



A R I S T O T L E
U N I V E R S I T Y
O F T H E S S A L O N I K I

Aristotle University of Thessaloniki
Department of Electrical and Computer Engineering

Αντάκης Μάριος 10619, mantakis@ece.auth.gr
Διακολουκάς Δημήτριος 10642, ddiakolou@ece.auth.gr
Καράτης Δημήτριος 10775, karatisd@ece.auth.gr
(Ομάδα 41)

BankDB - Βάση Δεδομένων Τράπεζας
(ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ)

Πρώτο Παραδοτέο

23/11/2025

Περιεχόμενα

1 Γενικότερη μοντελοποίηση του τραπεζικού συστήματος	3
2 Άλλαγές σε σχέση με την προηγούμενη υλοποιήση	5
3 Εισαγωγή	7
3.1 Σκοπός Εφαρμογής	7
3.2 Περιγραφή Εφαρμογής	7
3.3 Απαιτήσεις Εφαρμογής σε Δεδομένα	8
4 Κατηγορίες Χρηστών και Απαιτήσεις τους	9
4.1 Διαχειριστής Βάσης Δεδομένων Συστήματος (Admin)	9
4.2 Διευθυντής Καταστήματος	9
4.3 Τυπάλληλος Καταστήματος	10
4.4 Πελάτης	11
5 Μοντέλο Οντοτήτων/Συσχετίσεων	12
5.1 Γενική Περιγραφή	12
5.2 Καθορισμός Οντοτήτων	13
5.3 Καθορισμός Συσχετίσεων	16
5.4 Διάγραμμα Οντοτήτων/Συσχετίσεων	20
6 Σχεσιακό Μοντέλο	21
6.1 Πεδία Ορισμού	21
6.2 Σχέσεις	21
6.3 Σχεσιακό Διάγραμμα	26
6.4 Όψεις	26
6.4.1 CUSTOMER ACCOUNTS	26
6.4.2 EMPLOYEE BRANCH SUMMARY	27
6.4.3 LOAN DEBTS	28
6.4.4 ACCOUNTS BALANCE	29
6.4.5 CREDIT CARD BALANCE	30
6.4.6 V ACCOUNTS BALANCE LOAN	31
7 Παραδείγματα	32
7.1 Παραδείγματα Πινάκων	32
7.2 Παραδείγματα Ερωτημάτων	39

1 Γενικότερη μοντελοποίηση του τραπεζικού συστήματος

Παρακάτω παρατίθενται μια απλοποιημένη εκδοχή μιας βάσης δεδομένων τράπεζας. Αν και σε επόμενη ενότητα θα αναλύσουμε πλήρως τις υποθέσεις και παραδοχές που έγιναν για το σχεδιασμό της βάσης, σε αυτή την ενότητα θα εξηγήσουμε ακριβώς ποιες ενέργειες της τράπεζας έχουμε μοντελοποιήσει.

1. Για να λέγεται κάποιος πελάτης της τράπεζάς μας οφείλει να έχει τουλάχιστον έναν λογαριασμό σε αυτήν, και κάθε λογαριασμός στο άνοιγμά του έχει αρχικό ποσό ίσο με το μηδέν.
2. Κάθε πελάτης μπορεί να πάρει μέχρι ένα δάνειο.
3. Οι λογαριασμοί χωρίζονται σε Ταμιευτηρίου (Savings Accounts) και Τρεχούμενους (Checking Accounts), ενώ οι κάρτες σε Πιστωτικές (Credit Cards) και Χρεωστικές (Debit Cards). Λογαριασμοί πίστωσης δανείου και πιστωτικής θεωρούμε πως ανήκουν στις οντότητες Loan και Credit αντίστοιχα και δεν μοντελοποιούνται στην οντότητα Account για λόγους απλότητας του μοντέλου.
4. Η πιστωτική κάρτα ανήκει αυστηρά σε πελάτες, ενώ η χρεωστική σε λογαριασμούς. Η κάθε κάρτα σχετίζεται με ακριβώς έναν πελάτη ή λογαριασμό (για την περίπτωση της πιστωτικής και χρεωστικής αντίστοιχα), ενώ, κάθε πελάτης και λογαριασμός θεωρούμε ότι μπορεί να έχουν πολλαπλές κάρτες, πιστωτικές και χρεωστικές αντίστοιχα (για την περίπτωση πχ όπου μια κάρτα είναι για κάποιον λόγο μπλοκαρισμένη κλπ). Επιπλέον, κάναμε την παραδοχή ότι ένας πελάτης μπορεί να έχει πολλούς λογαριασμούς αλλά ένας λογαριασμός πρέπει υποχρεωτικά να ανήκει σε έναν ακριβώς πελάτη (θεωρήσαμε την μοντελοποίηση μόνο του κύριου δικαιούχου πχ). Οι απλοποιήσεις αυτές έγιναν για να μπορέσουμε να υπολογίσουμε τα παραγόμενα γνώρισματα Balance, CreditBalance και Debt, του λογαριασμού (Account), της πιστωτικής (Credit) και του δανείου (Loan), ευκολότερα.
5. Ο κάθε πελάτης μπορεί να πραγματοποιήσει συναλλαγές και μέσω της κάρτας του (μέσω της σχέσης Card -> performs -> Card Transaction), αλλά και μέσω του λογαριασμού του. Στη σχέση Card -> performs -> Card Transaction θεωρήσαμε ότι υπάγονται όλες οι αγορές που κάνει κάποιος πελάτης μέσω κάρτας, ενώ οι αναλήψεις και καταθέσεις που κάνει σε ATM μοντελοποιήθηκαν στη σχέση Account -> is affected by

-> Account Transaction, όπως αναφέρεται παρακάτω. Από την άλλη, μέσω ενός λογαριασμού του μπορεί να εκτελέσει τις εξής ενέργειες:

- Πληρωμή δόσης πιστωτικής κάρτας (μέσω της σχέσης Account -> funds -> Credit Payment Transaction)
- Μεταφορά, λήψη, πληρωμή δόσης δανείου, ανάληψη και κατάθεση χρημάτων (από ATM, μέσω κάρτας, ή και απευθείας από την τράπεζα) (μέσω της σχέσης Account -> is affected by -> Account Transaction). Εδώ θεωρούμε ότι εάν το γνώρισμα MovementType του Account Transaction έχει τιμή Transfer_IN τότε ο λογαριασμός δέχεται χρήματα από κάπου, αν είναι Transfer_OUT τότε στέλνει χρήματα, αν είναι Withdrawal ή Deposit τότε γίνεται ανάληψη ή κατάθεση από δικό του λογαριασμό (είτε από ATM ή απευθείας από κάποιο υποκατάστημα), αν είναι LoanPayment τότε εκτελεί πληρωμή δανείου. Να σημειωθεί, επίσης ότι θεωρήσαμε θετικές τιμές Amount σε περίπτωση που ο λογαριασμός λαμβάνει χρήματα (Deposit ή Transfer_IN) και αρνητικές σε αντίθετη περίπτωση. Αυτές οι διακρίσεις έγιναν ώστε τα παραγόμενα γνωρίσματα Balance, CreditBalance και Debt, του λογαριασμού (Account), της πιστωτικής (Credit) και του δανείου (Loan) αντίστοιχα, να είναι υπολογίσιμα ευκολότερα. Να σημειωθεί επίσης ότι έχουμε υποθέσει ότι οι πληρωμές των δόσεων δανείου και πιστωτικής μπορούν να γίνουν από οποιονδήποτε λογαριασμό που ανήκει στον πελάτη, ένω έχουμε επίσης κάνει την παραδοχή ότι οι οντότητες δάνειο (Loan) και πιστωτική (Credit) έχουν δικούς τους ξεχωριστούς λογαριασμούς πίστωσης που είναι ανεξάρτητοι από τους απλούς συμβατικούς λογαριασμούς που μοντελοποιούνται με χρήση της οντότητας Account.

6. Τέλος, το σύστημα έχει μοντελοποιηθεί ώστε να κρατά πληροφορίες για υπαλλήλους και υποκαταστήματα καθώς και πελάτες που αυτά εξυπηρετούν. Κάθε υποκατάστημα έχει μοναδική διεύθυνση και κάθε πελάτης και υπάλληλος το ίδιο. Θεωρήσαμε επίσης πως το κεντρικό κατάστημα της τράπεζας δεν εξυπηρετεί πελάτες και απλώς διαχειρίζεται την λειτουργία όλων των υπολοίπων.

2 Αλλαγές σε σχέση με την προηγούμενη υλοποιήση

- Στην οντότητα Δάνειο (Loan) προσθέσαμε το γνώρισμα Type, που δεικτοδοτεί τον τύπο του δανείου (π.χ στεγαστικό, καταναλωτικό, φοιτητικό κ.α) και αφαιρέθηκε το γνώρισμα LastPayment μιας και είναι πλεόν αχρείαστο.
- Στο διάγραμμα Ο/Σ η οντότητα κάρτα (Card) πλέον έχει αναβαθμιστεί σε υπερκλάση, με υποκλάσεις τις νέες πλέον Credit και Debit Cards. Αυτό έγινε ώστε να μπορέσουμε να συνδέσουμε ξεχωριστά τον Customer με την Credit και το Account με την Debit, όπως γίνεται και στις πραγματικές τράπεζες. Ένας πελάτης μπορεί να διαθέτει πολλές πιστωτικές αντίστοιχα, και κάθε πιστωτική πρέπει να συνδέεται υποχρεωτικά με έναν μόνο πελάτη (για λόγους απλότητας). Αντίστοιχα και για την σχέση του Account με την Debit και του Customer με το Account. Θεωρήσαμε επίσης ότι για να είναι κάποιος πελάτης της τράπεζας οφείλει να διατέθει λογαριασμό σε αυτήν (και όχι απαραίτητα πιστωτική).
- Επιπλέον η οντότητα Κίνηση (Transaction) μετατράπηκε και αυτή σε υπερκλάση με τις αντίστοιχες υποκλάσεις τις Card Transaction, Credit Payment Transaction και Account Transaction. Ο σκοπός αυτής της αλλαγής ήταν ο ορθότερος υπολογισμός των παραγόμενων γνωρισμάτων Debt, Balance και CreditBalance στα Loan, Account και Credit αντίστοιχα, καθώς και μια πιο σφαιρική μοντελοποίησης του τραπεζικού συστήματος. Έτσι, το Transaction πλέον μπορεί να μοντελοποίησει κινήσεις όχι μόνο λογαριασμού αλλά κάρτας και δανείου.
- Για τον σκοπό αυτόν προστέθηκαν 4 νέες σχέσεις:
(Account -> is affected by -> Account Transaction)
(Account -> funds -> Credit Payment Transaction)
(Credit Payment Transaction -> made for -> Credit Card)
(Card -> performs -> Card Transaction)

Έτσι, ένα Account μπορεί να κάνει πολλά CreditPaymentTransaction, και ένα CreditPaymentTransaction πρέπει να σχετίζεται με έναν ακριβώς λογαριασμό. Αντίστοιχη λογική εφαρμόζεται και για τις υπόλοιπες νέες σχέσεις.

