

# Περιγραφή Βάσης Δεδομενων

## ΟΔΗΓΙΕΣ

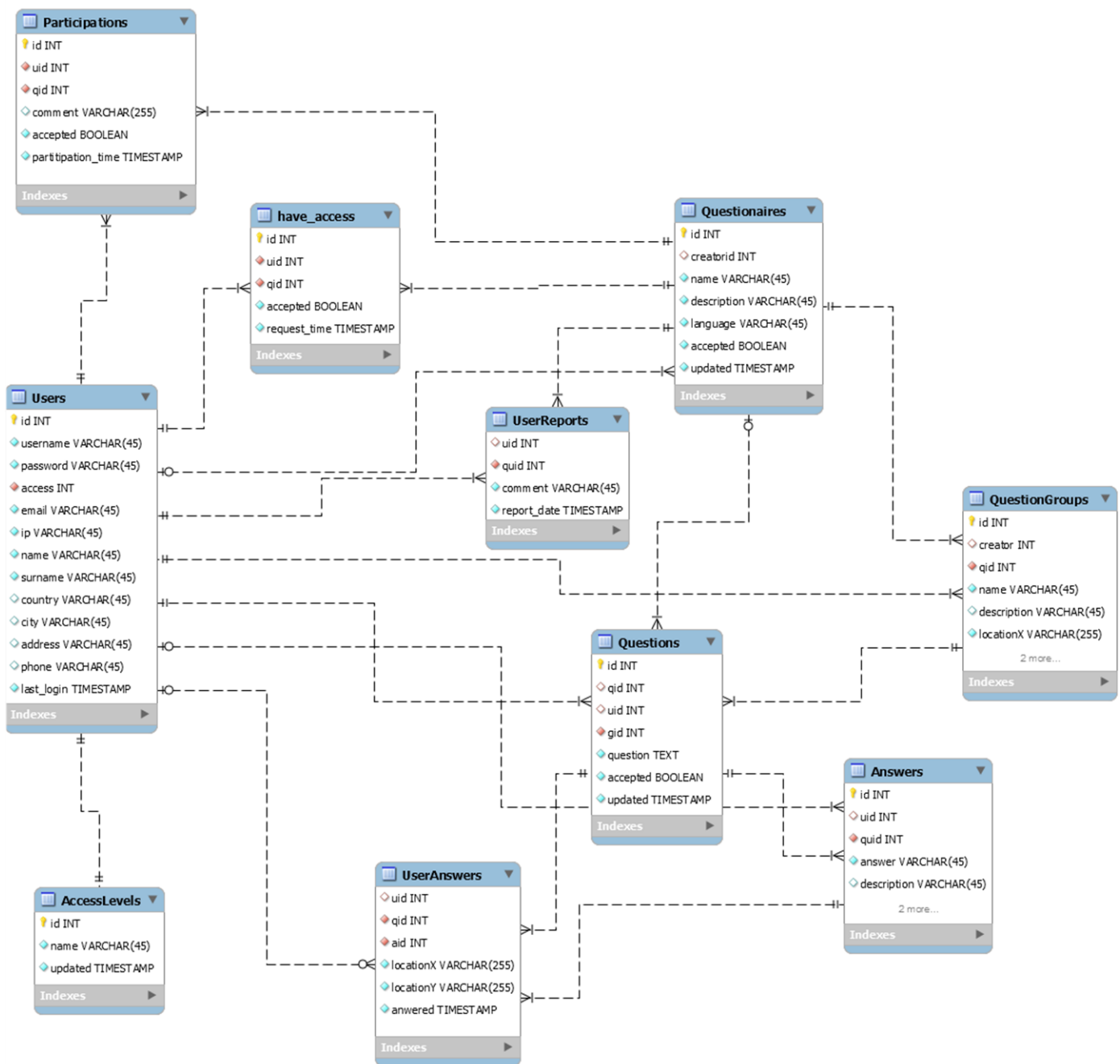
### Αναπαράσταση χρωμάτων

- 1) Ονοματα πινάκων
- 2) Τα ονοματα των υποχρεωτικών (NOT NULL) πεδιων τους
- 3) Τα ονοματα των προαιρετικων πεδίων τους

