

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ



ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Εξαμηνιαία εργασία
(Project)

Γαλανάκη Σοφία (03115060)
Διαμάντη Ιωάννα (03115035)
Λεμπέση Παρασκευή (03115149)

30 Μαΐου 2018

Εισαγωγή

“Πέντε από τις κορυφαίες αλυσίδες ξενοδοχείων, με μονάδες σε 14 διαφορετικά μέρη στην Ελλάδα, αποφάσισαν να συνεργαστούν και να αναπτύξουν μία εφαρμογή που να επιτρέπει σε πελάτες να κάνουν εύκολα και αποδοτικά κρατήσεις σε δωμάτια βλέποντας την διαθεσιμότητα σε πραγματικό χρόνο και επιτρέποντας στις μονάδες να έχουν μεγαλύτερα ποσοστά πληρότητας όλο τον χρόνο.”

Για το σκοπό αυτό δημιουργήσαμε μία εφαρμογή η οποία απευθύνεται τόσο στους εν δυνάμει πελάτες των ξενοδοχείων όσο και στο προσωπικό που εργάζεται σε αυτά.

Οι εν δυνάμει πελάτες έχουν τη δυνατότητα να αναζητήσουν δωμάτια στα ξενοδοχεία των αλυσίδων που μετέχουν στη συνεργασία βάσει πολλαπλών κριτηρίων όπως είναι οι ημερομηνίες, οι αλυσίδα των ξενοδοχείων, η χωρητικότητα, οι παροχές, η θέα κ.α. Μπορούν επίσης να κάνουν εγγραφή (sign up) και να προχωρήσουν σε κράτηση ενός ή περισσότερων δωματίων.

Τα μέλη του προσωπικού των ξενοδοχείων από την άλλη μπορούν να κάνουν check-in και check-out τις κρατήσεις που γίνονται στα ξενοδοχεία όπου εργάζονται, να προσλάβουν, να απολύσουν, να επεξεργαστούν έναν υπάλληλο ή να ανανεώσουν το συμβόλαιό του. Ακόμα, οι υπάλληλοι των ξενοδοχείων μπορούν να προσθαφαιρέσουν ξενοδοχεία ή δωμάτια, καθώς και να επεξεργαστούν τα ήδη υπάρχοντα. Τέλος έχουν τη δυνατότητα να προβάλλουν και να επεξεργάζονται τα στοιχεία των πελατών, καθώς επίσης και να προβάλλουν πληροφορίες σχετικά με τα υπάρχοντα δωμάτια και ξενοδοχεία, τις υπάρχουσες αλυσίδες, τις κρατήσεις και την κατάσταση στην οποία αυτές βρίσκονται, τις χρηματικές δοσοληψίες που έχουν πραγματοποιηθεί μεταξύ των ξενοδοχείων και των πελατών, αλλά και τους υπαλλήλους σε συνδυασμό με τις θέσεις στις οποίες αυτοί απασχολούνται.

Παράλληλα, το user interface είναι σχεδιασμένο έτσι ώστε να αποτρέπει σε μεγάλο βαθμό λάθη των χρηστών καθώς στο περισσότερο τμήμα του γίνεται χρήση drop down λιστών και radio buttons. Η ιστοσελίδα της εφαρμογής ενημερώνει το χρήστη σε περίπτωση λάθους με τη χρήση μηνυμάτων κατάλληλων για την εκάστοτε περίπτωση.

Ερώτημα 1ο - Σχεσιακό Διάγραμμα της Βάσης Δεδομένων

Με βάση το ER διάγραμμα που μας δόθηκε καθώς και ορισμένους περιορισμούς που ζητήθηκαν στην εκφώνηση (όπως για παράδειγμα να διατηρείται αρχείο κρατήσεων και ενοικιάσεων) αναπτύχθηκε το ακόλουθο σχεσιακό μοντέλο:

Με έντονα και υπογραμμισμένα γράμματα δηλώνουμε τα κλειδιά.

Αρχικά, μετατρέπουμε κάθε ισχυρό σύνολο οντοτήτων σε σχέση:

- employee (**IRS Number**, Last_Name, First_Name, Social_Security_Number, Street, Number, Postal_Code, City)
- hotel (**hotel ID**, hotel_name, Street, Number, Postal_Code, City, Stars, Number_of_rooms, Hotelgroup_ID)

Σημείωση:

Το χαρακτηριστικό Hotelgroup_ID οφείλεται στο γεγονός ότι η has_hotels είναι N:1

- Customer (**IRS Number**, Street, Number, Postal_Code, City, First_name, Last_name, Social_Security_Number, First_Registration)
- hotel_group (**Hotel group ID**, hotelgroup_name, Street, Number, Postal_Code, City_group, Number_of_hotels)

Για τους αδύναμους τύπους οντοτήτων έχουμε τα εξής:

Γενικά, για τους αδύναμους τύπους οντοτήτων χρησιμοποιούμε ως primary key το δικό τους σε συνδυασμό με τα primary keys του (των) πίνακα (πινάκων) από όπου εξαρτώνται.

