Αρχεία και Βάσεις Δεδομένων

ΗΥ-360

Ομαδική Εργασία

Credit Card Company (CCC)

Αθανασάκη Βαγγελιώ (3076)
Αναγνώστου Λεωνίδας (2828)
Πεχυνάκη - Μαμουνάκη Κατερίνα (3103)

22/1/2017

Περιεχόμενα:

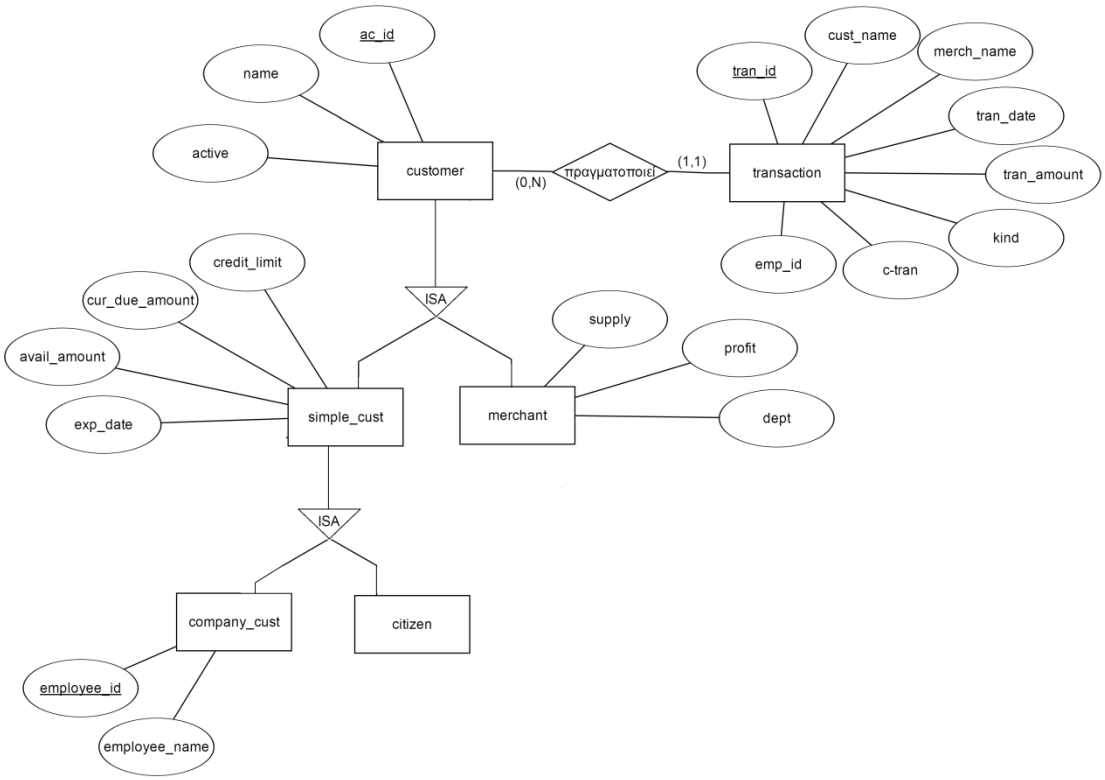
1 Περίληψη ομαδικής εργασίας

Στην εργασία που υλοποιήσαμε, σχεδιάσαμε και τεκμηριώσαμε ένα σύστημα, το οποίο θα μπορούσε να αυτοματοποιήσει ένα τμήμα των διεργασιών που λαμβάνουν χώρα σε μια εταιρεία (CCC: a Credit Card Company) που θα παρέχει μια πιστωτική κάρτα στους πελάτες της και θα τους δίνει μια λεπτομερή αναφορά, η οποία θα εκθέτει το σχεδιασμό αυτού του συστήματος.

Η συγκεκριμένη εταιρεία έχει τρεις τύπου πελατών: Merchants, οι οποίοι δέχονται την πιστωτική κάρτα σαν μορφή πληρωμής, Simple Customers, οι οποίοι μπορούν να χρησιμοποιούν την πιστωτική κάρτα για τις αγορές τους και Companies, που έχουν έναν εταιρικό λογαριασμό με την CCC, με την δυνατότητα περισσότερα από ένα άτομα να χρησιμοποιούν την ίδια κάρτα (αυτά τα άτομα, συγκεκριμένα θα είναι υπάλληλοι της εταιρείας και θα έχουν το όνομα στην βάση δεδομένων ως Employees). Υποθέσαμε ότι οι τρεις τύποι πελατών είναι διακεκριμένοι (δηλαδή ένας ιδιώτης δεν μπορεί να είναι συγχρόνως και έμπορος).

Η υλοποίηση του συστήματος βάσεων δεδομένων έγινε με την χρήση MySQL. Η δημιουργία των ιστοσελίδων με την χρήση HTML και CSS και τέλος η συνένωση των προαναφερθέντων προγραμμάτων και η διαχείριση των βάσεων δεδομένων μέσω της χρήσης PHP.