Για την ορθή αποτύπωση των μεταφορών κεφαλαίων, υιοθετήθηκε η λογική της διπλής λογιστικής εγγραφής μέσω δύο διακριτών

συναλλαγών. Σε αυτή τη δομή, κάθε μεταφορά καταγράφεται ως δύο ανεξάρτητες εγγραφές στον πίνακα TRANSACTION με διαφορετικό TransactionID, οι οποίες όμως φέρουν ταυτόσημη ημερομηνία και ώρα, υποδεικνύοντας ότι πραγματοποιήθηκαν ταυτόχρονα ως μέρη της ίδιας ενέργειας. Οι κινήσεις αυτές εξειδικεύονται στον πίνακα ACCOUNT TRANSACTION, όπου μέσω του πεδίου MovementType η μία ορίζεται ως Transfer_OUT για τον λογαριασμό του αποστολέα και η άλλη ως Transfer_IN για τον λογαριασμό του παραλήπτη.

- Τέλος, προστέθηκαν μερικά ακόμα πραδείγματα ερωτημάτων για τις πράξεις της ένωσης, τομής, διαφοράς και συνάθροισης, όπως υποδείχθηκε στις διορθώσεις, καθώς και μερικές ακόμα όψεις που σχετίζονται με τον υπολογισμό των παραγόμενων γνωρισμάτων, ενώ εξαιτίας των παραπάνω αλλαγών χρειάστηκε να τροποποιηθούν και ορισμένα ερωτήματα, όψεις, παραδείγματα πινάκων κλπ.

3 Εισαγωγή

3.1 Σκοπός Εφαρμογής

Σκοπός της παρούσας εφαρμογής είναι η δημιουργία μιας Βάσης Δεδομένων που θα υποστηρίζει τις λειτουργίες μιας τράπεζας. Η εφαρμογή θα επιτρέπει: την παρακολούθηση λογαριασμών και των συναλλαγών τους, τη διαχείριση δανείων και καρτών, καθώς επίσης και τον έλεγχο των συμβολαίων μεταξύ πελατών και τράπεζας. Μέσω της Βάσης θα είναι δυνατή η εύκολη προσπέλαση, ενημέρωση και αναζήτηση πληροφοριών, με στόχο την οργάνωση και αυτοματοποίηση διαδικασιών που σχετίζονται με την εξυπηρέτηση των πελατών και τη εύρυθμη λειτουργία της τράπεζας.

3.2 Περιγραφή Εφαρμογής

Η εφαρμογή θα χρησιμοποιείται για την καταγραφή και διαχείριση των βασικών λειτουργιών μιας τράπεζας. Στη Βάση Δεδομένων θα αποθηκεύονται πληροφορίες που σχετίζονται με τους πελάτες, τους λογαριασμούς τους, τις τραπεζικές κάρτες, τα δάνεια, τις καθημερινές κινήσεις χρημάτων, καθώς και τα υποκαταστήματα και τους υπαλλήλους που το κάθε ένα διαθέτει. Πιο αναλυτικά, τα δεδομένα που θα αποθηκεύονται είναι:

- Πελάτες: Προσωπικά στοιχεία (όνομα, διεύθυνση, αριθμός ταυτότητας κτλ.).
- Λογαριασμοί: Κωδικός λογαριασμού, ποσό, ημερομηνία δημιουργίας, καθώς και οι κάρτες που συνδέονται με αυτόν.
- Τραπεζικές Κάρτες: Αριθμός κάρτας, ημερομηνία λήξης, στοιχεία κατόχου, στοιχεία κάρτας, κατάσταση κάρτας (ενεργή, παγωμένη κλπ) και πελάτης ή λογαριασμός με τον οποίο συνδέεται.
- Κινήσεις: Αναλήψεις, καταθέσεις, μεταφορές χρημάτων και πληρωμές, μαζί με ημερομηνίες και ποσά.
- Δάνεια: Ποσό δανείου, τύπος, επιτόκιο, ημερομηνία λήξης και ο πελάτης που το έχει λάβει, μαζί με τον εγγυητή, δηλαδή το πρόσωπο που αναλαμβάνει να αποπληρώσει το δάνειο αν ο κύριος οφειλέτης αδυνατεί.
- Υποκαταστήματα: Τοποθεσία, ωράριο λειτουργίας, προσωπικό που εργάζεται εκεί κλπ.

- Της πάλληλοι: Προσωπικά στοιχεία, θέση εργασίας κλπ.

Χρήστες της εφαρμογής θα είναι οι υπάλληλοι της τράπεζας, οι οποίοι θα διαχειρίζονται και θα ενημερώνουν τα στοιχεία πελατών, λογαριασμών, δανείων κ.ά καθώς και οι διευθυντές καταστημάτων, οι οποίοι θα ελέγχουν τα καταστήματα και το προσωπικό. Ακόμα, χρήστες ενδείκνυται να είναι επίσης οι πελάτες της τράπεζας, οι οποίοι θα μπορούν να ενημερώνονται για τα προϊόντα τους (λογαριασμούς, κάρτες, κινήσεις) online, μέσω σχετικής εφαρμογής, ενώ τέλος, ο/οι διαχειριστές της βάσης δεδομένων θα είναι υπεύθυνοι για την πλήρη διαχείριση των δεδομένων της βάσης, τη ρύθμιση των δικαιωμάτων πρόσβασης κ.α.

3.3 Απαιτήσεις Εφαρμογής σε Δεδομένα

Η εκτίμηση του μεγέθους της βάσης δεδομένων βασίστηκε σε παραδοχές σχετικά με το πλήθος των εγγραφών και το μέσο μέγεθος ανά εγγραφή για κάθε οντότητα. Κάθε οντότητα (π.χ. Πελάτης, Λογαριασμός, Κίνηση) αναλύθηκε ξεχωριστά, λαμβάνοντας υπόψη τον αριθμό των εγγραφών που αναμένονται για κάθε τύπο δεδομένων, όπως για παράδειγμα τις κινήσεις που εκτιμήθηκαν σε 86.400.000 ετησίως. Να σημειωθεί ότι για την περίπτωση IS A σχέσων, εκτιμήθηκε σαν σύνολο η υπερκλάση και όχι οι αντίστοιχες υποκλάσεις ξεχωριστά. Στη συνέχεια, το συνολικό μέγεθος υπολογίστηκε πολλαπλασιάζοντας το πλήθος των εγγραφών με το μέσο μέγεθος ανά εγγραφή, ώστε να προκύψει το συνολικό μέγεθος για κάθε οντότητα και κατά συνέπεια, για όλη τη βάση δεδομένων. Προφανώς οι υπολογισμοί για το μέγεθος της βάσης δεν είναι απόλυτα ακριβής, καθώς βασίζονται σε εκτιμήσεις και πολύ αβέβαια δεδομένα πάνω στον τραπεζικό τομέα. Παρακάτω φαίνεται το σχετικό πλήθος και μέγεθος εγγραφών σε μορφή πίνακα:

Οντότητα	Πλήθος Εγγραφών	Μ.Ο. μεγέθους / εγγραφή	Συνολικό Μέγεθος
Πελάτης	200.000	1 KB	200 MB
Λογαριασμός	360.000	0.5 KB	180 MB
Κάρτα	360.000	0.2 KB	72 MB
Κίνηση	86.400.000 / έτος	0.2 KB	17.3 GB
Δάνειο	40.000	0.5 KB	20 MB
Τιποκατάστημα	120	0.5 KB	60 KB
Τπάλληλος	2.000	0.5 KB	1 MB

4 Κατηγορίες Χρηστών και Απαιτήσεις τους

4.1 Διαχειριστής Βάσης Δεδομένων Συστήματος (Admin)

Έχει ως ευθύνη τη συνολική διαχείριση της βάσης δεδομένων και τη ρύθμιση των δικαιωμάτων πρόσβασης. Τα δικαιώματά του περιλαμβάνουν:

- Πλήρη πρόσβαση σε όλα τα δεδομένα της βάσης.
- Δημιουργία, τροποποίηση και διαγραφή πινάκων, σχέσεων και χρηστών.
- Παρακολούθηση της ασφάλειας και της σωστής λειτουργίας του συστήματος.

Πιο συγκεκριμένα, ο Διαχειριστής της βάσης (Admin) είναι ο ‘παντοδύναμος’ χρήστης της τράπεζας. Έχει πλήρη έλεγχο (ALL PRIVILEGES) σε όλους τους πίνακες και όψεις, ενώ μέσω της εντολής WITH GRANT OPTION μπορεί να δίνει ή να αφαιρεί δικαιώματα από άλλους χρήστες. Ορίστηκε με localhost διότι, για λόγους ασφαλείας, η διαχείριση της βάσης πρέπει να γίνεται μόνο τοπικά από τον κεντρικό υπολογιστή (server) της τράπεζας, ώστε να μην μπορεί κανένας να τον ‘χακάρει’, ενδεχομένως, εξ αποστάσεως. Σχετικός κώδικας βρίσκεται στο αρχείο users.sql.

4.2 Διευθυντής Καταστήματος

Έχει ως ευθύνη την επίβλεψη του προσωπικού και των λειτουργιών του καταστήματος. Τα δικαιώματά του περιλαμβάνουν:

- Πρόσβαση και τροποποίηση στοιχείων υπαλλήλων και πρόσβαση σε στοιχεία πελατών του καταστήματος.
- Έγκριση ή απόρριψη αιτήσεων για δάνεια.

Πιο συγκεκριμένα, ο Διευθυντής Καταστήματος (Manager) διαθέτει έναν διευρυμένο εποπτικό και διοικητικό ρόλο. Έχει πλήρη δικαιώματα ανάγνωσης (SELECT) σε όλα τα επίπεδα της βάσης (πελάτες, λογαριασμούς, κάρτες και συναλλαγές), ώστε να διατηρεί μια συνολική εικόνα της δραστηριότητας του υποκαταστήματος. Ωστόσο, τα δικαιώματα τροποποίησης (INSERT, UPDATE) περιορίζονται αυστηρά στη διαχείριση του προσωπικού (Employee), των υποκαταστημάτων (Bank Branch) και των δανείων (Loan), επιτρέποντάς του να προβαίνει σε προσλήψεις, οργανωτικές αλλαγές και εγκρίσεις δανειακών αιτημάτων. Ο χρήστης ορίστηκε με το σύμβολο %, επιτρέποντάς του την απομακρυσμένη σύνδεση για τον έλεγχο της τράπεζας εκτός γραφείου, ενώ ο σχετικός κώδικας βρίσκεται στο αρχείο users.sql.

4.3 Υπάλληλος Καταστήματος

Έχει ως ευθύνη την εξυπηρέτηση πελατών και την εκτέλεση τραπεζικών συναλλαγών. Τα δικαιώματά του περιλαμβάνουν:

- Εισαγωγή και ενημέρωση στοιχείων πελατών και λογαριασμών.
Εισαγωγή δανείων και κινήσεων.