- hotel_room (**Room id**, **Hotel ID**, Room_Number, Capacity, View, Expandable, Repairs_need, Price)

Σημείωση:

Το χαρακτηριστικό Room_Number προστέθηκε με σκοπό να υπάρχει δυνατότητα στο user interface να προβάλλεται -όποτε χρειάζεται- ένα "όνομα" για το δωμάτιο καθώς ο χρήστης δεν πρέπει να γνωρίζει λεπτομέρειες της βάσης όπως είναι το Room_id.

- payment_transaction (**IRS Number Employee**, **IRS Number Customer**, **Room ID**, **Hotel ID**, Payment_Amount, Payment_Method)

Για τις συσχετίσεις έχουμε τα ακόλουθα:

- reserves (**IRS Number Customer**, **Room ID**, **Hotel ID**, **Start Date**, **Finish Date**, Hotel_name, Hotelgroup_name)

Σημείωση:

Τα χαρακτηριστικά Start_Date και Finish_Date έγιναν κλειδιά ώστε να είμαστε σε θέση να

προσδιορίζουμε μοναδικά τις κρατήσεις, διότι πολλοί πελάτες μπορεί να έχουν κάνει κράτηση για πολλά δωμάτια και συγχρόνως διαφορετικοί πελάτες να έχουν κάνει κράτηση για ίδια δωμάτια.

Ακόμη, τα προστέθηκαν τα χαρακτηριστικά *Hotel_name* και *Hotelgroup_name* ώστε σε περίπτωση που ο χρήστης (που ανήκει στο προσωπικό) θελήσει να δει το ιστορικό των κρατήσεων να μη βλέπει τα *ids* αλλά τα ονόματα. Να σημειωθεί ότι η ανάκτηση αυτών δεν είναι πάντα δυνατή από τα αντίστοιχα *ids* καθώς αυτά μπορεί να διαγραφούν νωρίτερα από τη βάση.

- rents (**IRS Number Employee**, **IRS Number Customer**, **Room id**, **Hotel id**, **Start Date**, **Finish Date**, *Hotel_name*, *Hotelgroup_name*)

Σημείωση:

Ομοίως με τη *reserves*.

- works (**IRS Number Employee**, **Hotel ID**, **Start Date**, **Position**, *Finish_Date*)

Σημείωση:

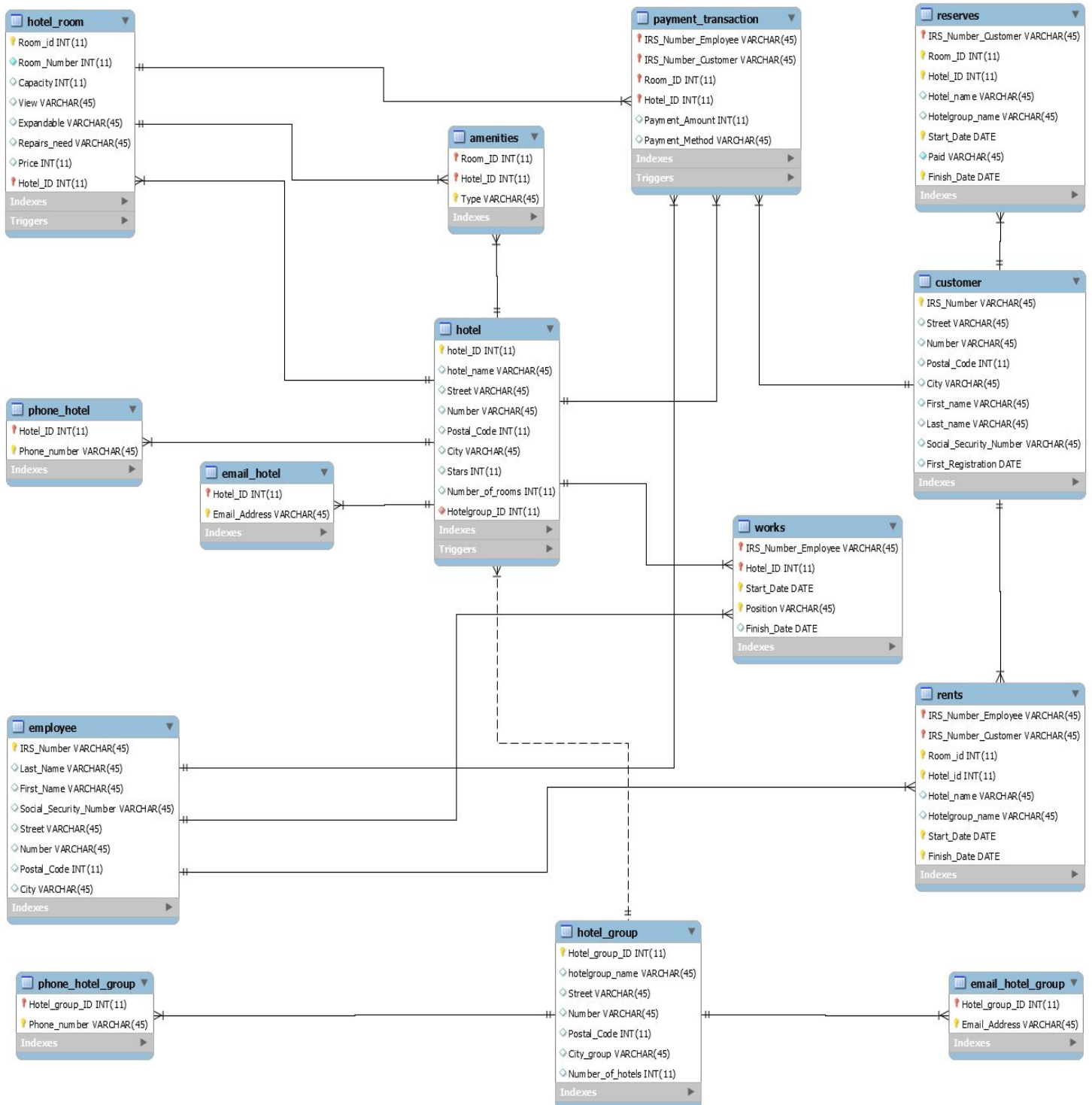
Ο ίδιος υπάλληλος μπορεί να δουλεύει ταυτόχρονα σε περισσότερες από 1 θέσεις στο ίδιο ξενοδοχείο, αλλά και να επαναπροσληφθεί στην ίδια θέση κάποια επόμενη χρονική στιγμή. Για αυτό συμπεριλαμβάνουμε στο *Primary key* και τα *Start_Date*, *Position*.