2 Πλήρες διάγραμμα οντοτήτων-σχέσεων για την εταιρεία



3 Γνωρίσματα (όνομα, τύπος) όλων των οντοτήτων και σχέσεων

Αρχικά έχουμε την οντότητα **customer** που έχει τα γνωρίσματα:

- ac_id (unsigned small integer συνολικού μεγέθους αριθμών 5), το οποίο αντιπροσωπεύει τον αριθμό λογαριασμού του customer (πιο συγκεκριμένα του merchant, της company ή του απλού citizen) και είναι κλειδί αυτής της οντότητας customer, άρα κληρονομείται και στις υπο-οντότητες της (simple_cust, merchant, company_cust, citizen)
- name (var character , δηλαδή μεταβλητού μεγέθους χαρακτήρες που μπορούν να φτάσουν περιπου μέχρι 25 ψηφια συνολικό μέγεθος), αντιπροσωπεύει το όνομα του customer (πιο συγκεκριμένα του merchant, της company ή του απλού citizen).
- active (*) (enum τιμών true και false, κάνει την λειτουργία ενός boolean), παραθέτει το status του account, το οποίο είναι ουσιαστικά active και inactive.

Επίσης, έχουμε την οντότητα **simple_cust** που έχει τα γνωρίσματα:

- ac_id, που το κληρονομεί από την οντότητα customer (για περισσότερες λεπτομέρειες στην οντότητα customer)
- credit_limit (float integer), το πιστωτικό όριο του πελάτη

- cur_due_amount (float integer), ουσιαστικά είναι το current due amount, δηλαδή το τρέχον ποσό που οφείλει
- avail_amount (float integer), ουσιαστικά είναι το available amount, δηλαδή το διαθέσιμο υπόλοιπο πίστωσης
- exp_date (date), ουσιαστικά το expire date, δηλαδή η ημερομηνία λήξης του λογαριασμού

Επιπρόσθετα, υπάρχει η οντότητα **company_cust**, η οποία είναι υπό-οντότητα του simple_cust και κατά συνέπεια της οντότητας customer, που έχει τα γνωρίσματα:

- ac_id, που το κληρονομεί από την οντότητα simple_cust (για περισσότερες λεπτομέρειες στην οντότητα customer), ουσιαστικά αντιπροσωπεύει τον αριθμό λογαριασμού της εταιρείας
- employee_id (charater αριθμού 6, σε αυτήν την περίπτωση επειδή δεν έχουμε var character, πρέπει και ζητάμε το employee_id να είναι ακριβώς 6 χαρακτήρες), το οποίο είναι ο αριθμός λογαριασμού του υπαλλήλου της εταιρείας και είναι μοναδικός
- employee_name(var character, δηλαδή μεταβλητού μεγέθους χαρακτήρες που μπορούν να φτάσουν περίπου μέχρι 25 συνολικό μέγεθος), αντιπροσωπεύει το όνομα του υπαλλήλου της εταιρείας.

Επίσης, υπάρχει η οντότητα **citizen,** που έχει το γνώρισμα:

- ac_id, που το κληρονομεί από την οντότητα simple_cust (για περισσότερες λεπτομέρειες στην οντότητα customer)

Η οντότητα citizen υπάρχει στο E-R για να δείξουμε το διαχωρισμό του customer σε ιδιώτες και εταιρίας. Τα γνωρίσματά του τα κληρονομεί από τον customer και από τον simple_cust , αλλά δεν χει δικα της ξεχωριστά γνωρίσματα αλλά ταυτίζονται με αυτά της οντότητας simple_cust .

Ακόμα, έχουμε την οντότητα **merchant** που έχει τα γνωρίσματα:

- ac_id, που το κληρονομεί από την οντότητα customer (για περισσότερες λεπτομέρειες στην οντότητα customer)
- supply (float integer), αντιπροσωπεύει την προμήθεια που λαμβάνει η CCC από τον έμπορο
- profit (float integer), είναι το συνολικό κέρδος για τις αγορές από πελάτες στην επιχείρηση
- debt (float integer) , το ποσό που οφείλει ο έμπορος στην CCC.

Τέλος, η οντότητα **transaction** που έχει τα γνωρίσματα:

- tran_id (*) (unsigned integer συνολικού μεγέθους 11 αριθμών), το οποίο αντιπροσωπεύει το αριθμό της δοσοληψίας και αποτελεί και το κλειδί της οντότητας transaction

- cust_name (var character, δηλαδή μεταβλητού μεγέθους χαρακτήρες που μπορούν να φτάσουν περιπου μέχρι 25 συνολικό μέγεθος), αντιπροσωπεύει το όνομα του πελάτη που πραγματοποιεί μια αγορά στην επιχείρηση ενός εμπόρου
- merch_name (var character, δηλαδή μεταβλητού μεγέθους χαρακτήρες που μπορούν να φτάσουν περιπου μέχρι 25 συνολικό μέγεθος), αντιπροσωπεύει το όνομα του εμπόρου που στην επιχείρηση του πραγματοποιείται η αγορά
- tran_date (timestamp), στην περίπτωση του γνωρίσματος tran_date γίνεται χρήση timestamp μεταβλητής για να έχουμε μια καλύτερη, πιο λεπτομερή και πιο ευκολό-διαχειριζόμενη ημερομηνία για τις δοσοληψίες
- tran_amount (double integer), αντιπροσωπεύει το χρηματικό ποσό της δοσοληψίας
- kind(enum τιμών credit και charge), μας δίνει τον τύπο της δοσοληψίας, δηλαδή αν είναι με πίστωση ή με χρέωση
- c_tran (*) (enum τιμών true και false, κάνει την λειτουργία ενός boolean), παραθέτει το αν έχουμε δοσοληψία εταιρείας ή όχι
- emp_id (*) (charater αριθμού 6), το οποίο είναι ο αριθμός λογαριασμού του υπαλλήλου της εταιρείας.