### Δομη επεξήγησης

- 1) Σειρα 1<sup>η</sup> Ονοματα πινάκων
- 2) Σειρα 2<sup>η</sup> Πεδια πινάκων
- 3) Σειρα 3<sup>η</sup> Επεξηγηση

## ΔΙΑΓΡΑΜΜΑ ΕΕΡ



## ΕΠΕΞΗΓΗΣΗ

- I. **Users** : Ο πίνακας αυτός διαθέτει όλους τους χρήστες με τα στοιχεία τους.
  - a. **ID** : **Primary key** Auto increment
  - b. **Username** : **UNIQUE** όνομα χρήστη
  - c. **Password** : Κωδικός χρήστη ( κρυπτογραφημένος )
  - d. **Access** : **Foreign Key** με αναφορά στον πίνακα AccessLevels στο πεδίο id (Δικαιώματα χρήστη)
  - e. **Email** : **UNIQUE** email address

- f. **IP** : Ip address (For each connection updated)
  - g. **Name** : πραγματικό όνομα
  - h. **Surname** : επίθετο χρήστη
  - i. **Country** : χώρα διαμονής του χρήστη
  - j. **City** : πόλη διαμονής του
  - k. **Address** : οδός κατοικίας του
  - l. **Phone** : τηλέφωνο επικοινωνίας
  - m. **Last\_login** : τελευταία σύνδεση χρήστη στο σύστημα μας
- II. **AccessLevels** : Ο πίνακας αυτός διαθέτει τα δικαιώματα των χρηστών. 1 = administrator 2=user 3=examiner 4=blocked.
- a. **ID** : **Primary key** Auto increment
  - b. **Name** : όνομα access level (Administrator,User,Examiner,Blocked)
  - c. **Updated** : Ποτε έγινε τελευταία ενημέρωση αυτή η εγγραφή
- III. **Questionnaires** : Ο Πίνακας διαθέτει όλες τις εγγραφές των ερωτηματολογίων.
- a. **ID** : **Primary key** Auto increment
  - b. **Creator** : **Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id ( δημιουργός ερωτηματολογίου. )
    - i. Το πεδίο αυτό μπορεί να είναι και NULL διότι σε περίπτωση που ο δημιουργός του questionnaire αποφασίσει/σούμε ότι θέλει/ούμε να διαγράψει/ούμε τα στοιχεία του από την βάση δεδομένων τα ερωτηματολόγια που έχει δημιουργήσει θα θέλαμε να κρατηθούν εφόσον και εκείνος το επιθυμεί.
  - c. **Name** : Όνομα ερωτηματολογίου
  - d. **Description** : Περιγραφή ερωτηματολογίου
  - e. **Language** : Γλώσσα ερωτηματολογίου
  - f. **Accepted** : Καθορίζει αν το ερωτηματολόγιο έχει γίνει αποδεκτό από τον/ους διαχειριστές του συστήματος (Default 0 = false)
  - g. **Updated** : Τελευταία ενημέρωση του ερωτηματολογίου (Μέρα ώρα δημιουργίας καθώς και επείτα ενημέρωση για κάθε διορθώση του)
- IV. **Questions** : Ο Πίνακας διαθέτει όλες τις ερωτήσεις των ερωτηματολογίων.
- a. **ID** : **Primary key** Auto increment
  - b. **Qid** : **Foreign Key** με αναφορά στον πίνακα Questionnaires στο πεδίο id
    - i. Το πεδίο αυτό είναι απαραίτητο να είναι ΚΑΙ NULL για τον λόγο ότι αν γίνει καταλαθος/εσκεμμένα διαγραφή κάποιου ερωτηματολογίου ίσως και να θέλαμε τις ερωτήσεις του να χρησιμοποιηθούν σε άλλο.Οποτε θα γίνει εφικτό να γίνει διαγραφή ερωτηματολογίου αλλά τα questions θα μείνουν ως null στο που ανήκουν. Αρα κατά την υλοποίηση κάποιου μενού διαγραφής θα χρειαστεί να διαγράψουμε πρώτα τα questiongroups + τις ερωτήσεις.Αυτό γίνεται για προστασία μας αν κάποιος από εμάς σβήσει κατά λάθος απευθείας από την βάση κάποιο ερωτηματολόγιο που δε πρέπει ώστε να μην χαθούν απευθείας τα πάντα και ψαχνουμε back ups.
  - c. **Uid** : **Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id
    - i. Το πεδίο αυτό είναι απαραίτητο να είναι ΚΑΙ NULL για τον λόγο ότι αν ένας χρήστης θελήσει/σούμε την διαγραφή του από το σύστημα ίσως και να θέλαμε τις ερωτήσεις που θα έχει καταχωρήσει

προηγούμενως σε περίπτωση που ήταν examiner εφόσον και εκείνος το επιθυμεί.