Πιο συγκεκριμένα, ο Υπάλληλος Καταστήματος (bank_emp) έχει έναν πιο επιχειρησιακό ρόλο, εστιάζοντας στην εξυπηρέτηση των πελατών. Διαθέτει δικαιώματα (SELECT, INSERT, UPDATE) για τη διαχείριση των πελατών και των τραπεζικών λογαριασμών, επιτρέποντάς του να ανοίγει νέους λογαριασμούς και να ενημερώνει στοιχεία επικοινωνίας. Στον τομέα των συναλλαγών και των δανείων, έχει δικαίωμα εισαγωγής νέων εγγραφών (INSERT) για την πραγματοποίηση κινήσεων ή την υποβολή αιτήσεων δανεισμού, αλλά δεν του επιτρέπεται καμία επιλέον τροποποίηση (UPDATE) σε καταχωρημένα δεδομένα, διασφαλίζοντας την ακεραιότητα του ιστορικού. Όσον αφορά το προσωπικό, η πρόσβασή του είναι αυστηρά περιορισμένη σε βασικά στοιχεία επικοινωνίας των συναδέλφων του, ενώ δεν έχει πρόσβαση σε ευαίσθητα δεδομένα όπως ημερομηνίες γέννησης και ΑΦΜ. Τέλος, ο χρήστης ορίστηκε με localhost, περιορίζοντας τη σύνδεσή του αποκλειστικά από τους τερματικούς σταθμούς του καταστήματος για λόγους ασφαλείας. Σχετικός κώδικας βρίσκεται στο αρχείο users.sql.

4.4 Πελάτης

Έχει ως ευθύνη τη διαχείριση των δικών του καρτών και λογαριασμών. Τα δικαιώματά του περιλαμβάνουν:

- Πρόσβαση μόνο στα προσωπικά του δεδομένα, όπως λογαριασμούς, κάρτες, κινήσεις.
- Ενημέρωση συγκεκριμένων προσωπικών στοιχείων όπως για παράδειγμα το Email και η διεύθυνση κατοικίας.
- Υποβολή αιτήσεων για νέα προϊόντα (π.χ. δάνεια, κάρτες).

Τέλος, ο ρόλος του Πελάτη (bank_customer) έχει σχεδιαστεί με γνώμονα την απόλυτη προστασία της ιδιωτικότητας και την αρχή του ελάχιστου προνομίου. Αντί για απευθείας πρόσβαση στους πίνακες της βάσης, ο πελάτης συνδέεται αποκλειστικά με την «έξυπνη» όψη

V_CUSTOMER_ACCOUNTS_LOAN, η οποία χρησιμοποιεί τη συνάρτηση USER() για να ταυτοποιήσει το ΑΦΜ (TIN) του συνδεδεμένου χρήστη και να φιλτράρει δυναμικά τα δεδομένα. Με αυτόν τον τρόπο, ο πελάτης έχει δικαιώματα ανάγνωσης (SELECT) μόνο για τους δικούς του λογαριασμούς, το υπόλοιπο αυτών (AccountBalance) και τα ενεργά του δάνεια. Τα δικαιώματα τροποποίησης (UPDATE) είναι εξαιρετικά περιορισμένα και αφορούν αποκλειστικά τα πεδία του Email και της Διεύθυνσης, διασφαλίζοντας ότι οποιαδήποτε κρίσιμη αλλαγή στοιχείων ή οικονομική ρύθμιση απαιτεί τη φυσική παρουσία σε υποκατάστημα.

Αξίζει να σημειωθεί ότι στην πλήρη ανάπτυξη του συστήματος, θα έπρεπε να προστεθούν επιπλέον «έξυπνες» όψεις (π.χ.

V_CUSTOMER_TRANSACTIONS, V_CUSTOMER_CARDS κλπ, οι οποίες θα λειτουργούν με την ίδια λογική του αυτόματου φιλτραρίσματος βάσει TIN. Αυτό θα επέτρεπε στον πελάτη να έχει μια πλήρη εικόνα του χαρτοφυλακίου του (ιστορικό συναλλαγών, διαθέσιμα υπόλοιπα πιστωτικών καρτών, κ.λπ.). Ο χρήστης ορίστηκε με το σύμβολο %, επιτρέποντάς του την πρόσβαση μέσω e-banking από οποιαδήποτε τοποθεσία, ενώ ο σχετικός κώδικας βρίσκεται στο αρχείο users.sql.

5 Μοντέλο Οντοτήτων/Συσχετίσεων

5.1 Γενική Περιγραφή

Οι βασικές οντότητες της βάσης δεδομένων είναι: Πελάτης (Customer), Λογαριασμός (Account), Λογαριασμός Ταμιευτηρίου (Savings Account), Τρεχούμενος Λογαριασμός (Checking Account), Κίνηση (Transaction), Κίνηση Κάρτας (Card Transaction), Κίνηση Λογαριασμού (Account Transaction), Κίνηση Πληρωμής Πιστωτικής (Credit Payment Transaction), Κάρτα (Card), Πιστωτική Κάρτα (Credit Card), Χρεωστική Κάρτα (Debit Card), Δάνειο (Loan), Τπάλληλος (Employee) και Τηλεοπτικό Σύστημα (Bank Branch).

Τηλεοπτικό Σύστημα:

- Κάθε πελάτης μπορεί να διατηρεί πολλαπλούς λογαριασμούς αλλά ένας λογαριασμός πρέπει να συνδέεται με έναν ακριβώς πελάτη (κύριος δικαιούχος). Παράλληλα, κάθε πελάτης πρέπει να έχει τουλάχιστον ένα λογαριασμό ώστε να θεωρείται πελάτης της τράπεζάς μας, ο οποίος έχει μηδενικό αρχικό ποσό κατά το ανοιγμά του.
- Ένας πελάτης μπορεί να εξυπηρετείται από πολλά υποκαταστήματα, και ένα υποκαταστήμα μπορεί να εξυπηρετεί πολλούς πελάτες. (Ο πελάτης υποχρεωτικά θα πρέπει να εξυπηρετείται από τουλάχιστον ένα υποκατάστημα, ενώ κάποιο υποκατάστημα μπορεί να μην εξυπηρετεί πελατεία πχ τα κεντρικά γραφεία της τράπεζας, τα οποία ασχολούνται με καίρια για την τράπεζα ζητήματα).
- Ένας πελάτης μπορεί να πάρει μέχρι ένα δάνειο, και ένα δάνειο μπορεί να σχετίζεται με πολλούς πελάτες, πχ δανειολήπτης και εγγυητής, δηλαδή το πρόσωπο που αναλαμβάνει να αποπληρώσει το δάνειο αν ο κύριος οφειλέτης αδυνατεί. Ωστόσο, κάθε δάνειο θα πρέπει να αντιστοιχεί σε τουλάχιστον ένα πελάτη.
- Ένας πελάτης μπορεί να διαθέτει πολλές πιστωτικές κάρτες, και κάθε πιστωτική πρέπει να συνδέεται με ακριβώς έναν πελάτη. Ωστόσο, ένας πελάτης μπορεί και να μην διαθέτει καμία πιστωτική κάρτα. Αντίστοιχα και για την χρεωστική με τον λογαριασμό.
- Οι λογαριασμοί μπορεί να είναι είτε τρεχούμενοι ή αποταμιευτικοί: Κάθε λογαριασμός μπορεί να είναι είτε αποταμιευτικός είτε τρεχούμενος, με τη διαφορά να καθορίζεται από τον τύπο του λογαριασμού (Savings Account ή Checking Account). Ο αποταμιευτικός

προσφέρει τόκους (InterestRate) για την αποθήκευση χρημάτων, ενώ ο τρεχούμενος παρέχει τη δυνατότητα να ξοδέψει ο πελάτης περισσότερα από το διαθέσιμο υπόλοιπο, μέχρι ένα όριο (OverdraftLimit).

- Οι κινήσεις/συναλλαγές χωρίζονται σε κινήσεις κάρτας (Card Transaction), λογαριασμού (Account Transaction) καθώς επίσης και κινήσεις που αφορούν πληρωμή δόσης πιστωτικής (Credit Payment Transaction). Έτσι, κάθε λογαριασμός μπορεί να κάνει πολλές κινήσεις λογαριασμού, αλλά κάθε συναλλαγή λογαριασμού πρέπει να έχει γίνει υποχρεωτικά από ακριβώς ένα λογαριασμό. (Κάθε Account Transaction πρέπει να προέρχεται από κάποιον λογαριασμό). Αντίστοιχη λογική εφαρμόζεται και για τις υπόλοιπες σχέσεις που αφορούν κινήσεις/συναλλαγές.
- Τέλος, κάθε υπάλληλος ανήκει σε ένα μόνο υποκατάστημα, αλλά ένα υποκατάστημα μπορεί να έχει πολλούς υπαλλήλους. Ωστόσο, κάθε υπάλληλος πρέπει να ανήκει σε τουλάχιστον ένα υποκατάστημα και ένα υποκατάστημα θα πρέπει να έχει τουλάχιστον έναν υπάλληλο. Θεωρούμε επίσης, ότι κάθε υποκατάστημα έχει μοναδική διεύθυνση και κάθε πελάτης και υπάλληλος το ίδιο.

Για την βάση μας υποθέσαμε επίσης ότι όλα τα γνωρίσματα των οντοτήτων θα πρέπει να έιναι μη μηδενικά (not NULL), εκτός από την περίπτωση του ξένου κλειδιού LoanID στον πίνακα CUSTOMER, καθώς μπορεί να υπάρχει πελάτης χωρίς δάνειο.