Για τα πλειότιμα χαρακτηριστικά ισχύουν τα εξής:

- email_hotel (**Hotel ID**, **Email Address**)
- phone_hotel (**Hotel ID**, **Phone number**)
- email_hotel_group (**Hotel group ID**, **Phone number**)
- phone_hotel_group (**Hotel group ID**, **Email Address**)
- amenities (**Room ID**, **Hotel ID**, **Type**)

Λαμβάνοντας υπόψη τα παραπάνω καταλήγουμε στο ακόλουθο σχεσιακό διάγραμμα που φαίνεται στην επόμενη σελίδα (βλ. και υπόμνημα):

- ! Key: (Part of) Primary Key
- ◆ Filled Diamond: NOT NULL
- ◇ Not filled Diamond: NULL
- !◆ Red colored: (Part of) Foreign key
- ◆ Blue lined Diamond: Simple attribute (no key)



Ερώτημα 2ο - Περιορισμοί που εξασφαλίζουν την ορθότητα της Βάσης Δεδομένων

α) Περιορισμοί Ακεραιότητας

- Περιορισμός “not null” :

Επιλέξαμε να χρησιμοποιήσουμε το συγκεκριμένο περιορισμό ακεραιότητας σε κάθε field το οποίο είναι primary key καθώς εξ' ορισμού τα fields αυτού του τύπου δεν επιτρέπεται να παίρνουν τιμές null. Επίσης, ως not null επιλέξαμε και τα fields των tables τα οποία αποτελούν foreign key με αναφορά σε κάποιο primary key ενός άλλου table.

- Περιορισμός “unique”:

Επιλέξαμε να χρησιμοποιήσουμε τον συγκεκριμένο περιορισμό στο field “Hotel Name” του table hotel καθώς, δεδομένου ότι ο χρήστης της εφαρμογής δεν γνωρίζει τα ID των ξενοδοχείων (τα οποία είναι μοναδικά), επιλέγει κάθε φορά το ξενοδοχείο το οποίο θέλει να επεξεργαστεί ή να διαγράψει με βάση το όνομά του. Παρατηρήσαμε λοιπόν πως χωρίς αυτόν τον περιορισμό δημιουργούνταν διάφορα προβλήματα, διότι στην περίπτωση ύπαρξης δύο ξενοδοχείων με το ίδιο όνομα υπάρχει σύγχυση σχετικά με το ποιο είναι τελικά εκείνο που ο χρήστης θέλει να τροποποιήσει ή να διαγράψει.

Επίσης, για τον ίδιο λόγο, ο περιορισμός αυτός χρησιμοποιήθηκε στο σύνολο “(Room Number, Hotel ID)” του table hotel_room, το οποίο χρησιμοποιείται με τον ίδιο τρόπο όπως το Hotel Name. Είναι δηλαδή το “όνομα” του δωματίου, το οποίο ο χρήστης γνωρίζει και με βάση αυτό επιλέγει το δωμάτιο που θα επεξεργαστεί ή θα διαγράψει.

β) Κλειδιά

- Primary Keys:

Τα fields που επιλέξαμε να χρησιμοποιήσουμε ως Primary Key σε κάθε table είναι τα εξής:

→ employee: (irs_number) - είναι κωδικός διαφορετικός για κάθε πολίτη οπότε είναι και αυτό που διαφοροποιεί τις καταχωρήσεις σε περίπτωση πχ συνωνυμίας

→ customer: (irs_number) - ομοίως

→ hotel_group: (hotel_group_number) - είναι κωδικός διαφορετικός για κάθε hotel group ο οποίος και πάλι διαφοροποιεί τις καταχωρήσεις μεταξύ τους σε περίπτωση συνωνυμίας

→ hotel: (hotel_id) - ομοίως με παραπάνω

→ hotel_room: (room_id, hotel_id) - εδώ χρησιμοποιούμε πέρα από το room_id και το hotel_id διότι το hotel_room είναι weak entity και έτσι απαιτείται και το Primary key του entity από το οποίο εξαρτάται.