- d. **Gid** : **Foreign Key** με αναφορά στον πίνακα QuestionGroups στο πεδίο id
    - i. Σε περίπτωση που γίνει διαγραφή ενός QuestionGroup θα διαγραφουν και όλες οι ερωτήσεις που διαθέτει η ομάδα αυτή
  - e. **Question** : **UNIQUE** κείμενο ερώτησης
  - f. **Accepted** : Καθορίζει αν η ερώτηση έχει γίνει αποδεκτή από τον/ους διαχειριστές του συστήματος (Default 0 = false)
  - g. **Updated** : Τελευταία ενημέρωση της ερώτησης (Μέρα ώρα δημιουργίας καθώς επίσης ενημέρωση για κάθε διορθωση της)
- V. **Answers** : Ο πίνακας αυτός διαθέτει όλες τις απαντήσεις των ερωτήσεων
- a. **ID** : **Primary key** Auto increment
  - b. **Uid** : **Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id
    - i. Το πεδίο αυτό είναι απαραίτητο να είναι ΚΑΙ NULL για τον λόγο ότι αν ένας χρήστης θελήσει/σουμε την διαγραφή του από το σύστημα ίσως και να θέλαμε τις απαντήσεις που θα έχει καταχωρήσει / ενημερώσει προηγούμενως να τις κρατήσουμε σε περίπτωση που ήταν examiner εφόσον και εκείνος το επιθυμεί.
  - c. **Qid** : **Foreign Key** με αναφορά στον πίνακα Questions στο πεδίο id
    - i. Σε περίπτωση διαγραφής κάποιας ερώτησης θα διαγραφουν σαφώς χωρίς κανένα πρόβλημα και οι απαντήσεις της.
  - d. **Answer** : κείμενο απάντησης
  - e. **Description** : κείμενο περιγραφής απάντησης
    - i. Το πεδίο αυτό μπορεί να είναι ΚΑΙ NULL για τον λόγο ότι ένας examiner θα καθορίσει αν θέλει μετά από την απάντηση του χρήστη αν θέλει να εμφανίζεται κάποια ίσως αιτιολόγηση για την επιλογή αυτής ως σωστή απάντηση
  - f. **isCorrect** : Καθορίζει αν η συγκεκριμένη απάντηση είναι η σωστή
  - g. **Updated** : Τελευταία ενημέρωση της απάντησης (Μέρα ώρα δημιουργίας καθώς επίσης ενημέρωση για κάθε διορθωση της)
- VI. **QuestionGroups** : Ο πίνακας αυτός διαθέτει όλες τις ομάδες ερωτήσεων
- a. **ID** : **Primary key** Auto increment
  - b. **Creator** : **Foreign Key** με αναφορά στον πίνακα Users στο πεδίο ID
    - i. Το πεδίο αυτό είναι απαραίτητο να είναι ΚΑΙ NULL για τον λόγο ότι αν ένας χρήστης θελήσει την διαγραφή του από το σύστημα ίσως και να θέλαμε τις ομάδες ερωτήσεων που θα έχει απαντήσει να τις κρατήσουμε εφόσον και εκείνος το επιθυμεί
  - c. **Qid** : **Foreign Key** με αναφορά στον πίνακα Questionnaires στο πεδίο id
    - i. Σε περίπτωση προσπαθείας διαγραφής κάποιου ερωτηματολογίου υπάρχει η εντολή RESTRICT οπότε δε θα γίνει εφικτό σε περίπτωση που υπάρχουν questiongroup οπότε κατά την υλοποίηση κάποιου μένου διαγραφής θα χρειαστεί να διαγράψουμε πρώτα τα questiongroups + τις ερωτήσεις. Αυτό γίνεται για προστασία μας αν κάποιος από εμάς σβήσει κατά λάθος από την βάση κάποιο ερωτηματολόγιο που δε πρέπει ώστε να μην χαθούν απευθείας τα πάντα και ψαχνούμε back ups.
  - d. **Name** : **UNIQUE** Καθορίζει το όνομα της ομάδας
  - e. **Description** : κείμενο περιγραφής ομάδας