4 Πρωτεύοντα κλειδιά

Τα πρωτεύοντα κλειδιά μας είναι το ac_id για την οντότητα customer, το tran_id για την οντότητα transaction και το employee_id για την company_cust. Το ac_id είναι unsigned small integer συνολικού μεγέθους αριθμών 5, το οποίο αντιπροσωπεύει τον αριθμό λογαριασμού του customer (πιο συγκεκριμένα του merchant, της company ή του απλού citizen) και επειδή είναι κλειδί της οντότητας customer κληρονομείται και στις υπο-οντότητες της (simple_cust, merchant, company_cust, citizen). Το tran_id είναι unsigned integer συνολικού μεγέθους 11 αριθμών, το οποίο αντιπροσωπεύει το αριθμό της δοσοληψίας. Τέλος έχουμε το employee_id το οποίο είναι, σταθερου μεγέθους χαρακτήρες 6 στον αριθμό, αντιπροσωπεύει το μοναδικο id του υπαλλήλου πουεχει στην εταιρεία.

5 Επεξηγήσεις για τα μη-προφανή γνωρίσματα και τις μη-προφανείς σχέσεις

Τα γνωρίσματα, τα οποία περιέχουν αυτό το σύμβολο(*), αποτελούν γνωρίσματα που προσθέσαμε εμείς για την καλύτερη διαχείριση της βάσης δεδομένων μας και την πιο αποτελεσματική λειτουργία της εργασίας μας, ως προς τα ζητούμενα που είχαμε να υλοποιήσουμε. Αρχικά το active του customer είναι μια enum μεταβλητή με την οποία κρατάμε εάν είναι active ή inactive, ο λογαριασμός ενός πελάτη. Η κύρια χρήση αυτής της μεταβλητής είναι για την «διαγραφή» - κλείσιμο λογαριασμού ενός πελάτη. Επίσης, η tran_id του transaction είναι μια βοηθητική μεταβλητή για να αποκτήσουμε μοναδικότητα για την κάθε μια δοσοληψία που θα γίνεται. Αυτό βοηθάει στην καλύτερη αναζήτηση και διαχείριση της βάσης δεδομένων μας. Ακόμα, έχουμε την c_tran, η οποία είναι και αυτή «enum» και τσεκάρει αν έχει κάνει την αγορά μια εταιρεία ή όχι. Και τέλος, την emp_id, η οποία μας παραθέτει το id του υπαλλήλου της εταιρείας (με όνομα cust_name), εφόσον έχει γίνει όμως αγορά από μια εταιρεία (δηλαδή το c_tran είναι true). Αυτά τα δύο μας

βοηθάνε να έχουμε έναν έλεγχο από την CCC, αλλά και από την ίδια την εταιρεία που έχει λογαριασμό στην CCC.

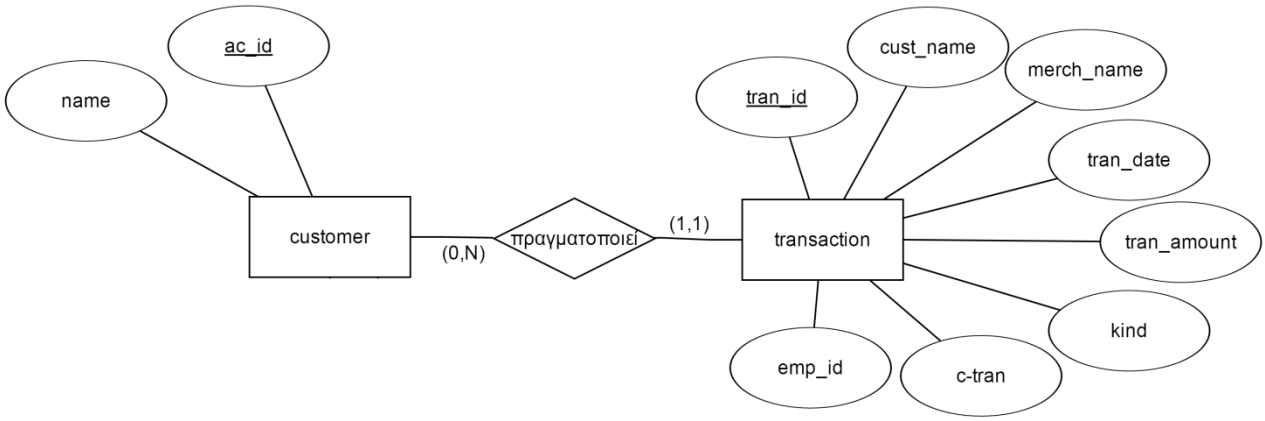
6 Περιορισμοί πληθικότητας

Οι περιορισμοί πληθικήτητας περιορίζουν το σύνολο των αποδεκτών σχέσεων μεταξύ συνόλων οντοτήτων. Παίρνοντας τον ορισμό του μοντέλου οντοτήτων- σχέσεων, έχουμε ότι «'Εστω E,F οντότητες οι οποίες συμμετέχουν σε μια σχέση R.

- αν max-card(E,R) = 1 , τότε η E έχει μονότιμη συμμετοχή στην R
- αν max-card(E,R) = 1 , τότε η E έχει πλειότιμη συμμετοχή στην R
- μια δυαδική σχέση R μεταξύ των οντοτήτων E, F είναι σχέση «πολλά- προς- πολλα» (many- to- many ή N-N) αν και η E και η F έχουν πλειότιμη συμμετοχή στην R
- αν και η E και η F έχουν μονότιμη συμμετοχή, η R είναι σχέση 1-1 (one- to- one)
- αν η E έχει μονότιμη συμμετοχή και η F έχει πλειότιμη συμμετοχή η R είναι σχέση 1-N (one- to- many).