→ email_hotel: (hotel_id, email_address) - χρησιμοποιούμε αυτά τα δύο καθώς ενδέχεται το ίδιο ξενοδοχείο να έχει περισσότερα από ένα email.

→ phone_hotel: (hotel_id, phone_number) - ομοίως

→ email_hotel_group: (hotel_group_id, email_address) - ομοίως

→ phone_hotel_group: (hotel_group_id, phone_number) - ομοίως

→ amenities: (room_id, hotel_id, type) - χρησιμοποιούμε και τα τρία αυτά field ως primary key διότι το hotel_room είναι weak entity οπότε το room_id θα πρέπει να συνοδεύεται από το hotel_id. Επίσης, ένα δωμάτιο μπορεί να διαθέτει περισσότερα του ενός amenities επομένως και το type θα πρέπει να συμπεριληφθεί στο primary key.

→ payment_transaction: (irs_number_employee, irs_number_customer, room_id, hotel_id) - εδώ και πάλι λόγω weak entity το room_id θα πρέπει να συνοδεύεται από το hotel_id. Επίσης, ως weak entity, το payment_transaction πρέπει να έχει ως primary keys όλα τα primary keys των οντοτήτων από τα

οποία εξαρτάται. Για το λόγο αυτό συμπεριλαμβάνουμε και τα `irs_number_employee`, `irs_number_customer`.

→ `works`: (`hotel_id`, `irs_number_employee`, `start_date`, `position`) - το table αυτό περιγράφει μία σχέση μεταξύ των entities `hotel` και `employee`. Επομένως, πρέπει στο primary key αυτού του table να συμπεριλαμβάνονται και τα primary keys των entities τα οποία συνδέει, δηλαδή τα (`hotel_id`, `irs_number_employee`). Επίσης, καθώς αναφέρεται στην εκφώνηση πως ένας υπάλληλος μπορεί να δουλεύει ταυτόχρονα σε περισσότερες από 1 θέσεις στο ίδιο ξενοδοχείο, αλλά και να επαναπροσληφθεί στην ίδια θέση για διαφορετική χρονική περίοδο, συμπεριλαμβάνουμε στο Primary key και τα `start_date`, `position`.

→ `reserves`: (`irs_number_customer`, `room_id`, `hotel_id`, `start_date`, `finish_date`) - το table αυτό περιγράφει μία σχέση μεταξύ των entities `hotel_room` και `customer`. Επομένως, πρέπει στο primary key αυτού του table να συμπεριλαμβάνονται και τα primary keys των entities τα οποία συνδέει, δηλαδή τα (`room_id`, `irs_number_customer`). Όπως και προηγουμένως, λόγω weak entity το `room_id` θα πρέπει να συνοδεύεται από το `hotel_id`. Επίσης, είναι δυνατόν να γίνει κράτηση στο ίδιο δωμάτιο από τον ίδιο πελάτη αλλά για διαφορετικές ημερομηνίες. Επομένως, προσθέτουμε στο primary key και τα `start_date`, `finish_date`.

→ `rents`: (`irs_number_customer`, `irs_number_employee`, `room_id`, `hotel_id`, `start_date`, `finish_date`) - το table αυτό περιγράφει μία σχέση μεταξύ των entities `hotel_room`, `employee` και `customer`. Επομένως, πρέπει στο primary key αυτού του table να συμπεριλαμβάνονται και τα primary keys των entities τα οποία συνδέει, δηλαδή τα (`room_id`, `irs_number_employee`, `irs_number_customer`). Όπως και προηγουμένως, λόγω weak entity το `room_id` θα πρέπει να συνοδεύεται από το `hotel_id`. Επίσης, είναι δυνατόν να γίνει ενοικίαση στο ίδιο δωμάτιο από τον ίδιο πελάτη και τον ίδιο υπάλληλο αλλά για διαφορετικές ημερομηνίες. Επομένως, προσθέτουμε στο primary key και τα `start_date`, `finish_date`.

γ) Αναφορική ακεραιότητα

- Foreign Keys:

→ Στο field `hotel_group_id` του table `hotel` με αναφορά στο table `hotel_group` (on update cascade, on delete cascade): Δεν θα υπάρξει ποτέ στη βάση κάποιο ξενοδοχείο το οποίο να ανήκει σε κάποιο `hotel_group` που έχει διαγραφεί.

→ Στο field `hotel_id` του table `hotel_room` με αναφορά στο table `hotel` (on update cascade, on delete cascade): Δεν θα υπάρξει ποτέ στη βάση κανένα δωμάτιο το οποίο να ανήκει σε ξενοδοχείο το οποίο έχει διαγραφεί.

→ Στο field `hotel_id` των tables `email_hotel` και `phone_hotel` με αναφορά στο table `hotel` (on update cascade, on delete cascade):

Με τη διαγραφή κάποιου ξενοδοχείου διαγράφονται αυτόματα όλα τα αντίστοιχα τηλέφωνα και email.

→ Στο field `hotel_group_id` των tables `email_hotel_group` και `phone_hotel_group` με αναφορά στο table `hotel_group` (on update cascade, on delete cascade): Με τη διαγραφή κάποιου `hotel_group` διαγράφονται αυτόματα όλα τα αντίστοιχα τηλέφωνα και email.