- i. Το πεδίο αυτό μπορεί να είναι ΚΑΙ NULL για τον λόγο ότι ένας examiner θα καθορίσει αν θέλει να γράψει κάποιο ίσως σχόλιο περιγραφής αυτού του συνολικού group ( πχ λευκός πυργός ερωτήσεις σχετικά με την ιστορία του )
  - f. **LocationX** : Συντεταγμένη X της τοποθεσίας που πρέπει να βρίσκεται ο χρήστης για να του εμφανιστεί το group των ερωτήσεων
    - i. Προφανώς θα υπάρχει μια αποκλυση για να φτιάξουμε ένα κύκλο που θα πρέπει να βρίσκεται και όχι μια στανταρ τοποθεσία
  - g. **LocationY** : Συντεταγμένη Y της τοποθεσίας που πρέπει να βρίσκεται ο χρήστης για να του εμφανιστεί το group των ερωτήσεων
    - i. Προφανώς θα υπάρχει μια αποκλυση για να φτιάξουμε ένα κύκλο που θα πρέπει να βρίσκεται και όχι μια στανταρ τοποθεσία
  - h. **Updated** : Τελευταία ενημέρωση της ομάδας (Μέρα ωρα δημιουργίας καθώς επίσης ενημέρωση για κάθε διορθωση της)
- VII. **Participations** : Ο πίνακας αυτός διαθέτει όλους τους συμμετεχόντες χρήστες των ερωτηματολογίων.
  - a. **ID : Primary key** Auto increment
  - b. **Uid : Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id
    - i. Σε περίπτωση διαγραφής ενός χρήστη ή συμμετοχής του επίσης ακυρώνεται/διαγράφεται από το σύστημα
  - c. **Qid : Foreign Key** με αναφορά στον πίνακα Questionnaires στο πεδίο id
    - i. Το ερωτηματολόγιο το οποίο ο χρήστης συμμετέχει.(Σε περίπτωση προσπάθειας διαγραφής του ερωτηματολογίου υπάρχει εντολή RESTRICT ώστε να γίνει κριτική από τον διαχειριστή πρώτα αν οντως θέλει να διαγραφεί το ερωτηματολόγιο με τους συγκεκριμένους χρήστες που έχουν εγγραφεί στο σύστημα.Ακόμα μια προστασία από μια καταλαθος διαγραφή απευθείας από την βάση και ψαχνουμε μετά back ups.
  - d. **Comment** : Καποιο σχόλιο από τον χρήστη για την αίτηση συμμετοχής στο σύστημα. Ίσως ορίσουμε να μας στέλνουν καποιο Αριθμο Μητρώου οτιδήποτε , ή πάνω κατά την αίτηση για συμμετοχή αν θέλουν να ρωτήσουν και οι ίδιοι κάτι.
  - e. **Accepted** : Καθορίζει αν η αίτηση για συμμετοχή σε ένα ερωτηματολόγιο έγινε αποδεκτή από τους διαχειριστές του συστήματος (Default 0 = false)
  - f. **Participation\_time** : Μέρα,Ωρα,Λεπτα που ο χρήστης έκανε αίτηση συμμετοχής στο σύστημα μας.
- VIII. **UserAnswers** : Ο πίνακας αυτός διαθέτει όλες τις πληροφορίες για τις απαντήσεις των χρηστών.
  - a. **Uid : Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id
    - i. Το πεδίο αυτό είναι απαραίτητο να είναι ΚΑΙ NULL για τον λόγο ότι αν ένας χρήστης θελήσει την διαγραφή του από το σύστημα ίσως και να θέλαμε τις ερωτήσεις που θα έχει απάντηση για λόγους εμφάνισης στατιστικών πως τα πήγαν οι χρήστες μας και έναν βαθμό δυσκολίας στο όλο αυτό ερωτηματολόγιο εφόσον και εκείνος το επιθυμεί
  - b. **Qid : Foreign Key** με αναφορά στον πίνακα Questions στο πεδίο id
  - c. **Aid : Foreign Key** με αναφορά στον πίνακα Answers στο πεδίο id