5.2 Καθορισμός Οντοτήτων

Όνομα Οντότητας	Customer
Περιγραφή	Πελάτης της τράπεζας.
Ιδιότητες	Ισχυρή Οντότητα (Strong Entity). Τποκλάσεις: - Τπερχλάσεις: -
Γνωρίσματα	CustomerID (PK), TIN, Name, Email (πλειότιμο), Tel (πλειότιμο), DoB, Address

Όνομα Οντότητας	Loan
Περιγραφή	Δάνειο που μπορεί να δώσει η τράπεζα.
Ιδιότητες	Ισχυρή Οντότητα. Τποκλάσεις: - Τπερχλάσεις: -
Γνωρίσματα	LoanID (PK), OpeningDate, ExpirationDate, InterestRate, Amount, Type, Status, Debt (παραγόμενο)

Όνομα Οντότητας	Account
Περιγραφή	Τραπεζικός λογαριασμός.
Ιδιότητες	Ισχυρή Οντότητα. Υπερχλάσεις: - Υποχλάσεις: Savings Account, Checking Account Σχέση : IS-A
Γνωρίσματα	AccountID (PK), AccountNumber, OpeningDate, Status, Currency, Balance (παραγόμενο)

Όνομα Οντότητας	Bank Branch
Περιγραφή	Υποκατάστημα της τράπεζας.
Ιδιότητες	Ισχυρή Οντότητα. Υπερχλάσεις: - Υποχλάσεις: -
Γνωρίσματα	BranchID (PK), Tel (πλειότιμο), Email (πλειότιμο), OperatingHours, Address

Όνομα Οντότητας	Employee
Περιγραφή	Εργαζόμενος σε υποκατάστημα.
Ιδιότητες	Ισχυρή Οντότητα. Υπερχλάσεις: - Υποχλάσεις: -
Γνωρίσματα	EmployeeID (PK), TIN, Name, Tel (πλειότιμο), Email (πλειότιμο), DoB, Address

Όνομα Οντότητας	Transaction
Περιγραφή	Συναλλαγή που πραγματοποιείται από έναν λογαριασμό ή μια κάρτα.
Ιδιότητες	Ισχυρή Οντότητα. Υπερχλάσεις: - Υποχλάσεις: Card Transaction, Credit Payment Transaction, Account Transaction Σχέση : IS-A
Γνωρίσματα	TransactionID (PK), Date, Time, Amount

Όνομα Οντότητας	Credit Payment Transaction
Περιγραφή	Συναλλαγή που πραγματοποιείται από έναν λογαριασμό για πληρωμή πιστωτικής κάρτας πελάτη.
Ιδιότητες	Ισχυρή Οντότητα. Υπερχλάσεις: Transaction. Υποχλάσεις: -
Γνωρίσματα	TransactionID (PK), Date, Time, Amount

Όνομα Οντότητας	Account Transaction
Περιγραφή	Συναλλαγή που πραγματοποιείται από έναν λογαριασμό (π.χ. μεταφορά, ανάληψη, κατάθεση χλπ).
Ιδιότητες	Ισχυρή Οντότητα. Τπερχλάσεις: Transaction. Τποκλάσεις: -
Γνωρίσματα	TransactionID (PK), MovementType, Date, Time, Amount

Όνομα Οντότητας	Card Transaction
Περιγραφή	Αγορά που πραγματοποιείται από μια κάρτα.
Ιδιότητες	Ισχυρή Οντότητα. Τπερχλάσεις: Transaction. Τποκλάσεις: -
Γνωρίσματα	TransactionID (PK), Date, Time, Amount, TerminalNumber

Όνομα Οντότητας	Card
Περιγραφή	Κάρτα που εκδίδεται για χρήση από πελάτη.
Ιδιότητες	Ισχυρή Οντότητα. Τπερχλάσεις: - Τποκλάσεις: Credit Card, Debit Card Σχέση : IS-A
Γνωρίσματα	CardID (PK), CardNumber, CardholderName, Status, CVV, ExpirationDate

Όνομα Οντότητας	Credit Card
Περιγραφή	Πιστωτική κάρτα πελάτη.
Ιδιότητες	Ισχυρή Οντότητα. Τπερχλάσεις: Card. Τποκλάσεις: -
Γνωρίσματα	CardID (PK), CardNumber, CardholderName, Status, CVV, ExpirationDate, CreditLimit, CreditBalance (παραγόμενο).

Όνομα Οντότητας	Debit Card
Περιγραφή	Χρεωστική κάρτα λογαριασμού.
Ιδιότητες	Ισχυρή Οντότητα. Τπερχλάσεις: Card. Τποκλάσεις: -
Γνωρίσματα	CardID (PK), CardNumber, CardholderName, Status, CVV, ExpirationDate, DailyTransactionLimit.

Όνομα Οντότητας	Savings Account
Περιγραφή	Τραπεζικός λογαριασμός για την αποθήκευση χρημάτων, τύπου ταμιευτηρίου.
Ιδιότητες	Ισχυρή Οντότητα. Υπερκλάσεις: Account. Υποκλάσεις: -
Γνωρίσματα	AccountID (PK), OpeningDate, ExpirationDate, Status, Currency, Balance (παραγόμενο), InterestRate.

Όνομα Οντότητας	Checking Account
Περιγραφή	Τραπεζικός λογαριασμός για την αποθήκευση χρημάτων, τύπου τρεχούμενος
Ιδιότητες	Ισχυρή Οντότητα. Υπερκλάσεις: Account. Υποκλάσεις: -
Γνωρίσματα	AccountID (PK), OpeningDate, ExpirationDate, Status, Currency, Balance (παραγόμενο), OverdraftLimit

5.3 Καθορισμός Συσχετίσεων

Όνομα Συσχέτισης	Customer_owns_Account
Περιγραφή	Κάθε πελάτης μπορεί να έχει έναν ή περισσότερους λογαριασμούς και ένας λογαριασμός πρέπει να ανήκει σε έναν μόνο πελάτη.
Ιδιότητες	HAS-A
Λόγος Πληθυκότητας	1:N
Συμμετοχή	Ολική Συμμετοχή του Customer Ολική Συμμετοχή του Account
Γνωρίσματα	-

Όνομα Συσχέτισης	Customer_takes_Loan
Περιγραφή	Κάθε πελάτης της τράπεζας μπορεί να επωφελείται από ένα δάνειο.
Ιδιότητες	HAS-A
Λόγος Πληθυκότητας	N:1
Συμμετοχή	Ολική Συμμετοχή του Loan Μερική συμμετοχή του Customer
Γνωρίσματα	-

Όνομα Συσχέτισης	Customer_served_by_Branch
Περιγραφή	Κάθε πελάτης της τράπεζας πρέπει να εξυπηρετείται από ένα υποκατάστημα.
Ιδιότητες	HAS-A
Λόγος Πληθικότητας	N:M
Συμμετοχή	Ολική Συμμετοχή του Customer Μερική συμμετοχή του BankBranch
Γνωρίσματα	-

Όνομα Συσχέτισης	Employee_belongs_to_BankBranch
Περιγραφή	Κάθε Τπάλληλος της τράπεζας πρέπει να ανήκει σε ένα υποκατάστημα.
Ιδιότητες	HAS-A
Λόγος Πληθικότητας	N:1
Συμμετοχή	Ολική Συμμετοχή του Employee Ολική συμμετοχή του BankBranch
Γνωρίσματα	-

Όνομα Συσχέτισης	Customer_has_Credit_Card
Περιγραφή	Κάθε πελάτης μπορεί να έχει καμία, μία ή περισσότερες πιστωτικές κάρτες, ενώ κάθε πιστωτική κάρτα μπορεί να συνδεθεί με ακριβώς έναν πελάτη.
Ιδιότητες	HAS-A
Λόγος Πληθικότητας	1:N
Συμμετοχή	Μερική Συμμετοχή του Customer. Ολική Συμμετοχή της Credit Card.
Γνωρίσματα	-

Όνομα Συσχέτισης	Account_connected_to_Debit_Card
Περιγραφή	Κάθε λογαριασμός μπορεί να έχει καμία, μία ή περισσότερες χρεωστικές κάρτες, ενώ κάθε χρεωστική κάρτα μπορεί να συνδεθεί με ακριβώς έναν λογαριασμό.
Ιδιότητες	HAS-A
Λόγος Πληθικότητας	1:N
Συμμετοχή	Μερική Συμμετοχή του Account. Ολική Συμμετοχή της Debit Card.
Γνωρίσματα	-

Όνομα Συσχέτισης	Account_is_affected_by_Account_Transaction
Περιγραφή	Κάθε λογαριασμός μπορεί να πραγματοποιεί πολλές συναλλαγές λογαριασμού, ενώ κάθε συναλλαγή λογαριασμού αντιστοιχεί σε ακριβώς έναν λογαριασμό.
Ιδιότητες	HAS-A
Λόγος Πληθικότητας	1:N
Συμμετοχή	Μερική Συμμετοχή του Account. Ολική Συμμετοχή του Account Transaction.
Γνωρίσματα	-

Όνομα Συσχέτισης	Account_funds_Credit_Payment_Transaction
Περιγραφή	Κάθε λογαριασμός μπορεί να πραγματοποιεί πολλές συναλλαγές πληρωμής πιστωτικής δόσης, ενώ κάθε τέτοια συναλλαγή αντιστοιχεί σε ακριβώς έναν λογαριασμό.
Ιδιότητες	HAS-A
Λόγος Πληθυκότητας	1:N
Συμμετοχή	Μερική Συμμετοχή του Account. Ολική Συμμετοχή του Credit Payment Transaction.
Γνωρίσματα	-

Όνομα Συσχέτισης	Credit_Payment_Transaction_made_for_Credit_Card
Περιγραφή	Πληρωμή δόσης πιστωτικής από έναν λογαριασμό.
Ιδιότητες	HAS-A
Λόγος Πληθυκότητας	N:1
Συμμετοχή	Μερική Συμμετοχή του Credit Card. Ολική Συμμετοχή του Credit Payment Transaction.
Γνωρίσματα	-

Όνομα Συσχέτισης	Card_performs_Card_Transaction
Περιγραφή	Αγορά μέσω κάρτας (πιστωτικής ή χρεωστικής).
Ιδιότητες	HAS-A
Λόγος Πληθυκότητας	1:N
Συμμετοχή	Μερική Συμμετοχή του Card. Ολική Συμμετοχή του Card Transaction.
Γνωρίσματα	-

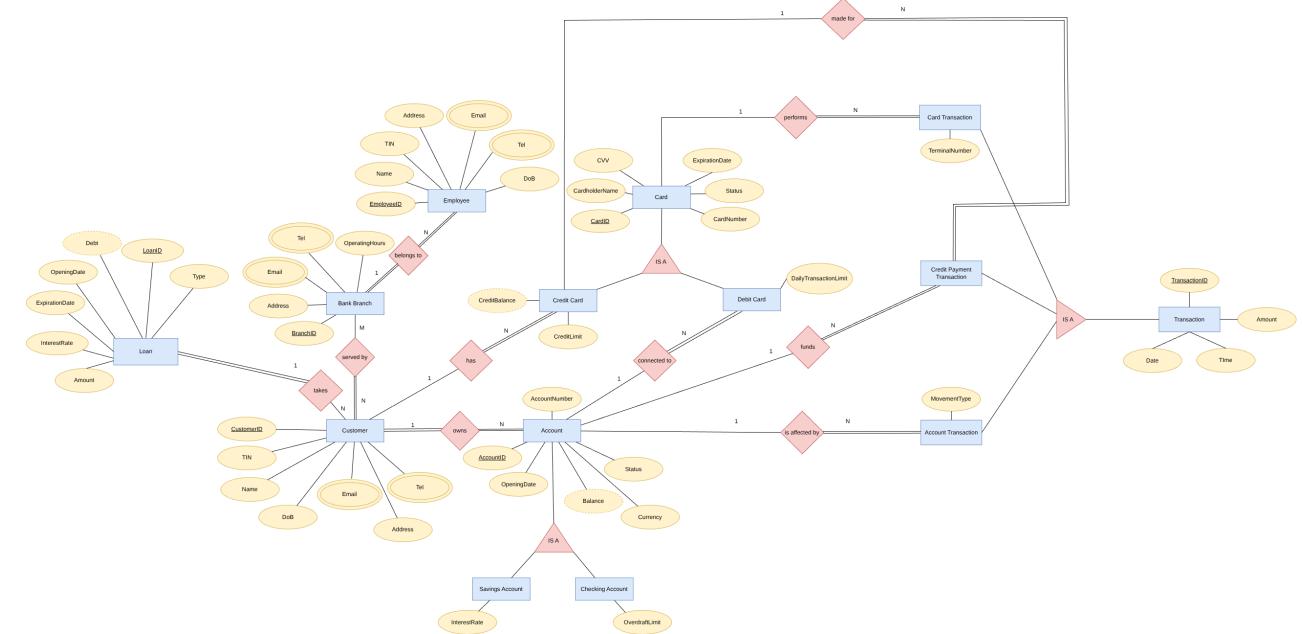
Όνομα Συσχέτισης	Account_is_a_Savings_Checking
Περιγραφή	Ο λογαριασμός εξειδικεύεται σε δύο υποκατηγορίες: Savings Account και Checking Account. Κάθε λογαριασμός ανήκει υποχρεωτικά σε μία από τις δύο υποκλάσεις.
Ιδιότητες	IS-A
Λόγος Πληθυκότητας	1:1
Συμμετοχή	Συσχέτιση Μέρους - Συνόλου
Γνωρίσματα	-