→ Στα fields `IRS Number Employee` (on update cascade, on delete no action) και `Hotel ID` (on update cascade, on delete cascade) του table της σχέσης `works` με αναφορά στα αντίστοιχα fields των tables `employee` και `hotel`: Αν τροποποιηθεί το `IRS` κάποιου υπαλλήλου, αυτόματα τροποποιείται και το `IRS` του υπαλλήλου αυτού μέσα στο table της `works`. Ακόμα, με τη χρήση του `Hotel ID` ως foreign key εξασφαλίζουμε ότι δεν θα υπάρχουν μέσα στον πίνακα `works` καταχωρήσεις που θα σχετίζονται με ξενοδοχεία που έχουν διαγραφεί

→ Στα fields Hotel ID και Room ID του table amenities με αναφορά στα tables hotel και hotel_room αντίστοιχα (on update cascade, on delete cascade): Εξασφαλίζουμε ότι με τη διαγραφή κάποιου δωματίου το οποίο ανήκει σε κάποιο συγκεκριμένο ξενοδοχείο θα διαγραφούν αυτόματα και όλες οι αντίστοιχες καταχωρήσεις στον πίνακα amenities.

→ Στα fields Room ID, Hotel ID, (on update cascade, on delete cascade) και Customer IRS και Employee IRS του table payment_transaction (on update cascade, on delete no action), με αναφορά στα αντίστοιχα fields των tables hotel_room, hotel, customer και employee αντίστοιχα: Έτσι εξασφαλίζουμε πως όλες συναλλαγές που αφορούν κάποιο δωμάτιο το οποίο διαγράφεται διαγράφονται και αυτές. Επιπλέον, με αυτόν τον τρόπο, με κάθε αλλαγή που γίνεται στα fields Customer IRS και Employee IRS των tables customer και employee αντίστοιχα, τα αντίστοιχα fields του table payment_transactions ενημερώνονται αυτόματα.

→ Στο field Customer IRS του table reserves (on update cascade, on delete no action), με αναφορά στο αντίστοιχο field του table, customer: Έτσι με κάθε αλλαγή που γίνεται στο field Customer IRS του table customer, το αντίστοιχο field του table reserves ενημερώνεται αυτόματα.

→ Στο field Customer IRS και Employee IRS του table rents (on update cascade, on delete no action), με αναφορά στα αντίστοιχα fields των tables customer και employee αντίστοιχα: Έτσι με κάθε αλλαγή που γίνεται στο field Customer IRS του table customer, το αντίστοιχο field του table rents ενημερώνεται αυτόματα.

Σημείωση:

Αρχικά, και στις δύο παραπάνω σχέσεις (reserves & rents) είχαμε συμπεριλάβει στα foreign keys και τα fields Room ID και Hotel ID, με αναφορά στα αντίστοιχα fields των tables hotel_room και hotel αντίστοιχα. Στην πορεία όμως, καθώς από την εκφώνηση της άσκησης ζητείται να κρατάμε αρχείο για τις κρατήσεις και τις ενοικιάσεις ακόμα και αν το δωμάτιο που σχετίζεται με αυτές δεν υπάρχει πλέον στη βάση, αλλά και δεδομένου του γεγονότος ότι όταν ένα πεδίο έχει οριστεί ως foreign key σε μία σχέση δεν είναι δυνατόν να παραμείνει ανεπηρέαστο όταν υπάρξει κάποια αλλαγή (update, delete) στο table-γονέα, αποφασίσαμε να αφαιρέσουμε αυτούς τους περιορισμούς.

Επιπλέον, καθώς αρχικά κάθε καταχώρηση των δύο αυτών σχέσεων περιλάμβανε μόνο το room id και το hotel id και όχι κάποια παραπάνω πληροφορία για το ξενοδοχείο (όπως πχ το όνομά του), συνειδητοποιήσαμε πως μετά τη διαγραφή κάποιου δωματίου ή ξενοδοχείου ή group δεν θα μπορούσαμε να βρούμε μία τέτοιου είδους πληροφορία, αφού οι καταχωρήσεις με το συγκεκριμένο ID δεν θα υπήρχαν πλέον στα tables hotel_room, hotel, hotel_group. Παρόλα αυτά, θεωρήσαμε ότι η γνώση σχετικά με το σε ποιο ξενοδοχείο και group αναφέρεται μία κράτηση ή ενοικίαση που διατηρείται στο αρχείο μας είναι μία πληροφορία απαραίτητη. Έτσι καταλήξαμε στο να συμπεριλάβουμε και στις δυο παραπάνω σχέσεις δύο ακόμα πεδία, το hotel_name και το hotel_group_name, ώστε ακόμα και αν διαγραφεί κάποιο ξενοδοχείο ή group να μην χάσουμε την πληροφορία για το σε ποιο ξενοδοχείο και group ήταν η κράτηση ή η ενοικίαση.