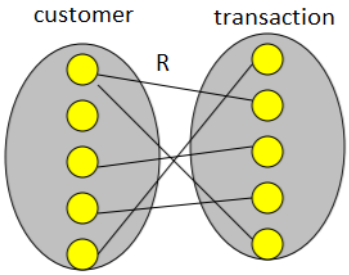
Ακόμα από έναν 2^{ο} ορισμό έχουμε ότι αν μια οντότητα E, που συμμετέχει σε μια σχέση R έχει min-card(E,R) = 1 , τότε η E έχει υποχρεωτική (mandatory) συμμετοχή στην R, ενώ αν min-card(E,R) = 0 , τότε έχει προαιρετική (optional) συμμετοχή στην R.

Στην δική μας περίπτωση, έχουμε μόνο την εξής σχέση:



Και χωρίς τα γνωρίσματα, έχουμε τις εξής οντότητες με την μεταξύ τους σχέση:

Παρατηρούμε δηλαδή ότι έχουμε τον εξής περιορισμό πληθικότητας:



min-card(customer,R) = 0
max-card(customer,R) = N
min-card(transaction,R) = 1
max-card(transaction,R) = 1

7 Μετάφραση μοντέλου σε σχεσιακό μοντέλο

Το E-R διάγραμμα μας μεταφράζεται σε σχεσιακό μοντέλο ως εξής:

- customer:

ac_id	name

- simple_cust:

ac_id	credit_limit	cur_due_amount	avail_amount	exp_date	active

- merchant:

ac_id	supply	profit	debt

- company_cust:

ac_id	employee_id	employee_name

- transaction:

tran_id	cust_name	merch_name	tran_date	tran_amount	kind	c_tran	…
…	emp_id						

8 Εντολές της γλώσσας ορισμού δεδομένων για τις σχέσεις που προκύπτουν

Για τη δημιουργία του πίνακα **company_cust** κάναμε:

```
CREATE TABLE `company_cust` (

 `ac_id` smallint(5) UNSIGNED NOT NULL,

 `employee_id` char(6) NOT NULL,
```

```
  `employee_name` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Αλλά αυτή ήταν η αρχική μας database. Στην συνέχεια άλλαξε και έγινε:

```
ALTER TABLE `company_cust`
  ADD PRIMARY KEY (`ac_id`,`employee_id`);
ALTER TABLE `company_cust`
  ADD CONSTRAINT `company_cust_ibfk_1` FOREIGN KEY (`ac_id`) REFERENCES `simple_cust` (`ac_id`);
```

Για τη δημιουργία του πίνακα **customer** κάναμε:

```
CREATE TABLE `customer` (
  `ac_id` smallint(5) UNSIGNED NOT NULL,
  `name` varchar(25) NOT NULL,
  `active` enum('true','false') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Και στην συνέχεια τον αλλάξαμε ως εξής:

```
ALTER TABLE `customer`
  ADD PRIMARY KEY (`ac_id`) USING BTREE,
  ADD UNIQUE KEY `name` (`name`);
```

Για τη δημιουργία του πίνακα **merchant** κάναμε:

```
CREATE TABLE `merchant` (
  `ac_id` smallint(5) UNSIGNED NOT NULL,
  `supply` float NOT NULL,
  `profit` float NOT NULL,
  `debt` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Και στην συνέχεια τον αλλάξαμε ως εξής:

```
ALTER TABLE `merchant`
  ADD PRIMARY KEY (`ac_id`);
ALTER TABLE `merchant`
  ADD CONSTRAINT `merchant_ibfk_1` FOREIGN KEY (`ac_id`) REFERENCES `customer` (`ac_id`);
```

Για τη δημιουργία του πίνακα **simple_cust** κάναμε:

```
CREATE TABLE `simple_cust` (
  `ac_id` smallint(5) UNSIGNED NOT NULL,
  `credit_limit` float NOT NULL,
  `cur_due_amount` double NOT NULL,
  `avail_amount` float NOT NULL,
  `exp_date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Και στην συνέχεια τον αλλάξαμε ως εξής:

```
ALTER TABLE `simple_cust`
  ADD PRIMARY KEY (`ac_id`);
ALTER TABLE `simple_cust`
  ADD CONSTRAINT `simple_cust_ibfk_1` FOREIGN KEY (`ac_id`) REFERENCES `customer` (`ac_id`);
```

Για τη δημιουργία του πίνακα **transaction** κάναμε:

```
CREATE TABLE `transaction` (
  `tran_id` int(11) UNSIGNED NOT NULL,
  `cust_name` varchar(25) NOT NULL,
  `merch_name` varchar(25) NOT NULL,
  `tran_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `tran_amount` double NOT NULL,
  `kind` enum('credit','charge') NOT NULL,
  `c_tran` enum('true','false') NOT NULL,
  `emp_id` char(6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Και στην συνέχεια τον αλλάξαμε ως εξής:

```
ALTER TABLE `transaction`
  ADD PRIMARY KEY (`tran_id`,`cust_name`,`merch_name`);
ALTER TABLE `transaction`
  MODIFY `tran_id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=67;
```

9 Περιορισμοί ακεραιότητας και συναρτησιακές εξαρτήσεις

Οι περιορισμοί ακεραιότητας αποτελούν μηχανισμό για τον έλεγχο της συνέπειας των δεδομένων. Χρησιμοποιούνται για να εξασφαλιστεί ότι μια βάση δεδομένων δεν θα βρεθεί σε ασυνεπή κατάσταση. Οι περιορισμοί ακεραιότητας στο μοντέλο Οντοτήτων – Σχέσεων ορίζεται με τα εξής: ορισμό κλειδιών, δήλωση ενός πρωτεύοντος (ή υποψηφίου κλειδιού) για ένα σύνολο οντοτήτων περιορίζει τις αποδεκτές εισαγωγές και ενημερώσεις σε αυτές ώστε να μην δημιουργούνται οντότητες με ίδια τιμή στο κλειδί και περιορισμοί πληθικότητα, περιορίζουν το σύνολο των αποδεκτών σχέσεων μεταξύ συνόλων οντοτήτων. Εν γένει, οι περιορισμοί ακεραιότητας μπορούν να είναι αυθαίρετα κατηγορήματα που αναφέρονται σε μια βάση δεδομένων.

Έχουμε τους εξής τύπους περιορισμών ακεραιοτήτων:

1. Οντοτήτων:
   Αφορά τα πρωτεύοντα κλειδιά τα οποία απαγορεύονται να είναι null. Αυτό ισχύει λόγω των περιορισμών και των ελέγχων που τέθηκαν κατά την υλοποίηση του project.
2. Σημασιολογικής ακεραιότητας:
   Αφορά τους περιορισμούς που υπάρχουν στον λογικό περιβάλλον- κόσμο.
   - ac_id: δεν μπορεί να είναι αρνητικός αριθμός
   - name: πρέπει να αποτελείται από ονόματα του φυσικού κόσμου
   - exp_date: πρέπει να ακολουθεί τις λογικές συμβάσεις (π.χ. όχι 31/2/2025)
   - credit_limit:δεν έχει νόημα να βάλουμε αρνητικούς αριθμούς
   - avail_amount:δεν έχει νόημα να βάλουμε αρνητικούς αριθμούς
   - profit: δεν μπορεί να είναι αρνητικός αριθμός
   - supply: δεν μπορεί να είναι αρνητικός αριθμός
   - employee_id: ακολουθεί λογικούς κανόνες και απαρτίζεται από 6 χαρακτήρες (π.χ. αριθμός ταυτότητας)
   - tran_id: δεν έχει νόημα να είναι αρνητικός αριθμός
   - tran_date: πρέπει να ακολουθεί τις λογικές συμβάσεις (π.χ. όχι 31/2/2025)
   - tran_amount: δεν έχει νόημα να είναι αρνητικός αριθμός
   - kind: μπορεί να πάρει μόνο τις τιμές credit(πίστωση) και charge(χρέωση)
3. Αναφορικής ακεραιότητας:
   Όταν μια πλειάδα αναφέρεται σε μία άλλη τότε και η δεύτερη πρέπει να υπάρχει. Πράγμα που ισχύει λόγω των περιορισμών και των ελέγχων που τέθηκαν κατά την υλοποίηση του project.
4. Πεδίο Ορισμού:
   Η τιμή κάθε γνωρίσματος Α πρέπει να είναι μια ατομική τιμή στο πεδίο ορισμού αυτού του γνωρίσματος. Πιο συγκεκριμένα για το project μας έχουν αναλυθεί στην $3^{η}$ ερώτηση.
5. Κλειδιού:
   Κάθε σχέση πρέπει να έχει τουλάχιστον ένα υπερκλείδι, καθώς και όλες οι πλειάδες πρέπει να είναι διαφορετικές και τέλος για δύο διαφορετικές πλειάδες πρέπει t1

και t2, όπου t1[sk] ≠ t2[sk], και sk (super key) το σύνολο των γνωρισμάτων της σχέσης R, r(R): στιγμιότυπο R. Πιο συγκεκριμένα για το project μας έχουν αναλυθεί στην 4η ερώτηση.

10 Καθορισμός κλειδιών των σχέσεων βάσει των συναρτησιακών εξαρτήσεων

Για τον simple_cust (citizen):

ac_id → {name, credit_limit, cur_due_amount, avail_amount, exp_date, active}

Για τον company_cust:

ac_id → {name, credit_limit, cur_due_amount, avail_amount, exp_date, active, employee_id, employee_name}

Για τον merchant:

ac_id → {name, supply, profit, debt}

Για τα transaction:

tran_id → {cust_name, merch_name, tran_date, tran_amount, kind, c_tran, emp_id}

11 Μετατροπή μοντέλου σε 3η κανονική μορφή με διατήρηση των συναρτησιακών εξαρτήσεων και χωρίς απώλεια πληροφορίας

Σύμφωνα με τον ορισμό της BCNF, όταν για κάθε συναρτησιακή εξάρτηση X → Y ισχύει ένα από τα εξής:

- είτε Y ∈ X (τετριμμένη περίπτωση) ή
- X περιέχει το κλειδί της R.

τότε το μοντέλο μας βρίσκεται σε BCNF.

Άρα αφού το μοντέλο βρίσκεται σε BCNF βρίσκεται και σε τρίτη κανονική μορφή.

12 Περιγραφή ερωτήσεων προς τη βάση δεδομένων με SQL

Παρακάτω παραθέτονται οι ερωτήσεις (query) προς τη βάση δεδομένων με SQL:

Στην pay_due_mer.php :

```
$query = "UPDATE merchant SET debt=($due-$am) WHERE ac_id=$cust_id";

$query = "SELECT debt FROM merchant WHERE ac_id=$cust_id;";

$query = "UPDATE merchant SET debt=0 WHERE ac_id=$cust_id";
```

Στην search_ust.php :

```
$query = "SELECT name FROM customer WHERE ac_id=$cid";