Όνομα Συσχέτισης	Card_is_a_Credit_Debit
Περιγραφή	Η κάθε κάρτα χωρίζεται σε πιστωτική (Credit) και χρεωστική (Debit).
Ιδιότητες	IS-A
Λόγος Πληθυκότητας	1:1
Συμμετοχή	Συσχέτιση Μέρους - Συνόλου
Γνωρίσματα	-

Όνομα Συσχέτισης	Transaction_is_a_Card_Account_Credit_Payment
Περιγραφή	Η κάθε συναλλαγή χωρίζεται σε Card Transaction, Account Transaction και Credit Payment Transaction.
Ιδιότητες	IS-A
Λόγος Πληθυκότητας	1:1
Συμμετοχή	Συσχέτιση Μέρους - Συνόλου
Γνωρίσματα	-

5.4 Διάγραμμα Οντοτήτων/Συσχετίσεων

Παρακάτω φαίνεται το Διάγραμμα Οντοτήτων/Συσχετίσεων για την βάση δεδομένων τράπεζας (BankDB), ενώ παραθέτουμε επίσης και τον σχετικό σύνδεσμο για ευκολία: [Διάγραμμα Ο/Σ Draw.io](#).



Σχήμα 1: Διάγραμμα Οντοτήτων Συσχετήσεων BankDB

6 Σχεσιακό Μοντέλο

6.1 Πεδία Ορισμού

Πεδίο Ορισμού	Τύπος Δεδομένων	Περιγραφή
ID	INT	Χρησιμοποιείται για primary keys
TIN	CHAR(9)	ΑΦΜ πελάτη
Όνομα προσώπου	VARCHAR(40)	Ονόματα πελατών και υπαλλήλων
Email	VARCHAR(40)	Ηλεκτρονική διεύθυνση
Τηλέφωνο	CHAR(10)	Τηλεφωνικός αριθμός
Διεύθυνση	VARCHAR(60)	Διεύθυνση προσώπου ή υποκαταστήματος
Ημερομηνία	DATE	Ημερομηνίες (DoB, OpeningDate κ.ά.)
Ώρα	TIME	Χρησιμοποιείται για να καταγράψει την ώρα μια συναλλαγής
Ώρες λειτουργίας	CHAR(11)	Ώρες λειτουργίας υποκαταστήματος (HH:MM-HH:MM)
Ποσό	DECIMAL(15,2)	Χρηματικά ποσά (Balance, Amount, Debt, OverdraftLimit...)
Επιτόκιο	DECIMAL(4,2)	InterestRate λογαριασμών, δανείων κ.λπ.
Κατάσταση	VARCHAR(15)	Status λογαριασμού/δανείου/χάρτας
Νόμισμα	CHAR(3)	ISO currency code (EUR, USD κ.λπ.)
CVV	CHAR(3)	Κωδικός ασφαλείας χάρτας
Αριθμός χάρτας	CHAR(16)	Αριθμός Κάρτας (CardID στα διαγράμματά μας)
Αριθμός Λογαρ.	CHAR(27)	Αριθμός Λογαριασμού (AccountID στα διαγράμματά μας)
Τύπος	VARCHAR(20)	Τύπος συναλλαγής, χάρτας κ.λπ.

6.2 Σχέσεις

Όνομα Σχέσης: CUSTOMER	
Γνωρίσματα:	
'Όνομα	Τύπος
CustomerID	ID
TIN	TIN
Name	Όνομα προσώπου
DoB	Ημερομηνία
Email (πλειότυπο)	Email
Tel (πλειότυπο)	Τηλέφωνο
Address	Διεύθυνση
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	CustomerID
Ξένα Κλειδιά	LoanID → LOAN

Όνομα Σχέσης: BANK BRANCH	
Γνωρίσματα:	
Όνομα	Τύπος
BranchID	ID
Tel (πλειότυπο)	Τηλέφωνο
Email (πλειότυπο)	Email
OperatingHours	Ώρες λειτουργίας
Address	Διεύθυνση
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	BranchID
Ξένα Κλειδιά	-

Όνομα Σχέσης: EMPLOYEE	
Γνωρίσματα:	
Όνομα	Τύπος
EmployeeID	ID
TIN	TIN
Name	Όνομα προσώπου
Tel (πλειότυπο)	Τηλέφωνο
Email (πλειότυπο)	Email
DoB	Ημερομηνία
Address	Διεύθυνση
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	EmployeeID
Ξένα Κλειδιά	BranchID → BANK BRANCH

Όνομα Σχέσης: ACCOUNT	
Γνωρίσματα:	
Όνομα	Τύπος
AccountID	ID
AccountNumber	Αριθμός Λογαρ.
OpeningDate	Ημερομηνία
Currency	Νόμισμα
Status	Κατάσταση
Balance (παραγόμενο)	Ποσό
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	AccountID
Ξένα Κλειδιά	CustomerID → CUSTOMER

Όνομα Σχέσης: LOAN	
Γνωρίσματα:	
Όνομα	Τύπος
LoanID	ID
Amount	Ποσό
Debt (παραγόμενο)	Ποσό
InterestRate	Επιτόκιο
Status	Κατάσταση
OpeningDate	Ημερομηνία
ExpirationDate	Ημερομηνία
Type	Τύπος
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	LoanID
Ξένα Κλειδιά	-

Όνομα Σχέσης: SAVINGS ACCOUNT	
Γνωρίσματα:	
Όνομα	Τύπος
AccountID	ID
InterestRate	Επιτόκιο
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	AccountID
Ξένα Κλειδιά	AccountID → ACCOUNT

Όνομα Σχέσης: CHECNKING ACCOUNT	
Γνωρίσματα:	
Όνομα	Τύπος
AccountID	ID
OverdraftLimit	Ποσό
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	AccountID
Ξένα Κλειδιά	AccountID → ACCOUNT

Όνομα Σχέσης: CARD	
Γνωρίσματα:	
Όνομα	Τύπος
CardID	ID
CardNumber	Αριθμός Κάρτας
CardholderName	Όνομα προσώπου
CVV	CVV
ExpirationDate	Ημερομηνία
Status	Κατάσταση
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	CardID
Ξένα Κλειδιά	-

Όνομα Σχέσης: CREDIT CARD	
Γνωρίσματα:	
Όνομα	Τύπος
CardID	ID
CreditLimit	Ποσό
CreditBalance (παραγόμενο)	Ποσό
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	CardID
Ξένα Κλειδιά	CardID → CARD CustomerID → CUSTOMER

Όνομα Σχέσης: DEBIT CARD	
Γνωρίσματα:	
Όνομα	Τύπος
CardID	ID
DailyTransactionLimit	Ποσό
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	CardID
Ξένα Κλειδιά	CardID → CARD AccountID → ACCOUNT

Όνομα Σχέσης: TRANSACTION	
Γνωρίσματα:	
Όνομα	Τύπος
TransactionID	ID
Date	Ημερομηνία
Time	Ώρα
Amount	Ποσό
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	TransactionID
Ξένα Κλειδιά	TransactionID → TRANSACTION

Όνομα Σχέσης: CARD TRANSACTION	
Γνωρίσματα:	
Όνομα	Τύπος
TransactionID	ID
TerminalNumber	ID
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	TransactionID
Ξένα Κλειδιά	TransactionID → TRANSACTION CardID → CARD

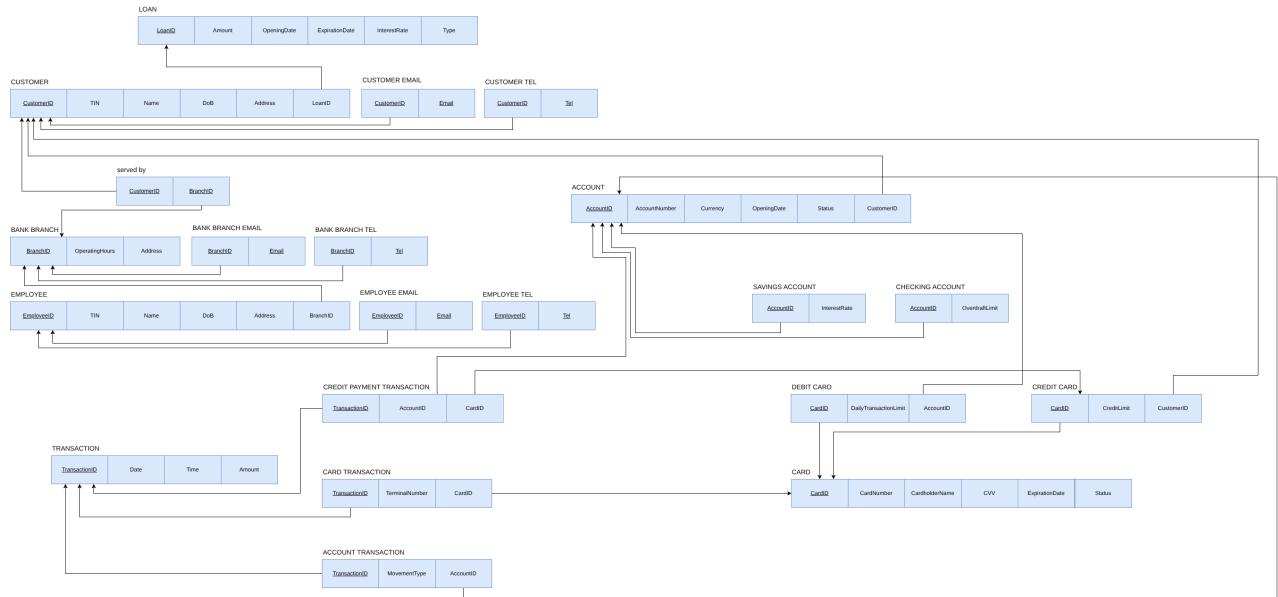
'Όνομα Σχέσης: ACCOUNT TRANSACTION	
Γνωρίσματα:	
'Όνομα	Τύπος
TransactionID	ID
MovementType	Τύπος
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	TransactionID
Ξένα Κλειδιά	TransactionID → TRANSACTION

'Όνομα Σχέσης: CREDIT PAYMENT TRANSACTION	
Γνωρίσματα:	
'Όνομα	Τύπος
TransactionID	ID
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	TransactionID
Ξένα Κλειδιά	TransactionID → TRANSACTION AccountID → ACCOUNT CardID → CREDIT CARD

'Όνομα Σχέσης: served by	
Γνωρίσματα:	
'Όνομα	Τύπος
CustomerID	ID
BranchID	ID
Περιορισμοί Ακεραιότητας:	
Πρωτεύον Κλειδί	(CustomerID, BranchID)
Ξένα Κλειδιά	CustomerID → CUSTOMER, BranchID → BANK BRANCH

6.3 Σχεσιακό Διάγραμμα

Παρακάτω φαίνεται το Διάγραμμα Οντοτήτων/Συσχετίσεων για την βάση δεδομένων τράπεζας (BankDB), ενώ παραθέτουμε επίσης και τον σχετικό σύνδεσμο για ευκολία: [Σχεσιακό Διάγραμμα Draw.io](#).