Ωστόσο, λαμβάνοντας υπόψη ότι οι επιπλέον πληροφορίες αυτές ενδέχεται να αλλάξουν (δηλαδή πχ να αλλάξει το όνομα κάποιου ξενοδοχείου), δημιουργείται πλέον ένα πρόβλημα καθώς, εφόσον τα πεδία hotel_name, hotel_group_name δεν αποτελούν foreign keys δεν θα ενημερωθούν στην περίπτωση μίας τέτοιας αλλαγής. Για να διορθώσουμε αυτό το πρόβλημα, κατασκευάσαμε ένα trigger το οποίο σε κάθε update που γίνεται στο όνομα του ξενοδοχείου στο table hotel, ενημερώνει και το όνομα του ξενοδοχείου με το ίδιο ID στα tables reserves και rents.

δ) Ακεραιότητα πεδίου τιμών

- Σχεδόν το σύνολο των fields της βάσης που αναπτύχθηκε είναι τύπου VARCHAR (ακόμα και σε πεδία που φαινομενικά είναι αριθμοί). Ο λόγος για αυτή την επιλογή είναι πως σε περιπτώσεις όπως πχ τα Phone, Postal Code, IRS ή SSN Numbers όταν χρησιμοποιήσαμε τον τύπο INT παρατηρήσαμε πως σε περίπτωση ύπαρξης μηδενικών ψηφίων στην αρχή του κωδικού αυτά χάνονταν κατά την μετατροπή σε INT. Εξαίρεση αποτελούν κάποια fields όπως τα εξής:

→ Τα hotel_group_id, hotel_id, hotel_room_id που είναι τύπου INT.
 → Το number_of_hotels και number_of_rooms των hotel_group και hotel αντίστοιχα που είναι τύπου INT.
 → Το field stars του table hotel που είναι τύπου INT.
 → Τα fields capacity και price του table hotel_room που είναι τύπου INT.
 → Το field payment_amount του table payment_transaction που είναι τύπου INT.
 → Τα fields first_registration του table customer, start_date και finish_date των tables works, reserves και rents

ε) Περιορισμοί οριζόμενοι από το χρήστη

- Triggers:

→ Με κάθε εισαγωγή δωματίου το field number_of_rooms του αντίστοιχου ξενοδοχείου αυξάνεται κατά 1.
 → Με κάθε διαγραφή δωματίου το field number_of_rooms του αντίστοιχου ξενοδοχείου μειώνεται κατά 1.
 → Με κάθε εισαγωγή ξενοδοχείου το field number_of_hotels του αντίστοιχου group αυξάνεται κατά 1.
 → Με κάθε διαγραφή ξενοδοχείου το field number_of_hotels του αντίστοιχου group μειώνεται κατά 1.
 → Με κάθε update ξενοδοχείου τα αντίστοιχα (για το ίδιο hotel id) fields hotel_name των tables reserves και rents ενημερώνονται με την νέα τιμή του hotel_name του table hotel.
 → Με κάθε εισαγωγή στον πίνακα payment_transaction για κάποιο συγκεκριμένο room_id το πεδίο paid του πίνακα reserves γίνεται yes για το αντίστοιχο room_id.

Σημείωση:

Μέσα στην Php υλοποιούνται κάποιοι επιπλέον περιορισμοί οι οποίοι εξασφαλίζουν την λογική ορθότητα των δεδομένων που εισάγονται στη βάση και την κάλυψη των απαιτήσεων της εκφώνησης.

Μερικά παραδείγματα αυτών είναι τα παρακάτω:

→ Σε μία κράτηση ή ενοικίαση η τελική ημερομηνία πρέπει να είναι μεγαλύτερη της αρχικής.
 → Κρατήσεις είναι δεν είναι δυνατόν να πραγματοποιούνται για ημερομηνίες που έχουν παρέλθει.
 → Το check-in, δηλαδή η μετατροπή μίας κράτησης σε ενοικίαση μπορεί να πραγματοποιηθεί μόνο κατά την άφιξη του πελάτη στο ξενοδοχείο, δηλαδή την αρχική ημερομηνία της κράτησης. Επιπλέον, ο υπάλληλος ο οποίος αναλαμβάνει το check-in και είναι υπεύθυνος για την ενοικίαση πρέπει να εργάζεται στο ξενοδοχείο όπου η κράτηση αυτή έχει γίνει.
 → Το check-out, δηλαδή η πληρωμή μίας κράτησης και ο έλεγχος για τυχόν ανάγκη επισκευών στο δωμάτιο που είχε ενοικιαστεί, δεν μπορεί να γίνει αργότερα της τελικής ημερομηνίας της κράτησης. Επίσης, είναι και πάλι υποχρεωτικό ο υπάλληλος ο οποίος αναλαμβάνει να κάνει το check-out να εργάζεται στο ξενοδοχείο όπου η κράτηση αυτή έχει γίνει.

κ.α.

Ερώτημα 3ο - Ευρετήρια Βάσης Δεδομένων

Τα ευρετήρια (indexes) χρησιμοποιούνται για την γρήγορη εύρεση σειρών με συγκεκριμένες τιμές μιας στήλης. Ωστόσο, η χρήση τους δεν προσφέρεται σε πίνακες που είναι συχνή η προσθαφαίρεση σειρών διότι έχουμε χρονικό κόστος. Είναι ωφέλιμο να χρησιμοποιούνται σε πίνακες στους οποίους αποθηκεύεται πληθώρα δεδομένων. Επίσης επιταχύνουν την εκτέλεση όταν χρησιμοποιούνται στα where, order by.