- d. **LocationX** : Συντεταγμένη X της τοποθεσίας που βρισκόταν ο χρήστης όταν απάντησε την ερώτηση
  - e. **LocationY** : Συντεταγμένη Y της τοποθεσίας που βρισκόταν ο χρήστης όταν απάντησε την ερώτηση
  - f. **Answered** : Μερα,ωρα,δευτερολεπτα που απαντηθηκε η ερώτηση απο τον χρήστη
- IX. **UserReports** : Ο πίνακας αυτός διαθέτει όλες τις αναφορές/σχολία των χρηστών για το ερωτηματολόγιο που συμμετείχαν
- a. **Uid** : **Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id
    - i. Το πεδίο αυτό είναι απαραίτητο να είναι ΚΑΙ NULL για τον λόγο ότι αν ένας χρήστης θελήσει την διαγραφή του απο το σύστημα ίσως και να θέλαμε να κρατήσουμε τα σχολία που εκανε για καποιο ερωτηματολόγιο για να τα λαβουμε υποψην αργότερα σε καποιο ελεγχο των ερωτησεων μας εφοσον και εκεινος το επιθυμει.
  - b. **Qid** : **Foreign Key** με αναφορά στον πίνακα Questions στο πεδίο id
  - c. **Comment** : Αναφορά/Σχολίο του χρηστη σχετικά με καποιο λαθος,διορθωση,επισημανση που εχει να κανει σε ενα ερωτηματολόγιο για κατι που παρατηρησε.
  - d. **Report\_date** : Μερα,ωρα,δευτερολεπτα που σταλθηκε η αναφορά απο τον χρήστη
- X. **Have\_access** : Ο πίνακας αυτός διαθέτει όλα τα δικαιώματα των χρηστών πάνω σε καποιο ερωτηματολόγιο. Ο πίνακας αυτός είναι για τους
- a. **ID** : **Primary key** Auto increment
  - b. **Uid** : **Foreign Key** με αναφορά στον πίνακα Users στο πεδίο id
    - i. Σε περιπτωση διαγραφης ενος χρηστη σε περιπτωση που ειχε δικαιωματα σε καποιο ερωτηματολόγιο τοτε ακυρωνονται/διαγραφονται απο το σύστημα και εκεινα γιατι δε υπαρχει λογος να μειουν εντος του συστηματος.
  - c. **Qid** : **Foreign Key** με αναφορά στον πίνακα Questionnaires στο πεδίο id
    - i. Το ερωτηματολόγιο το οποιο ο χρηστης εχει δικαιωματα.(Σε περιπτωση διαγραφης του ερωτηματολογιου ακυρωνονται/διαγραφονται απο το σύστημα και οι διαχειριστές του
  - d. **Accepted** : Καθοριζει αν έγινε αποδεκτη η αιτηση για δικαιωματα σε καποιο ερωτηματολόγιο του συστηματος μας.
  - e. **Request\_time** : Μερα,ωρα,λεπτα που εγινε η αιτηση για παροχη δικαιωματος σε καποιο ερωτηματολόγιο του συστηματος.

## Κώδικας

-- MySQL Script generated by MySQL Workbench

-- 11/17/15 21:44:59

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE,  
SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-----

-- Schema treasure-thess

-----

DROP SCHEMA IF EXISTS `treasure-thess` ;

-----

-- Schema treasure-thess

-----

CREATE SCHEMA IF NOT EXISTS `treasure-thess` DEFAULT CHARACTER SET utf8 COLLATE  
utf8\_general\_ci ;

USE `treasure-thess` ;

-----

-- Table `treasure-thess`.`AccessLevels`

-----

DROP TABLE IF EXISTS `treasure-thess`.`AccessLevels` ;