$query = "SELECT * FROM transaction WHERE cust_name='$cname' AND tran_date BETWEEN '$date_from' AND '$date_until' ORDER BY tran_date DESC";
```

Στην search_merch.php :

```
$query = "SELECT * FROM transaction WHERE merch_name='$cname' AND tran_date BETWEEN '$date_from' AND '$date_until' ORDER BY tran_date DESC";
```

Στην bad_customer.php :

```
$query = "SELECT customer.name, customer.ac_id, simple_cust.cur_due_amount FROM customer INNER JOIN simple_cust ON customer.ac_id=simple_cust.ac_id WHERE customer.active='true' AND simple_cust.cur_due_amount>0 ORDER BY simple_cust.cur_due_amount DESC";

$query ="SELECT customer.name, customer.ac_id, merchant.debt FROM customer INNER JOIN merchant ON customer.ac_id=merchant.ac_id WHERE customer.active='true' AND merchant.debt>0 ORDER BY merchant.debt DESC";
```

Στην add_employee.php :

```
$query = "INSERT INTO company_cust (ac_id, employee_id, employee_name) VALUES (?,?,?)";
```

Στην cust_transaction.php:

```
$query = "SELECT customer.name FROM customer INNER JOIN merchant ON customer.ac_id=merchant.ac_id";

$query = "SELECT name FROM customer WHERE ac_id=$cid";

$query = "SELECT * FROM merchant WHERE ac_id= (SELECT ac_id FROM customer WHERE name='$merch')";

$query = "UPDATE simple_cust SET avail_amount=$av_amount WHERE ac_id=$cid";

$query = "UPDATE merchant SET profit=$profit, debt=$debt WHERE ac_id=$mid";

$query = "UPDATE simple_cust SET cur_due_amount=$c_d_amount WHERE ac_id=$cid";

"UPDATE merchant SET profit=$profit, debt=$debt WHERE ac_id=$mid";

"INSERT INTO transaction (tran_id, cust_name, merch_name, tran_date, tran_amount, kind, c_tran, emp_id) VALUES (NULL, ?, ?, NOW(), ?, ?,'false','')";

"SELECT * FROM transaction ORDER BY tran_date DESC LIMIT 1"
```

Στην delete_customer.php:

```
$query = "SELECT name FROM customer WHERE ac_id=$id";

$query = "UPDATE customer SET active='false' WHERE ac_id=$id";
```

Στην delete_merchant.php:

```
$query = "SELECT name FROM customer WHERE ac_id=$id";

$query = "SELECT * FROM merchant WHERE ac_id=$id";

$query = "UPDATE customer SET active='false' WHERE ac_id=$id";
```

Στην employee_transaction.php:

```
$query = "SELECT customer.name FROM customer INNER JOIN merchant ON
customer.ac_id=merchant.ac_id";

$query = "SELECT ac_id FROM company_cust WHERE employee_id = '$emp_id'";

$query = "SELECT name FROM customer WHERE ac_id = $cust_ac";

$query = "SELECT * FROM simple_cust WHERE ac_id = $cust_ac";

$query = "SELECT * FROM merchant WHERE ac_id= (SELECT ac_id FROM customer WHERE
name='$merch')";

$query = "UPDATE simple_cust SET avail_amount=$av_amount WHERE ac_id = $cust_ac";

$query = "UPDATE merchant SET profit=$profit, debt=$debt WHERE ac_id = $mid";

UPDATE simple_cust SET cur_due_amount=$c_d_amount WHERE ac_id=$cust_ac";

$query = "UPDATE merchant SET profit=$profit, debt=$debt WHERE ac_id=$mid";

$query = "INSERT INTO transaction (tran_id, cust_name, merch_name, tran_date, tran_amount, kind,
c_tran, emp_id) VALUES (NULL, ?, ?, NOW(), ?, ?, 'true', ?)";

$query = "SELECT * FROM transaction ORDER BY tran_date DESC LIMIT ";
```

Στην good_customer.php:

```
$query = "SELECT customer.name, customer.ac_id FROM customer INNER JOIN simple_cust ON
customer.ac_id=simple_cust.ac_id WHERE customer.active='true' AND simple_cust.cur_due_amount<=0";

$query ="SELECT customer.name, customer.ac_id FROM customer INNER JOIN merchant ON
customer.ac_id=merchant.ac_id WHERE customer.active='true' AND merchant.debt<=0";
```

Στην home.php:

```
$sql = "SELECT active FROM customer WHERE ac_id = $tmp";

$sql = "SELECT ac_id FROM company_cust WHERE ac_id =$tmp";

$sql = "SELECT ac_id FROM merchant WHERE ac_id =$tmp";

$sql = "SELECT employee_id, ac_id FROM company_cust WHERE employee_id = '$tmp'";

$sql="SELECT active From customer WHERE ac_id= $a";
```

Στην loged_ccc.php:

```
$sql = "SELECT * FROM simple_cust WHERE ac_id =$ac_id";

$sql1 = "SELECT name FROM customer WHERE ac_id =$ac_id";
```

Στην loged_customer.php:

```
$sql = "SELECT * FROM simple_cust WHERE ac_id =$ac_id";

$sql1 = "SELECT name FROM customer WHERE ac_id =$ac_id";
```

Στην loged_merch.php:

```
$sql = "SELECT * FROM merchant WHERE ac_id =$ac_id";

$sql1 = "SELECT name FROM customer WHERE ac_id =$ac_id";
```

Στην make_return.php:

```
$query = "SELECT name FROM customer WHERE ac_id=$c_id";

$query = "SELECT * FROM transaction WHERE cust_name='$cname' AND tran_amount>0";