Αναφορικά με τη βάση μας εκτός των primary keys που χρησιμοποιήθηκαν (βλ παραπάνω ερωτήματα) που είναι ένας είδος index, δε χρησιμοποιήσαμε κάπου αλλού κάποιο index. Αυτό συνέβη γιατί, σε queries, όπως αυτό με το οποίο προκύπτουν τα διαθέσιμα δωμάτια, χρησιμοποιούσαν -στα join, where, order by- είτε στήλες των primary keys, είτε στήλες οι οποίες ανανεώνονται αρκετά συχνά κατά τη διάρκεια χρήσης της εφαρμογής. Τέτοιο παράδειγμα είναι η στήλη Paid στον πίνακα reserves. Δε θα οφελούσε να κάνουμε αυτή τη στήλη index καθώς η reserves αλλάζει αρκετά συχνά με τις προσθήκες κρατήσεων. Έπειτα queries που χρησιμοποιήθηκαν για την προσθαφαίρεση και την επεξεργασία πινάκων όπως για παράδειγμα στην επεξεργασία ξενοδοχείου και ξενοδοχειακής μονάδας δε βάλαμε indexes καθώς είναι διαδικασίες που δε συμβαίνουν συχνά και συνεπώς η διαφορά στην ταχύτητα αν χρησιμοποιούσαμε index δε θα ήταν αισθητή.

Ερώτημα 4ο - Σύστημα και Γλώσσες Προγραμματισμού που χρησιμοποιήθηκαν

- Για την υλοποίηση της βάσης δεδομένων χρησιμοποιήσαμε την έκδοση 8.0 του MySQL Server.
- Για την επικοινωνία client-server χρησιμοποιήσαμε Apache Web.
- Για το server side τμήμα της εφαρμογής χρησιμοποιήσαμε PHP.
- Για το client side τμήμα της εφαρμογής χρησιμοποιήσαμε κυρίως HTML, αλλά και κάποια στοιχεία javascript (όπως πχ για την υλοποίηση pop-up windows).
- Για τον καθορισμό των αισθητικών στοιχείων της εφαρμογής (όπως πχ για τα χρώματα του background, για τα περιγράμματα, για το styling των κουμπιών κλπ) χρησιμοποιήσαμε CSS.



Ερώτημα 5ο - Οδηγίες για εγκατάσταση της εφαρμογής

Για να εγκαταστήσει κάποιος την εφαρμογή μας στον υπολογιστή του θα πρέπει να διαθέτει MySQL Server και Apache Web με PHP. Αυτά μπορούν να βρεθούν αντίστοιχα στα ακόλουθα links.

- Για τον Apache:
<https://httpd.apache.org/download.cgi#apache24>
- Για τον MySQL Server:
<https://dev.mysql.com/downloads/mysql/>
- Για την PHP:
<http://php.net/downloads.php>

Χρειάζεται ιδιαίτερη προσοχή κατά την εγκατάσταση των ανωτέρω προγραμμάτων ώστε να υπάρχει πλήρης αντιστοιχία με τα αρχεία κώδικα που έχουμε επισυνάψει.

Πιο συγκεκριμένα:

- Κατά την εγκατάσταση του MySQL Server θα χρειαστεί να γίνει η επιλογή που φαίνεται στο σημείο της εγκατάστασης που ακολουθεί.

Authentication Method

☐ Use Strong Password Encryption for Authentication (RECOMMENDED)

MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward.



Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching_sha2_password authentication).

Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.



Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)

Using the old MySQL 5.x legacy authentication method should only be considered in the following cases:

- If applications cannot be updated to use MySQL 8 enabled Connectors and drivers.
- For cases where re-compilation of an existing application is not feasible.
- An updated, language specific connector or driver is not yet available.

Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.

< Back

Next >

Cancel

- Επίσης θα χρειαστεί για την PHP να αντιγράψουμε το αρχείο php.ini-development και να το μετονομάσουμε σε php.ini. Ύστερα, θα πρέπει να κάνουμε uncomment τα ακόλουθα:

```
;extension=mbstring
```

```
;extension=exif ; Must be after mbstring as it depends on it
```

```
extension=mysqli
```

```
;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
```

```
;extension=odbc
```

; Directory in which the loadable extensions (modules) reside.

; <http://php.net/extension-dir>

;extension_dir = "./"