CREATE TABLE IF NOT EXISTS `treasure-thess`.`AccessLevels` (  
 `id` INT NOT NULL AUTO\_INCREMENT COMMENT 'Auto increment access level id',  
 `name` VARCHAR(45) NOT NULL COMMENT 'Access level name',  
 `updated` TIMESTAMP NOT NULL COMMENT 'Access level updated time.',  
 PRIMARY KEY (`id`) COMMENT '',  
 UNIQUE INDEX `name\_UNIQUE` (`name` ASC) COMMENT '')  
ENGINE = InnoDB;

```

-----
-- Table `treasure-thess`.`Users`
-----

DROP TABLE IF EXISTS `treasure-thess`.`Users` ;

CREATE TABLE IF NOT EXISTS `treasure-thess`.`Users` (
  `id` INT NOT NULL COMMENT 'Auto increment user id',
  `username` VARCHAR(45) NOT NULL COMMENT 'User id given from the user. Used for the
login system.',
  `password` VARCHAR(45) NOT NULL COMMENT 'User password.Used for the login system.',
  `access` INT NOT NULL COMMENT 'Access level id (Access:id)',
  `email` VARCHAR(45) NOT NULL COMMENT 'User email address',
  `ip` VARCHAR(45) NOT NULL COMMENT 'User global ip',
  `name` VARCHAR(45) NOT NULL COMMENT 'The real name from a user',
  `surname` VARCHAR(45) NOT NULL COMMENT 'The real surname from a user',
  `country` VARCHAR(45) NULL COMMENT 'Country name where the user lives.Can be null',
  `city` VARCHAR(45) NULL COMMENT 'City name where the user lives.Can be null',
  `address` VARCHAR(45) NULL COMMENT 'Home address from a user.Can be null',
  `phone` VARCHAR(45) NULL COMMENT 'The phone number from a user.Can be null',
  `last_login` TIMESTAMP NOT NULL COMMENT 'Last login date from the user.',
  PRIMARY KEY (`id`) COMMENT "",
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) COMMENT "",
  UNIQUE INDEX `username_UNIQUE` (`username` ASC) COMMENT "",
  INDEX `alvl_idx` (`access` ASC) COMMENT "",
  CONSTRAINT `access_level_id`
    FOREIGN KEY (`access`)
      REFERENCES `treasure-thess`.`AccessLevels` (`id`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```



```

-----
-- Table `treasure-thess`.`Questionnaires`
-----

DROP TABLE IF EXISTS `treasure-thess`.`Questionnaires` ;

CREATE TABLE IF NOT EXISTS `treasure-thess`.`Questionnaires` (
  `id` INT NOT NULL AUTO_INCREMENT COMMENT 'Auto increment questionnaire id',
  `creatorid` INT NULL COMMENT 'Author id(Users:id)',
  `name` VARCHAR(45) NOT NULL COMMENT 'Questionnaire name',
  `description` VARCHAR(45) NOT NULL COMMENT 'Questionnaire description',
  `language` VARCHAR(45) NOT NULL COMMENT 'Questionnaire language',
  `accepted` TINYINT(1) NOT NULL DEFAULT 0 COMMENT 'If questionnaire accepted from one
admin',
  `updated` TIMESTAMP NOT NULL COMMENT 'Questionnaire last updated time.',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `uid_idx` (`creatorid` ASC) COMMENT '',
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) COMMENT '',
  CONSTRAINT `questionnaire_creator_user_id`
    FOREIGN KEY (`creatorid`)
      REFERENCES `treasure-thess`.`Users` (`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `treasure-thess`.`QuestionGroups`
-----

DROP TABLE IF EXISTS `treasure-thess`.`QuestionGroups` ;

```

```

CREATE TABLE IF NOT EXISTS `treasure-thess`.`QuestionGroups` (
  `id` INT NOT NULL AUTO_INCREMENT COMMENT 'Auto increment category id',
  `creator` INT NULL COMMENT '',
  `qid` INT NOT NULL COMMENT 'Quastionaires id (Quastionaires:id)',
  `name` VARCHAR(45) NOT NULL COMMENT 'Category name',
  `description` VARCHAR(45) NULL COMMENT 'Category description text. Can be null',
  `locationX` VARCHAR(255) NOT NULL COMMENT '',
  `locationY` VARCHAR(255) NOT NULL COMMENT '',
  `updated` TIMESTAMP NOT NULL COMMENT 'Category last updated time.',
  PRIMARY KEY (`id`) COMMENT '',
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) COMMENT '',
  INDEX `qid_idx` (`qid` ASC) COMMENT '',
  INDEX `questionGroup_creator_idx` (`creator` ASC) COMMENT '',
  CONSTRAINT `questionGroup_questionnaire_id`
    FOREIGN KEY (`qid`)
      REFERENCES `treasure-thess`.`Questionnaires` (`id`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `questionGroup_creator`
    FOREIGN KEY (`creator`)
      REFERENCES `treasure-thess`.`Users` (`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `treasure-thess`.`Questions`
-----

```