Σχήμα 2: Σχεσιακό Διάγραμμα BankDB

6.4 Όψεις

Παρακάτω παραθέτουμε ενδεικτικά δύο από τις πιο σημαντικές όψεις της βάσης. Επιπλέον, θεωρήσαμε πως όλα τα γνωρίσματα πρέπει να είναι μη μηδενικά (not NULL) (εκτός από την περίπτωση του ξένου κλειδιού LoanID στον πίνακα CUSTOMER).

6.4.1 CUSTOMER ACCOUNTS

- Σκοπός:** Συγκεντρώνει βασικές πληροφορίες για όλους τους πελάτες και τους λογαριασμούς που διαθέτουν.
- Περιλαμβάνει:** CustomerID, Name, TIN (από CUSTOMER), AccountID, AccountNumber, Currency (από ACCOUNT).
- Σύνδεση:** CUSTOMER → owns → ACCOUNT.

Με χρήση σχεσιακής άλγεβρας έχουμε:

$$A = \left(\begin{array}{c} \Pi_{CustomerID, Name, TIN}(\text{CUSTOMER}) \\ \bowtie \\ \Pi_{AccountID, CustomerID, AccountNumber, Currency}(\text{ACCOUNT}) \end{array} \right)$$

και μετονομάζοντας την όψη για ευκολία σε CUSTOMER ACCOUNTS, προκύπτει η εξής έκφραση σχεσιακής άλγεβρας:

$$\rho_{\text{CUSTOMER ACCOUNTS}}(A)$$

6.4.2 EMPLOYEE BRANCH SUMMARY

- **Σκοπός:** Εμφανίζει το πλήθος και γενικές πληροφορίες υπαλλήλων που απασχολούνται από κάθε υποκατάστημα.
- **Περιλαμβάνει:** BranchID, Address (από BANK BRANCH), EmployeeID, Email, (από EMPLOYEE EMAIL), EmployeeID, Tel, (από EMPLOYEE TEL), EmployeeID, TIN, Name, Address (από EMPLOYEE).
- **Σύνδεση:** EMPLOYEE → belongs to → BANK BRANCH.

Με χρήση σχεσιακής άλγεβρας έχουμε:

Αρχικά θα μετονομάσουμε τα γνωρίσματα Address των πινάκων BANK BRANCH και EMPLOYEE ως BranchAddress και EmpAddress αντίστοιχα:

$$\begin{aligned} & \rho_{\text{BANK BRANCH}(\dots, \text{BranchAddress}, \dots)}(\text{BANK BRANCH}), \\ & \rho_{\text{EMPLOYEE}(\dots, \text{EmpAddress}, \dots)}(\text{EMPLOYEE})) \end{aligned}$$

Έπειτα έχουμε:

$$\Pi_{BranchID, BranchAddress}(\text{BANK BRANCH}) = A$$

και

$$\left(\begin{array}{c} (\text{EMPLOYEE EMAIL}) \bowtie (\text{EMPLOYEE TEL}) \\ \bowtie \\ \Pi_{EmployeeID, TIN, Name, EmpAddress, BranchID}(\text{EMPLOYEE}) \end{array} \right) = B$$

ενώ, τέλος, μετονομάζοντας τις παραπάνω εκφράσεις για λόγους απλότητας ως

$$\rho_{branch_info}(A)) \quad \text{και} \quad \rho_{employee_info}(B)$$

τελικά προκύπτει η ζητούμενη όψη:

$$\begin{aligned} \Pi_{Name, TIN, EmpAddress, Tel, Email, BranchID, BranchAddress}(&branch_info \bowtie employee_info) \\ &= C \end{aligned}$$

μετονομασμένη για ευκολία σε EMPLOYEE BRANCH SUMMARY, ως εξής:

$$\rho_{EMPLOYEE\ BRANCH\ SUMMARY}(C)$$

6.4.3 LOAN DEBTS

- **Σκοπός:** Συγκεντρώνει όλα τα δάνεια μαζί με το υπολοιπόμενο χρέος που το κάθε ένα έχει καθώς και σχετικές πληροφορίες για τον πελάτη (ή τους πελάτες συνδικαλούχους) που αυτά αφορούν.
- **Περιλαμβάνει:** CustomerID, Name, TIN (από CUSTOMER), LoanID, ExpirationDate, Type, Amount, (από LOAN) και το παραγόμενο Debt (από Loan).
- **Σύνδεση:** ACCOUNT TRANSACTION → IS A → TRANSACTION, ACCOUNT → is affected by → ACCOUNT TRANSACTION και CUSTOMER → takes → LOAN.

Με χρήση σχεσιακής άλγεβρας έχουμε:

Αρχικά, οι συνολικές πληρωμές, custLoanPaymentSum (B), υπολογίζονται ανά CustomerID μέσω φιλτραρίσματος και συνενώσεων (joins). Στη συνέχεια, αυτό το άθροισμα συνδέεται με τα στοιχεία του δανείου (C) χρησιμοποιώντας την κοινή στήλη CustomerID (υπό την παραδοχή ότι κάθε πελάτης μπορεί να πάρει μέχρι ένα δάνειο). Τέλος, το Debt παράγεται ως μια νέα στήλη, μέσω της Γενικευμένης Προβολής (Π), εκτελώντας την αριθμητική πρόσθεση $Debt = Amount + custLoanPaymentSum$. Να σημειωθεί ότι τα περιεχόμενα του custLoanPaymentSum είναι αρνητικά εκ κατασκευής/θεώρησης, όπως αναφέραμε και στις υποθέσεις πιο πάνω. Έτσι έχουμε:

$$A = \sigma_{MovementType=LoanPayment}(\text{ACC. TRANS.} \bowtie \text{TRANS.} \bowtie \text{ACCOUNT})$$

$$B = \underset{\text{CustomerID}}{\text{G}_{\text{sum}}}(\text{Amount}) \text{ as custLoanPaymentSum } (A)$$

$$C = \Pi_{CustomerID, Name, TIN, LoanID, ExpiratonDate, Type, Amount} (\text{CUSTOMER} \bowtie \text{LOAN})$$

$$D = C \text{ Left Outer Join } B$$

$$\Pi_{CustomerID, Name, TIN, LoanID, ExpiratonDate, Type, Amount, Debt=Amount+custLoanPaymentSum} (D)$$

$$= E$$

και μετονομάζοντας την όψη για ευκολία σε LOAN DEBTS, έχουμε:

$$\rho_{\text{LOAN DEBTS}}(E)$$

Αξίζει να σημειωθεί ότι χρησιμοποιήθηκε η πράξη του Left Outer Join μεταξύ των συνόλων C και B , ώστε να διασφαλιστεί ότι στην όψη θα συμπεριληφθούν όλα τα καταχωρημένα δάνεια, ακόμα και εκείνα για τα οποία δεν έχει πραγματοποιηθεί ακόμα καμία πληρωμή.

6.4.4 ACCOUNTS BALANCE

- **Σκοπός:** Συγκεντρώνει όλους τους λογαριασμούς μαζί με το διαθέσιμο υπόλοιπο που ο καθένας έχει.
- **Περιλαμβάνει:** AccountID (από ACCOUNT) και όλα τα σχετικά TRANSACTION που αφορούν λογαριασμό, όπως (ACCOUNT TRANSACTION), (CREDIT PAYMENT TRANSACTION) και (CARD TRANSACTION, μόνο στην περίπτωση της χρεωστικής κάρτας).
- **Σύνδεση:** TRANSACTION → IS A → ACCOUNT TRANSACTION, CREDIT PAYMENT TRANSACTION, CARD TRANSACTION, ACCOUNT → is affected by → ACCOUNT TRANSACTION, ACCOUNT → funds → CREDIT PAYMENT TRANSACTION, CARD → performs → CARD TRANSACTION, CARD → is a → DEBIT CARD, και DEBIT CARD → connected to → ACCOUNT.

Με χρήση σχεσιακής άλγεβρας έχουμε:

$$A = \text{ACC. TRANS.} \bowtie \text{TRANS.} \bowtie \text{ACCOUNT}$$

$$B = \underset{\text{AccountID}}{\text{G}_{\text{sum}}}(\text{Amount}) \text{ as AccSum}(A)$$

$$C = \text{CREDIT PAYMENT TRANS.} \bowtie \text{TRANS.} \bowtie \text{ACCOUNT}$$

$$D = \underset{\text{AccountID}}{\text{G}_{\text{sum}}}(\text{Amount}) \text{ as CrPaySum}(C)$$

$$E = \text{CARD TRANS.} \bowtie \text{TRANS.} \bowtie \text{CARD} \bowtie \text{DEBIT CARD} \bowtie \text{ACCOUNT}$$

$$F = \underset{\text{AccountID}}{\text{G}_{\text{sum}}}(\text{Amount}) \text{ as DebSum}(E)$$

$$G = B \bowtie D \bowtie F$$

$H = \Pi_{\text{AccountID}, \text{AccSum}, \text{CrPaySum}, \text{DebSum}(\text{AccSum} + \text{CrPaySum} + \text{DebSum})} \text{ as Balance}(G)$
 και μετονομάζοντας την όψη για ευκολία σε ACCOUNTS BALANCE,
 έχουμε:

$$\rho_{\text{ACCOUNTS BALANCE}}(H)$$

6.4.5 CREDIT CARD BALANCE

- **Σκοπός:** Συγκεντρώνει όλες τις πιστωτικές κάρτες με το διαθέσιμο υπόλοιπο που η κάθε μια έχει.
- **Περιλαμβάνει:** CardID (από CARD), CREDIT PAYMENT TRANSACTION, TRANSACTION και CREDIT CARD.
- **Σύνδεση:** TRANSACTION → IS A → CREDIT PAYMENT TRANSACTION, CREDIT PAYMENT TRANSACTION → made for → CREDIT CARD.

