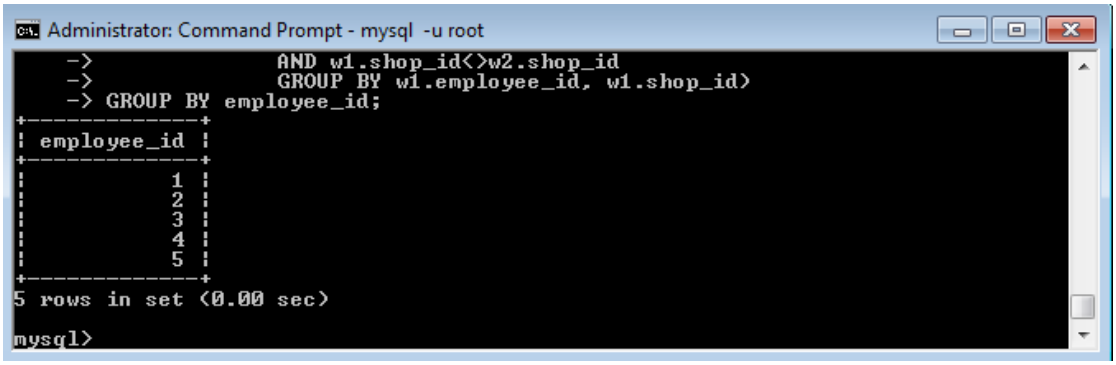
```
SELECT employee_id, COUNT(*)
FROM works_in_shop
WHERE (employee_id, SHOP_ID)IN
        (SELECT w1.employee_id, w1.shop_id
         FROM works_in_shop w1,works_in_shop w2
         WHERE w1.employee_id=w2.employee_id
         AND w1.shop_id<>w2.shop_id
          GROUP BY w1.employee_id, w1.shop_id)
GROUP BY employee_id;
```



Κανένας υπάλληλος δεν έχει δουλέψει σε περισσότερα του ενός καταστήματος.

ΕΡΩΤΗΜΑ 18: *Ποιοι είναι οι υπάλληλοι που έχουν δουλέψει μόνο σε ένα κατάστημα;*

```
SELECT employee_id
FROM works_in_shop
WHERE (employee_id, SHOP_ID)NOT IN
        (SELECT w1.employee_id, w1.shop_id
         FROM works_in_shop w1,works_in_shop w2
         WHERE w1.employee_id=w2.employee_id
         AND w1.shop_id<>w2.shop_id
         GROUP BY w1.employee_id, w1.shop_id)
GROUP BY employee_id;
```



Οι κωδικοί των υπαλλήλων που έχουν υπάρξει σε ένα και μόνο κατάστημα φαίνεται στο παραπάνω πίνακα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Σχεσιακές Βάσεις Δεδομένων, Σκουρλάς Χρήστος, έκδοση 2000
- Υλοποίηση εφαρμογών με γλώσσα SQL, Σκουρλάς Χρήστος, έκδοση 2001
- http://www.coopatlantic.ca/
- Για το Οργανόγραμμα πήρα πληροφορίες από:
  http://www.3bmarkets.gr/company-profile
  http://www.masoutis.gr/swift.jsp?CMCCode=2004&extLang=
  http://www.lisi.gr/index.php?id=54
  http://www.evroslead.gr
- Για τον τρόπο λειτουργίας πήρα στοιχεία από τις εξής εταιρίες :
  http://www.masoutis.gr/swift.jsp?CMCCode=2005&extLang=
  www.dmst.aueb.gr/damianos/SA03/ERD.ppt (Οικονομικό πανεπιστήμιο Αθηνών)
  http://www.anedik.com/online/index.php/2010-01-19-19-14-28.html
  http://www.qualitypath.gr/iso9001.html
- Εγκατάσταση: http://java.sun.com
- Εγκατάσταση Apache Tomcat : http://tomcat.apache.org/
- Εγκατάσταση MySQL : http://dev.mysql.com/get/Downloads
- Για την σχεσιακή άλγεβρα άντλησα πληροφορίες από :
  ftp://ftp.teiser.gr/pliroforiki/Baseis_Dedomenon_I/B-Theory-slides.pdf
  www.neural.uom.gr/Documents/DataBases/chapter5.pdf