```

DROP TABLE IF EXISTS `treasure-thess`.`Questions` ;

```

```

CREATE TABLE IF NOT EXISTS `treasure-thess`.`Questions` (
  `id` INT NULL AUTO_INCREMENT COMMENT 'Auto increment question id',
  `qid` INT NULL COMMENT 'Questionnaires id (Questionnaires:id)',
  `uid` INT NULL COMMENT '',
  `gid` INT NOT NULL COMMENT 'QuestionGroups id (QuestionGroups:id)',
  `question` TEXT NOT NULL COMMENT 'Question name',
  `accepted` TINYINT(1) NOT NULL DEFAULT 0 COMMENT 'If this question accepted from one admin',
  `updated` TIMESTAMP NOT NULL COMMENT 'Question last updated time.',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `cid_idx` (`gid` ASC) COMMENT '',
  INDEX `questionnaire_id_idx` (`qid` ASC) COMMENT '',
  INDEX `uid_idx` (`uid` ASC) COMMENT '',
  UNIQUE INDEX `question_UNIQUE` (`question` ASC) COMMENT '',
  CONSTRAINT `QuestionGroup_id_for_this_question`
    FOREIGN KEY (`gid`)
      REFERENCES `treasure-thess`.`QuestionGroups` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `Questionnaire_id_for_this_question`
    FOREIGN KEY (`qid`)
      REFERENCES `treasure-thess`.`Questionnaires` (`id`)
      ON DELETE SET NULL
      ON UPDATE CASCADE,
  CONSTRAINT `user_update_last_time_this_QuestionGroup`
    FOREIGN KEY (`uid`)
      REFERENCES `treasure-thess`.`Users` (`id`)
      ON DELETE SET NULL
      ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `treasure-thess`.`Answers`
-----

DROP TABLE IF EXISTS `treasure-thess`.`Answers`;

CREATE TABLE IF NOT EXISTS `treasure-thess`.`Answers` (
  `id` INT NOT NULL AUTO_INCREMENT COMMENT 'Auto increment id',
  `uid` INT NULL COMMENT 'user who update last time this answer',
  `qid` INT NOT NULL COMMENT 'Question id (Questions:id)',
  `answer` VARCHAR(45) NOT NULL COMMENT 'Answer name',
  `description` VARCHAR(45) NULL COMMENT 'Answer description',
  `isCorrect` TINYINT(1) NOT NULL COMMENT 'if this answer is the correct for the question id',
  `updated` TIMESTAMP NOT NULL COMMENT 'when answer created',
  PRIMARY KEY (`id`) COMMENT "",
  INDEX `qid_idx` (`qid` ASC) COMMENT "",
  INDEX `user_updated_last_time_this_answer_idx` (`uid` ASC) COMMENT "",
  CONSTRAINT `question_for_this_answer`
    FOREIGN KEY (`qid`)
      REFERENCES `treasure-thess`.`Questions` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `user_who_update_last_time_this_answer`
    FOREIGN KEY (`uid`)
      REFERENCES `treasure-thess`.`Users` (`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `treasure-thess`.`UserAnswers`
-----

DROP TABLE IF EXISTS `treasure-thess`.`UserAnswers` ;