Με χρήση σχεσιακής άλγεβρας έχουμε:

$$A = \text{CREDIT PAYMENT TRANS.} \bowtie \text{TRANS.} \bowtie \text{CREDIT CARD}$$

$$B = \text{CardID} \mathcal{G}_{\text{sum}}(\text{Amount}) \text{ as CreditPaySum}$$

$$A = \text{CREDIT PAYMENT TRANS.} \bowtie \text{TRANS.} \bowtie \text{CREDIT CARD}$$

$$\Pi_{..., (\text{CreditLimit} + \text{CreditPaySum}) \text{ as Balance}}(A)$$

και μετονομάζοντας την όψη για ευκολία σε CREDIT CARD BALANCE, έχουμε:

$$\rho_{\text{CREDIT CARD BALANCE}}(B)$$

6.4.6 V ACCOUNTS BALANCE LOAN

- **Σκοπός:** Συγκεντρώνει βασικές πληροφορίες για έναν συγκεκριμένο πελάτη και τους λογαριασμούς/δάνεια που διαθέτει. Δημηιουργήθηκε συγκεκριμένα για τον bank customer του users.sql. Κάνει φιλτράρισμα με βάσει το ΑΦΜ (TIN) του πελάτη. Για περισσότερη επεξήγηση ανατρέξτε στο αρχείο users.sql.
- **Περιλαμβάνει:** CUSTOMER, CUSTOMER EMAIL, CUSTOMER TEL, ACCOUNTS, ACCOUNTS BALANCE, LOAN.
- **Σύνδεση:** CUSTOMER → takes → LOAN, CUSTOMER → owns → ACCOUNT, CUSTOMER → owns → ACCOUNT και ACCOUNTS BALANCE.

Με χρήση σχεσιακής άλγεβρας έχουμε:

$$A = \text{CUSTOMER} \bowtie \text{CUSTOMER EMAIL} \bowtie \text{CUSTOMER TEL}$$

$$B = \text{ACCOUNT} \bowtie \text{ACCOUNTS BALANCE} \bowtie \text{LOAN}$$

$$C = A \bowtie B$$

Τέλος, γίνεται φιλτράρισμα με βάση το ΑΦΜ (TIN) ώστε να εξάγουμε τον συγκεκριμένο πελάτη. Περισσότερες πληροφορίες μπορούν να βρεθούν στο αρχείο users.sql.

και μετονομάζοντας την όψη για ευκολία σε CREDIT CARD BALANCE, έχουμε:

$$\rho V \text{ ACCOUNTS BALANCE LOAN}(C)$$

7 Παραδείγματα

7.1 Παραδείγματα Πινάκων

Παρακάτω παρουσιάζονται ενδεικτικές εγγραφές για κάθε πίνακα της βάσης δεδομένων.

LOAN

<u>LoanID</u>	Amount	OpeningDate	ExpirationDate	InterestRate	Type
1	7500.43	2020-02-24	2025-02-15	3.12	Housing
2	2100.03	2022-03-21	2024-03-03	0.19	Housing
3	1523.13	2020-02-1	2025-04-15	4.12	Consumer
4	5002.30	2021-03-01	2024-04-21	2.20	Student
5	400.43	2021-12-15	2025-04-25	1.12	Consumer

CUSTOMER

<u>CustomerID</u>	TIN	Name	DoB	Address	LoanID
1	123456789	Maria Papadopoulou	1985-06-15	Ermou 56, 10555, Monastiraki	NULL
2	321456789	Maria Papadopoulou	1991-01-05	Stadiou 1, 10561, Syntagma	1
3	987654321	Nikos Georgiou	1990-09-22	Kifisia 13, 11523, Ampelokipoi	2
4	112233445	Elena Dimitriou	1978-01-13	Patision 173, 11251, Agios Loukas	3
5	223344556	Kostas Vassilakis	1978-01-13	Evelpidon 38, 11362, Kypseli	4
6	334455667	Yiannis Papadakis	1988-05-20	Vasilissis Sofias 40, 10676, Kolonaki	5

CUSTOMER EMAIL

<u>CustomerID</u>	<u>Email</u>
1	mariapap@gmail.com
1	mariapap@hotmail.com
2	mpap@hotmail.com
3	nikosg@gmail.com
4	elenad@outlook.com
5	kostasv@hotmail.com
6	yiannis.papadakis@gmail.com

CUSTOMER TEL

<u>CustomerID</u>	<u>Tel</u>
1	6980000011
2	6945567890
2	2310934567
3	6931122334
4	6988991122
5	6977889900
6	6975919900

BANK BRANCH

<u>BranchID</u>	<u>OperatingHours</u>	<u>Address</u>
1	07:00-14:00	Peiraios 110, 11854, Gazi
2	07:00-14:00	Sofokleous 10, 10552, Omonoia
3	08:00-15:00	Aristotelous 62, 10433, Plateia Vathis
4	08:00-15:00	3is Septemvriou 100, 10434, Plateia Attikis
5	08:00-15:00	Ippokratous 15, 10679, Kolonaki

BANK BRANCH EMAIL

<u>BranchID</u>	<u>Email</u>
1	central@bank.gr
2	second@bank.gr
3	third@bank.gr
4	fourth@bank.gr
4	fourthB@bank.gr
5	fifth@bank.gr

BANK BRANCH TEL

<u>BranchID</u>	<u>Tel</u>
1	2101234567
2	2310123456
3	2107654321
4	2316111054
4	2316789054
5	2103345678

EMPLOYEE

<u>EmployeeID</u>	<u>TIN</u>	<u>Name</u>	<u>DoB</u>	<u>Address</u>	<u>BranchID</u>
1	111155667	Giorgos Kotsis	1978-04-12	Kapodistriou 5, 10682, Exarcheia	1
2	222255667	Eleni Markou	1988-11-30	Deligianni 33, 10438, Proastia	1
3	333355667	Nikos Stavropoulos	1990-02-01	Alexandras 17, 11473, Neapoli	2
4	444455667	Maria Korinthiou	1985-06-15	Syngrou 14, 11742, Koukaki	3
5	555555667	Kostas Nikandros	1982-09-18	Syngrou 4, 11742, Koukaki	4
6	555555999	Kostas Nikopoulos	1992-10-08	Syngrou 1, 11742, Koukaki	5

EMPLOYEE EMAIL

<u>EmployeeID</u>	<u>Email</u>
1	gkotsis@gmail.gr
1	gkotsisG@gmail.gr
2	emarkou@gmail.gr
3	nstavropoulos@gmail.gr
4	mkorinth@gmail.gr
5	knikan@gmail.gr
6	knikopoulos@gmail.gr

EMPLOYEE TEL

<u>EmployeeID</u>	<u>Tel</u>
1	2105678901
1	6989744444
2	2310789456
3	2102345678
4	2314567890
5	2108765432
6	2310934777

ACCOUNT

AccountID	AccountNumber	OpeningDate	Currency	Status	CustomerID
1	GR1634441234567890123456789	2020-03-02	EUR	Active	1
2	GR1311891434567890123456789	2021-10-02	EUR	Active	1
3	GR1223459876543210123456789	2021-07-10	EUR	Active	2
4	GR1034562345678901234567899	2022-01-15	EUR	Active	3
5	GR1745673456789012345678900	2023-05-12	USD	Active	3
6	GR1356784567890123456789010	2023-08-30	GBP	Active	3
7	GR1167895678901234567890121	2023-10-01	EUR	Active	4
8	GR1478906789012345678901231	2023-11-05	USD	Active	5
9	GR1589017890123456789012342	2023-12-15	GBP	Active	5
10	GR1990128901234567890123453	2024-01-20	EUR	Active	6
11	GR2001239012345678901234564	2024-02-10	USD	Closed	6

SAVINGS ACCOUNT

<u>AccountID</u>	<u>InterestRate</u>
1	1.20
2	0.90
3	1.50
4	1.10
5	0.85

CHECKING ACCOUNT

<u>AccountID</u>	OverdraftLimit
6	800.00
7	600.00
8	1200.00
9	950.00
10	1100.00
11	1100.00

TRANSACTION

<u>TransactionID</u>	Date	Time	Amount
1	2023-02-15	11:30:00	-150.00
2	2023-03-01	10:33:10	50000.00
3	2023-05-12	12:20:11	-500.00
4	2023-06-20	10:32:01	1200.00
5	2023-06-20	10:32:01	-1200.00
6	2023-08-01	09:12:02	-300.00
7	2024-08-01	15:38:03	-222.00
8	2023-12-01	14:39:04	-100.00
9	2020-03-11	15:37:05	-302.00
10	2019-08-21	16:37:06	-120.00
11	2017-08-01	17:36:07	-600.00
12	2018-08-21	18:35:08	-90.00
13	2018-10-01	19:34:09	-650.00
14	2022-11-29	20:30:19	-330.00
15	2021-12-30	21:20:24	-145.00
16	2018-01-15	22:10:00	-7340.00

ACCOUNT TRANSACTION

<u>TransactionID</u>	<u>AccountID</u>	MovementType
1	3	Withdrawal
2	4	Deposit
3	3	LoanPayment
4	3	Transfer_IN
5	4	Transfer_OUT
6	4	LoanPayment

CARD TRANSACTION

TransactionID	TerminalNumber	CardID
7	1	3
8	2	3
9	3	4
10	3	5
11	3	6

CREDIT PAYMENT TRANSACTION

TransactionID	AccountID	CardID
12	1	1
13	3	2
14	3	2
15	7	4
16	7	5

CARD

CardID	CardNumber	CardholderName	CVV	ExpirationDate	Status
1	1234567890123456	Maria Papadopoulou	123	2027-05-01	Active
2	2345678901234567	Maria Papadopoulou	482	2026-11-01	Frozen
3	3456789012345678	Maria Papadopoulou	301	2028-09-15	Active
4	4567890123456789	Elena Dimitriou	559	2027-06-23	Active
5	5678901234567890	Elena Dimitriou	741	2029-02-18	Active
6	6789012345678901	Maria Papadopoulou	102	2028-03-14	Active
7	7890123456789012	Nikos Georgiou	305	2029-07-12	Active
8	8901234567890123	Nikos Georgiou	420	2027-01-29	Active
9	9012345678901234	Elena Dimitriou	591	2029-11-10	Frozen
10	0123456789012345	Elena Dimitriou	678	2028-06-25	Active

DEBIT CARD

CardID	DailyTransactionLimit	AccountID
6	500.00	3
7	400.00	4
8	500.00	6
9	600.00	7
10	1000.00	7

CREDIT CARD

<u>CardID</u>	<u>CreditLimit</u>	<u>CustomerID</u>
1	5000.00	1
2	6000.00	2
3	7000.00	2
4	8000.00	4
5	9000.00	4

served by

<u>CustomerID</u>	<u>BranchID</u>
1	2
1	3
1	4
1	5
2	2
2	3
2	4
2	5
3	2
3	3
3	4
3	5
4	2
4	3
4	4
4	5
5	2
5	3
5	4
5	5
6	2
6	3
6	4
6	5

Εκτίμηση πλήθους εγγραφών

- CUSTOMER: 200.000

- ACCOUNT: 360.000
- LOAN: 40.000
- TRANSACTION: 86.400.000
- CARD: 360.000
- BRANCH: 120
- EMPLOYEE: 2000

7.2 Παραδείγματα Ερωτημάτων

Στην ενότητα αυτή δίνονται ενδεικτικά κάποια χρήσιμα ερωτήματα πάνω στη βάση δεδομένων της τράπεζάς μας.