$query = "SELECT * FROM simple_cust WHERE ac_id=$cid";

$query = "SELECT * FROM transaction WHERE tran_id=$chosen_trans";

$query = "SELECT ac_id FROM customer WHERE name='$merch'";

$query = "SELECT * FROM merchant WHERE ac_id=$mid";

$query = "UPDATE merchant SET profit=$profit, debt=$debt WHERE ac_id=$mid";

$query = "UPDATE simple_cust SET avail_amount=$av_amount WHERE ac_id=$cid";

$query = "UPDATE transaction SET tran_amount=0 WHERE tran_id=$chosen_trans  ";
```

Στην company_new_ac.php:

```
$query = "INSERT INTO customer (ac_id, name, active) VALUES (?,?,'true')";

$query2 = "INSERT INTO simple_cust (ac_id, credit_limit, cur_due_amount, avail_amount,exp_date)
VALUES (?,?,0,?,?)";

$query = "INSERT INTO company_cust (ac_id, employee_id, employee_name) VALUES (?,?,?)";
```

Στην month_merchant.php:

```
$sql="SELECT merch_name, COUNT(*) AS numoftrans FROM transaction WHERE tran_amount>0 AND
tran_date BETWEEN (CURRENT_TIMESTAMP - INTERVAL 30 DAY) AND (CURRENT_TIMESTAMP) GROUP BY
merch_name ORDER BY numoftrans DESC LIMIT 1";

$sql="SELECT * FROM merchant INNER JOIN customer ON customer.ac_id=merchant.ac_id WHERE
customer.name='$merch_name'";
```

Στην new_account.php:

```
$query = "INSERT INTO customer (ac_id, name, active) VALUES (?,?,'true')";

$query2 = "INSERT INTO simple_cust (ac_id, credit_limit, cur_due_amount, avail_amount,exp_date)
VALUES (?,?,0,?,?)";
```

Στην pay_due.php:

```
$query = "SELECT cur_due_amount FROM simple_cust WHERE ac_id=$cust_id;";

$query = "UPDATE simple_cust SET cur_due_amount=($due-$am) WHERE ac_id=$cust_id";

$query = "UPDATE simple_cust SET cur_due_amount=0 WHERE ac_id=$cust_id";
```

Στην merchant_new_ac.php:

```
$query = "INSERT INTO customer (ac_id, name, active) VALUES (?,?,'true')";

$query2 = "INSERT INTO merchant (ac_id, supply, profit , debt) VALUES (?,?,0,0)";
```

Στην pay_due_com.php:

```
$query = "SELECT cur_due_amount FROM simple_cust WHERE ac_id=$cust_id;";

$query = "UPDATE simple_cust SET cur_due_amount=($due-$am) WHERE ac_id=$cust_id";