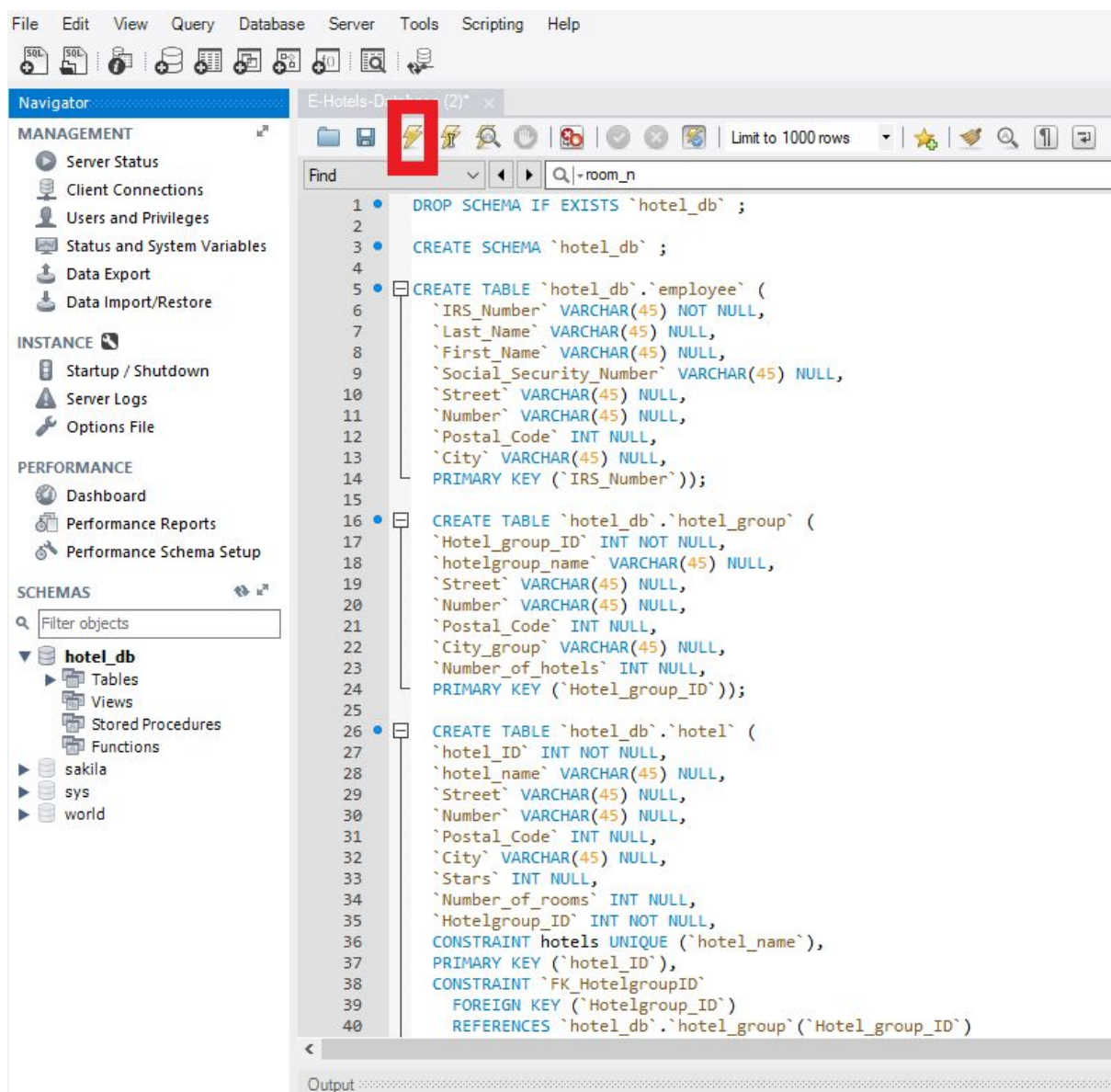
; On windows:

extension_dir = "C:\php\ext"

Ειδικά στο δεύτερο πλαίσιο συμπληρώνουμε το σωστό path του αρχείου το οποίο περιλαμβάνει το php_mysql.dll.

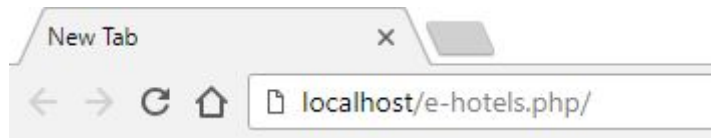
Όλα τα παραπάνω είναι απαραίτητα ώστε να γίνει ορθά η σύνδεση της PHP με τον Server.

Για την εγκατάσταση της, θα πρέπει αρχικά να κατασκευάσει τη βάση δεδομένων. Αυτό θα γίνει ανοίγοντας τον κώδικα SQL που έχουμε γράψει (E-Hotels-Database.sql) στο MySQL Workbench και εκτελώντας τον.



Για το user interface της εφαρμογής θα πρέπει να αντιγράψει το σύνολο των αρχείων που βρίσκονται μέσα στο φάκελο **e-hotels-php** και να τα επικολλήσει στον φάκελο **htdocs** του Apache24. Στη

συνέχεια για να περιηγηθεί στην εφαρμογή θα πρέπει να πληκτρολογήσει τη διεύθυνση localhost/e-hotels.php στον web browser που χρησιμοποιεί όπως φαίνεται στην παρακάτω εικόνα.



Πηγές

Για τη διαμόρφωση της εφαρμογής συμβουλευτήκαμε τις διαφάνειες του μαθήματος αλλά και τους παρακάτω συνδέσμους:

- <https://dev.mysql.com/doc/refman/8.0/en/>
- <https://www.w3schools.com/php/>
- <https://www.w3schools.com/css/default.asp>
- <https://www.w3schools.com/html/default.asp>
- <https://www.w3schools.com/sql/default.asp>
- <http://php.net/docs.php>