CREATE TABLE IF NOT EXISTS `treasure-thess`.`UserAnswers` (
  `uid` INT NULL COMMENT 'User id (Users:id)',
  `qid` INT NOT NULL COMMENT 'Question id (Questions:id)',
  `aid` INT NOT NULL COMMENT 'Answer id (Answers:id)',
  `locationX` VARCHAR(255) NOT NULL COMMENT "",
  `locationY` VARCHAR(255) NOT NULL COMMENT "",
  `answered` TIMESTAMP NOT NULL COMMENT 'When the question answered',
  INDEX `uid_idx` (`uid` ASC) COMMENT "",
  INDEX `qid_idx` (`qid` ASC) COMMENT "",
  INDEX `aid_idx` (`aid` ASC) COMMENT "",
  CONSTRAINT `User_who_answer_this_question`
    FOREIGN KEY (`uid`)
      REFERENCES `treasure-thess`.`Users` (`id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
  CONSTRAINT `question_answered`
    FOREIGN KEY (`qid`)
      REFERENCES `treasure-thess`.`Questions` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `answer_which_answered`
    FOREIGN KEY (`aid`)
      REFERENCES `treasure-thess`.`Answers` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)

```

```
ENGINE = InnoDB;
```

```
-- Table `treasure-thess`.`UserReports`
```

```
DROP TABLE IF EXISTS `treasure-thess`.`UserReports` ;
```

```
CREATE TABLE IF NOT EXISTS `treasure-thess`.`UserReports` (  
  `uid` INT NULL COMMENT 'user id (Users:id)',  
  `quid` INT NOT NULL COMMENT 'Questionnaires id (Questionnaires:id)',  
  `comment` VARCHAR(45) NOT NULL COMMENT 'Report comment from a user for a  
question id',  
  `report_date` TIMESTAMP NOT NULL COMMENT '',  
  INDEX `uid_idx` (`uid` ASC) COMMENT '',  
  INDEX `qid_idx` (`quid` ASC) COMMENT '',  
  CONSTRAINT `user_who_send_the_report`  
    FOREIGN KEY (`uid`)  
    REFERENCES `treasure-thess`.`Users` (`id`)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE,  
  CONSTRAINT `questionnaire_reported`  
    FOREIGN KEY (`quid`)  
    REFERENCES `treasure-thess`.`Questionnaires` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
-- Table `treasure-thess`.`Participations`
```

```
-- -----  
DROP TABLE IF EXISTS `treasure-thess`.`Participations` ;
```

```
CREATE TABLE IF NOT EXISTS `treasure-thess`.`Participations` (  
  `id` INT NOT NULL AUTO_INCREMENT COMMENT 'Auto increment question id',  
  `uid` INT NOT NULL COMMENT 'Users id (Users:id)',  
  `qid` INT NOT NULL COMMENT 'Questionnaires id (Questionnaires:id)',  
  `comment` VARCHAR(255) NULL COMMENT 'User partitipation comment to  
administrators.',  
  `accepted` TINYINT(1) NOT NULL DEFAULT 0 COMMENT '',  
  `partitipation_time` TIMESTAMP NOT NULL COMMENT 'User partitipation time.',  
  PRIMARY KEY (`id`) COMMENT '',  
  INDEX `qid_idx` (`qid` ASC) COMMENT '',  
  INDEX `uid_idx` (`uid` ASC) COMMENT '',  
  CONSTRAINT `questionnaire_partitipations`  
    FOREIGN KEY (`qid`)  
    REFERENCES `treasure-thess`.`Questionnaires` (`id`)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE,  
  CONSTRAINT `participant_user`  
    FOREIGN KEY (`uid`)  
    REFERENCES `treasure-thess`.`Users` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `treasure-thess`.`have_access`  
-- -----
```

```
DROP TABLE IF EXISTS `treasure-thess`.`have_access` ;
```

```

CREATE TABLE IF NOT EXISTS `treasure-thess`.`have_access` (
  `id` INT NOT NULL COMMENT "",
  `uid` INT NOT NULL COMMENT "",
  `qid` INT NOT NULL COMMENT "",
  `accepted` TINYINT(1) NOT NULL DEFAULT 0 COMMENT "",
  `request_time` TIMESTAMP NOT NULL COMMENT "",
  PRIMARY KEY (`id`) COMMENT "",
  INDEX `uid_idx` (`uid` ASC) COMMENT "",
  INDEX `qid_idx` (`qid` ASC) COMMENT "",
  CONSTRAINT `user_access_request`
    FOREIGN KEY (`uid`)
      REFERENCES `treasure-thess`.`Users` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `questionnaire_id_for_access`
    FOREIGN KEY (`qid`)
      REFERENCES `treasure-thess`.`Questionnaires` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;

```

```

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

```

```

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```