$query = "UPDATE simple_cust SET cur_due_amount=0 WHERE ac_id=$cust_id";
```

13 Κώδικα των προγραμμάτων τα οποία υλοποιούν τις διαδικασίες που καθορίζουν τα παραπάνω

Οι κώδικες των προγραμμάτων που υλοποιούν τις διαδικασίες που καθορίζουν τα παραπάνω, έχουν συμπεριληφθεί στον φάκελο που στάλθηκε, μαζί με αυτό εδώ το pdf.

14 Σύντομο εγχειρίδιο χρήσης συστήματος

Αφότου ενεργοποιήσουμε τον xampp server, παίρνουμε μέσα από το δοθέν αρχείο μας htdocs το home.php και το «σέρνουμε- μεταφέρουμε» στον ήδη ανοιγμένο browser μας, αλλάζοντας όμως το file path πάνω στην μπάρα αναζήτησης από *file:///(….)/XAMPP/htdocs/home.php* σε *localhost/home.php*.

Αρχική σελίδα: localhost/home.php:

Στην αρχική μας σελίδα μας εμφανίζει ένα πεδίο με όνομα «Account ID» και «Sign In as: Customer» ή «Administrator(CCC)». Άμα θέλουμε να συνδεθούμε ως ένας χρήστης (administrator) από την ίδια την εταιρεία της CCC ή ως κάποιος πελάτης (customer), όπου πρέπει να γράψουμε το Account ID στο πεδίο (να θυμίσουμε ότι το Account ID έχει θεωρηθεί από την εκφώνηση της άσκηση ότι είναι μοναδικό και ο καθένας από τους πελάτες είναι διακεκριμένος (π.χ. ένας ιδιώτης δεν μπορεί να είναι συγχρόνως και έμπορος) .Έτσι, σύμφωνα με αυτήν την λογική πληκτρολογώντας το Account ID και πατώντας «Sign In as : Customer» πηγαίνει απευθείας χωρίς να χρειάζεται καμία άλλη επεξήγηση στο αντίστοιχο λογαριασμό με το Account ID που βάλαμε. Εδώ πρέπει να τονιστεί ότι δεν μπορεί οποιοδήποτε απλός πελάτης να εγγραφεί από μόνος του στην εταιρεία CCC. Αυτό το κάνει ο administrator της CCC, δηλαδή άμα κάνουμε Sign In as: Administrator(CCC).

Σελίδα Administrator:

Σε αυτήν εδώ την σελίδα, που έχουμε μπει ως Administrator της CCC, έχουμε 6 επιλογές-κουμπιά τα οποία μπορούμε να επιλέξουμε:

- New Account , πατώντας αυτό το κουμπί ο administrator μπορεί να δημιουργήσει ένα νέο λογαριασμό ενός πελάτη
- Good Customer Status, πατώντας αυτό το κουμπί ο administrator έχει μια πλήρη λίστα των «καλών πελατών», δηλαδή των πελατών που δεν χρωστάνε τίποτα στην εταιρεία CCC (ή τους χρωστάει η CCC, για κάποιο λόγο, δηλαδή το debt του merchant ή το cur_due_amount του customer είναι μικρότερα του μηδενός)
- Bad Customer Status, πατώντας αυτό το κουμπί ο administrator έχει μια πλήρη λίστα των «κακών πελατών», δηλαδή των πελατών που χρωστάνε στην εταιρεία CCC
- Merchant of the Month, πατώντας αυτό το κουμπί ο administrator έχει τα στοιχεία του έμπορου του μήνα

New Account:

Σε αυτήν την σελίδα ο administrator κάνει μια νέα εγγραφή λογαριασμού ενός πελάτη στην βάση δεδομένων μας ως απλός πελάτη(citizen), ή άλλου είδους λογαριασμούς πατώντας τα κουμπιά «Company Account», για δημιουργία λογαριασμού για νέα εταιρεία ή «Merchant Account», για δημιουργία λογαριασμού για νέο έμπορο.

Σελίδα Customer:

Εφόσον έχει πληκτρολογήσει το Account ID του και πατώντας το «Sign In as: Customer», ο συνδεδεμένος απλός πελάτης της CCC, μπορεί να δει τα στοιχεία του λογαριασμού του και έπειτα έχει τις εξής επιλογές:

- Buy Something, πατώντας αυτό το κουμπί ο πελάτης με τον συγκεκριμένο λογαριασμό μπορεί να κάνει αγορά ενός προϊόντος, επιλέγοντας πρώτα τον έμπορο (από μια λίστα από εμπόρους), το είδος της συναλλαγής και τέλος πληκτρολογώντας το ποσό που θα δώσει
- Pay Due, πατώντας αυτό το κουμπί ο πελάτης με τον συγκεκριμένο λογαριασμό μπορεί να πληρώσει το οφειλόμενο ποσό
- Return, πατώντας αυτό το κουμπί ο πελάτης με τον συγκεκριμένο λογαριασμό μπορεί να επιστρέψει ένα προϊόν που έχει ήδη αγοράσει
- Transactions Status by Date, πατώντας αυτό το κουμπί ο πελάτης με τον συγκεκριμένο λογαριασμό μπορεί να τσεκάρει τις δοσοληψίες που έχει κάνει σύμφωνα με το «Mechant's Account», «Transaction's Date» και «Transaction's Kind»
- Delete Account, πατώντας αυτό το κουμπί και αν ο πελάτης δεν έχει οφειλόμενο ποσό στην CCC, μπορεί να διαγράψει τον λογαριασμό του (ουσιαστικά τον θέτει inactive)

Σελίδα Merchant:

Εφόσον έχει πληκτρολογήσει το Account ID του και πατώντας το «Sign In as: Customer», ο συνδεδεμένος έμπορος της CCC, μπορεί να δει τα στοιχεία του λογαριασμού του και έπειτα έχει τις εξής επιλογές:

- Pay Due, πατώντας αυτό το κουμπί ο πελάτης με τον συγκεκριμένο λογαριασμό μπορεί να πληρώσει το οφειλόμενο ποσό
- Transactions Status, πατώντας αυτό το κουμπί ο πελάτης με τον συγκεκριμένο λογαριασμό μπορεί να τσεκάρει τις δοσοληψίες για κάποιο συγκεκριμένο διάστημα
- Delete Account, πατώντας αυτό το κουμπί και αν ο έμπορος δεν έχει οφειλόμενο ποσό στην CCC, μπορεί να διαγράψει τον λογαριασμό του (ουσιαστικά τον θέτει inactive)

Σελίδα Company:

Εφόσον έχει πληκτρολογήσει το Account ID του και πατώντας το «Sign In as: Customer», η συνδεδεμένη εταιρεία της CCC, μπορεί να δει τα στοιχεία του λογαριασμού του και έπειτα έχει τις εξής επιλογές:

- Pay Due, πατώντας αυτό το κουμπί η εταιρεία με τον συγκεκριμένο λογαριασμό μπορεί να πληρώσει το οφειλόμενο ποσό
- Return, πατώντας αυτό το κουμπί η εταιρεία με τον συγκεκριμένο λογαριασμό μπορεί να επιστρέψει ένα προϊόν που έχει ήδη αγοράσει
- Transactions Status, πατώντας αυτό το κουμπί η εταιρεία με τον συγκεκριμένο λογαριασμό μπορεί να τσεκάρει τις δοσοληψίες για κάποιο συγκεκριμένο διάστημα και να καθορίσει από ποιους υπαλλήλους έγιναν
- Add Employee, πατώντας αυτό το κουμπί η εταιρεία προσθέτει κάποιο υπάλληλο
- Delete Account, πατώντας αυτό το κουμπί και αν η εταιρεία δεν έχει οφειλόμενο ποσό στην CCC, μπορεί να διαγράψει τον λογαριασμό του (ουσιαστικά τον θέτει inactive)

15 Ενδεικτικά αποτελέσματα από εκτέλεση διαδικασιών

Έχουμε παραδείγματα από πελάτες και δοσοληψίες μέσα στην βάση δεδομένων που έχουμε στείλει. Όταν τρέχουμε το πρόγραμμά μας, όλα λειτουργούν όπως θα έπρεπε. Δηλαδή ουσιαστικά τα test files μας είναι μέσα στην βάση δεδομένων.