Ερώτημα 1: Πελάτες με ενεργά δάνεια

Το ερώτημα αυτό αποσκοπεί στην επιλογή όλων των δανείων που είναι ακόμα σε ισχύ, δηλαδή των δανείων που δεν έχουν εξοφληθεί ακόμα.

Η σχεσιακή άλγεβρα που εκφράζει το ερώτημα είναι η εξής:

$$\sigma_{Debt>0}(\text{LOAN DEBTS})$$

Εδώ, χρησιμοποιούμε την πράξη της επιλογής (σ) για να φιλτράρουμε τις εγγραφές που αφορούν δάνεια τα οποία έχουν χρέος (Debt) διάφορο του μηδενός. Για τον σκοπό αυτόν κάνουμε χρήση της όψης/πίνακα LOAN DEBTS που κατασκευάσαμε σε προηγούμενη ενότητα και ο οποίος περιέχει τα πεδία CustomerID, Name, TIN, LoanID, ExpiratonDate, Type, Amount, Debt.

Ερώτημα 2: Ενεργές χρεωστικές κάρτες πελάτη

Το ερώτημα αυτό αποσκοπεί στην επιλογή όλων των ενεργών χρεωστικών καρτών όλων των πελατών.

Η σχεσιακή άλγεβρα που εκφράζει το ερώτημα είναι η εξής:

$$\left(\begin{array}{c} \Pi_{CardID, CardNumber, CardholderName, CVV, ExpirationDate}(\sigma_{Status=Active}(\text{CARD})) \bowtie \text{DEBIT CARD} \\ \bowtie \\ \text{CUSTOMER ACCOUNTS} \end{array} \right) = B$$

και

$$\Pi_{CardID, CardNumber, CardholderName, CVV, ExpirationDate, CustomerID, TIN, Name}(B)$$

Αρχικά, χρησιμοποιούμε την πράξη της επιλογής (σ) για να φιλτράρουμε τη σχέση CARD και να βρούμε τις ενεργές (Active) κάρτες (πιστωτικές και μη). Στη συνέχεια, με την πράξη της προβολής (Π), επιλέγουμε τα πεδία CardID, CardNumber, CardholderName και ExpirationDate από τα αποτελέσματα αυτής της επιλογής.

Ακολουθεί η συνένωση, Join (\bowtie), με την DEBIT CARD, ώστε να πάρουμε όλες τις χρεωστικές κάρτες που είναι ενεργές.

Τέλος, ακολουθεί η συνένωση, Join (\bowtie), των παραπάνω με την όψη/πίνακα CUSTOMER ACCOUNTS που δημιουργήσαμε σε προηγούμενη ενότητα και τέλος μέσω μιας ακόμη προβολής επιλέγουμε τα πεδία που επιθυμούμε να κρατήσουμε. Έτσι, το τελικό αποτέλεσμα θα περιλαμβάνει όλες τις ενεργές χρεωστικές κάρτες που σχετίζονται με κάθε πελάτη και λογαριασμό, μαζί με αντίστοιχα στοιχεία, όπως για παράδειγμα το CardNumber, την ημερομηνία λήξης κ.α. Ας μην ξεχνάμε πως κατά την υπόθεσή μας, μια χρεωστική κάρτα συνδέεται με ακριβώς ένα λογαριασμό.

Ερώτημα 3: Τι πάλληλοι που εργάζονται σε υποκαταστήματα με συγκεκριμένα IDs

Το ερώτημα αυτό αποσκοπεί στην επιλογή όλων των υπαλλήλων που εργάζονται στο υποκατάστημα με ID=1 ή 2.

Η σχεσιακή áλγεβρα που εκφράζει το ερώτημα είναι η εξής:

$$\sigma_{BranchID=1 \cup BranchID=2}(\text{EMPLOYEE BRANCH SUMMARY})$$

Χρησιμοποιούμε την πράξη της επιλογής (σ) για να φιλτράρουμε την όψη/πίνακα BANK BRANCH SUMMARY που δημιουργήσαμε σε προηγούμενη ενότητα η οποία περιλαμβάνει τα πεδία Name, TIN, EmpAddress, Tel, Email, BranchID και BranchAddress και να βρούμε τα υποκαταστήματα με BranchID ίσο με 1 ή 2. Τέλος, εμφανίζουμε τις σχετικές πληροφορίες για τους υπαλλήλους αυτούς, όπως είναι αποθηκευμένες στο BANK BRANCH SUMMARY.

Ερώτημα 4: Πελάτες με ενεργό στεγαστικό δάνειο.

Το ερώτημα αυτό αποσκοπεί στην εύρεση όλων των πελατών που έχουν ενεργό στεγαστικό δάνειο.

Η σχεσιακή áλγεβρα που εκφράζει το ερώτημα είναι η εξής:

$$\Pi_{LoanID, Type}(\sigma_{Debt > 0 \cap Type = Housing}(\text{LOAN DEBTS}))$$

▷◁

$$\Pi_{CustomerID, TIN, Name, Address, LoanID}(\text{CUSTOMER})$$

Αρχικά, χρησιμοποιούμε την πράξη της επιλογής (σ) για να φιλτράρουμε την όψη/πίνακα LOAN DEBTS με δύο κριτήρια:

- Ενεργό Δάνειο: Debt > 0 (χρέος διάφορο του μηδενός)
- Τύπος: Type = Housing (στεγαστικό δάνειο).

Στη συνέχεια, εφαρμόζουμε την πράξη της τομής (Π) στις δύο επιλογές. Αυτό εξασφαλίζει ότι επιλέγονται πεδία που ικανοποιούν και τα δύο κριτήρια (ενεργό δάνειο και στεγαστικό δάνειο).

Έπειτα, με την πράξη προβολής (ΠΙ), επιλέγουμε τα πεδία LoanID, Type από το αποτέλεσμα της τομής, ώστε να έχουμε τα αναγνωριστικά των δανείων που πληρούν τις προϋποθέσεις.

Τέλος, συνδυάζουμε τα παραπάνω χρησιμοποιώντας την πράξη του Join (\bowtie), αφού πρώτα φιλτράρουμε αντίστοιχα πεδία ενδιαφέροντος για την σχέση CUSTOMER. Έτσι, το τελικό αποτέλεσμα θα περιλαμβάνει όλους τους πελάτες που έχουν ενεργό στεγαστικό δάνειο, μαζί με τα αντίστοιχα στοιχεία τους.

Ερώτημα 5: Ενεργά δάνεια με χρέος μεγαλύτερο του 500, αλλά όχι μεγαλύτερο του 5.000.

Το ερώτημα αυτό αποσκοπεί στην εύρεση όλων των ενεργών δανείων με χρέος μεγαλύτερο του 500, αλλά όχι μεγαλύτερο του 5.000.

Η σχεσιακή άλγεβρα που εκφράζει το ερώτημα είναι η εξής:

$$\Pi_{LoanID, ExpirationDate, Type, Amount, Debt} (\sigma_{Debt > 500}(\text{LOAN DEBTS})) - \Pi_{LoanID, ExpirationDate, Type, Amount, Debt} (\sigma_{Debt > 5.000}(\text{LOAN DEBTS}))$$

Η έκφραση χρησιμοποιεί την πράξη Διαφοράς (–) για τον περιορισμό εύρους στο πεδίο Debt της όψης/πίνακα LOAN DEBTS που αναφέραμε και πιο πάνω.

- Αρχική Επιλογή: Επιλέγονται τα LoanID, ExpirationDate, Type, Amount, Debt όπου το χρέος είναι μεγαλύτερο των 500 ($\sigma_{Debt > 500}(\text{LOAN DEBTS})$).
- Σύνολο Αφαίρεσης: Επιλέγονται τα LoanID, ExpirationDate, Type, Amount, Debt όπου το χρέος είναι μεγαλύτερο των 5.000 ($\sigma_{Debt > 5.000}(\text{LOAN DEBTS})$).
- Τελικό Αποτέλεσμα: Από το αρχικό σύνολο A , αφαιρούνται τα δάνεια που περιλαμβάνονται στο σύνολο B . Έτσι, διατηρούνται μόνο εκείνα που ικανοποιούν τον περιορισμό: $500 < \text{Debt} \leq 5.000$.

Ερώτημα 6: Άθροισμα πληρωμών πιστωτικής κάρτας από λογαριασμό.

Το ερώτημα αυτό αποσκοπεί στην εύρεση του αθροίσματος των πληρωμών πιστωτικής κάρτας που έχουν γίνει από κάθε λογαριασμό.

Η σχεσιακή άλγεβρα που εκφράζει το ερώτημα είναι η εξής:

$$\text{AccountID} \mathcal{G}_{\text{sum}}(\text{Amount}) \text{ as creditTransSum} \left(\begin{array}{c} \text{CREDIT PAYMENT TRANSACTION} \bowtie \text{TRANSACTION} \\ \bowtie \\ \text{ACCOUNT} \end{array} \right) = A$$

ενώ, εκτελόντας τη συνένωση (join) της παραπάνω σχέσης με την όψη/πίνακα CUSTOMER ACCOUNTS τελικά προκύπτει το ζητούμενο ερώτημα ως:

$$\text{CUSTOMER ACCOUNTS} \bowtie A$$

Η έκφραση χρησιμοποιεί την πράξη Συνάθροισης (\mathcal{G}) για τον υπολογισμό του αθροίσματος.

- Συνένωση Οντοτήτων: Αρχικά, συνενώνουμε τις σχέσεις CREDIT PAYMENT TRANSACTION και TRANSACTION με βάση το TransactionID.
- Προσθήκη Πελατών: Το αποτέλεσμα της πρώτης συνένωσης ενώνεται στη συνέχεια με τη σχέση ACCOUNT μέσω του AccountID. Με αυτόν τον τρόπο, έχουμε μια ενιαία σχέση που περιέχει τα στοιχεία όλων των πληρωμών πιστωτικής καθώς και των λογαριασμών από τους οποίους έγιναν.
- Ομαδοποίηση και Άθροιση: Τέλος, εφαρμόζεται η πράξη της Συνάθροισης (\mathcal{G}). Η ομαδοποίηση γίνεται με βάση το AccountID, και για κάθε μοναδικό λογαριασμό, υπολογίζεται το άθροισμα (SUM) του Amount όλων των πληρωμών πιστωτικής του.

Το τελικό αποτέλεσμα είναι μια λίστα με όλα τα πεδία της όψης CUSTOMER ACCOUNTS και το συνολικό ποσό πληρωμών δόσεων πιστωτικής κάρτας που έγιναν, το οποίο έχει πλέον μετονομαστεί σε *creditTransSum*. Έτσι, λοιπόν, θα περιέχει τα πεδία CustomerID, TIN, Name, AccountID, AccountNumber, Currency και creditTransSum.

Αναφορές

- [1] Υλικό και Διαλέξεις Μαθήματος Βάσεις Δεδομένων, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης (ΑΠΘ), Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.