ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Προχωρημένα Θέματα Τεχνολογίας και Εφαρμογών Βάσεων Δεδομένων

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ για το ακαδημαϊκο έτος 2015-2016

ΟΜΑΔΑ ΧΧΧ

Φοιτητής 1, ΑΜ

Φοιτητής 2, ΑΜ

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2016

ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

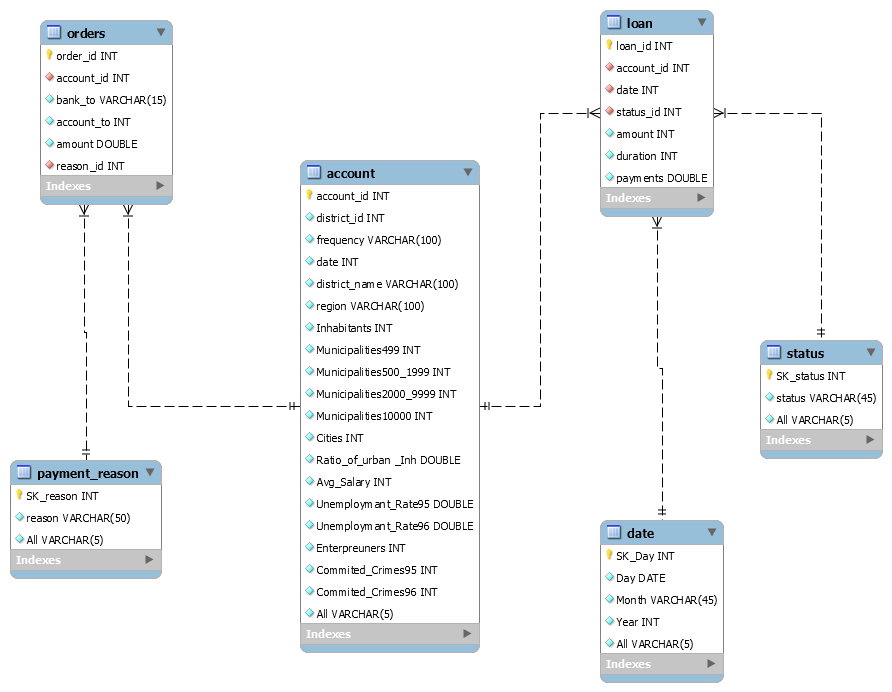
|  |  |  |  |
| --- | --- | --- | --- |
| Ημερομηνία | Έκδοση | Περιγραφή | Συγγραφέας |
| yyyy/mm/dd | x.x |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Το κείμενο συμπληρώνεται προοδευτικά, όπως προχωρείτε στις φάσεις του Project.

# βαση δεδομενων

Στην παρούσα ενότητα περιγράφονται τα σχήματα της βάσης (ή βάσεων, αν είναι παραπάνω από μία) δεδομένων που χρησιμοποιούνται στο project.

## Σχεσιακό σχήμα σε λογικό επίπεδο



Σχήμα 1.1 Σχεσιακό σχήμα της βάσης δεδομένων του συστήματος

Section-break (continuous)

-- MySQL Script generated by MySQL Workbench

-- Sat Feb 19 13:48:30 2022

-- 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 mydb

-- ----------------------

-- ----------------------

-- Schema pkdd99\_star

-- ----------------------

DROP SCHEMA IF EXISTS `pkdd99\_star` ;

-- ----------------------

-- Schema pkdd99\_star

-- ----------------------

CREATE SCHEMA IF NOT EXISTS `pkdd99\_star` DEFAULT CHARACTER SET utf8 ;

USE `pkdd99\_star` ;

-- ----------------------

-- Table `pkdd99\_star`.`account`

-- ----------------------

DROP TABLE IF EXISTS `pkdd99\_star`.`account` ;

CREATE TABLE IF NOT EXISTS `pkdd99\_star`.`account` (

`account\_id` INT NOT NULL,

`district\_id` INT NOT NULL,

`frequency` VARCHAR(100) NOT NULL,

`date` INT NOT NULL,

`district\_name` VARCHAR(100) NOT NULL,

`region` VARCHAR(100) NOT NULL,

`Inhabitants` INT NOT NULL,

`Municipalities499` INT NOT NULL,

`Municipalities500\_1999` INT NOT NULL,

`Municipalities2000\_9999` INT NOT NULL,

`Municipalities10000` INT NOT NULL,

`Cities` INT NOT NULL,

`Ratio\_of\_urban \_Inh` DOUBLE NOT NULL,

`Avg\_Salary` INT NOT NULL,

`Unemploymant\_Rate95` DOUBLE NOT NULL,

`Unemploymant\_Rate96` DOUBLE NOT NULL,

`Enterpreuners` INT NOT NULL,

`Commited\_Crimes95` INT NOT NULL,

`Commited\_Crimes96` INT NOT NULL,

`All` VARCHAR(5) NOT NULL,

PRIMARY KEY (`account\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- ----------------------

-- Table `pkdd99\_star`.`date`

-- ----------------------

DROP TABLE IF EXISTS `pkdd99\_star`.`date` ;

CREATE TABLE IF NOT EXISTS `pkdd99\_star`.`date` (

`SK\_Day` INT NOT NULL,

`Day` DATE NOT NULL,

`Month` VARCHAR(45) NOT NULL,

`Year` INT NOT NULL,

`All` VARCHAR(5) NOT NULL,

PRIMARY KEY (`SK\_Day`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- ----------------------

-- Table `pkdd99\_star`.`status`

-- ----------------------

DROP TABLE IF EXISTS `pkdd99\_star`.`status` ;

CREATE TABLE IF NOT EXISTS `pkdd99\_star`.`status` (

`SK\_status` INT NOT NULL,

`status` VARCHAR(45) NOT NULL,

`All` VARCHAR(5) NOT NULL,

PRIMARY KEY (`SK\_status`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- ----------------------

-- Table `pkdd99\_star`.`loan`

-- ----------------------

DROP TABLE IF EXISTS `pkdd99\_star`.`loan` ;

CREATE TABLE IF NOT EXISTS `pkdd99\_star`.`loan` (

`loan\_id` INT NOT NULL,

`account\_id` INT NOT NULL,

`date` INT NOT NULL,

`status\_id` INT NOT NULL,

`amount` INT NOT NULL,

`duration` INT NOT NULL,

`payments` DOUBLE NOT NULL,

PRIMARY KEY (`loan\_id`),

CONSTRAINT `account\_fk`

FOREIGN KEY (`account\_id`)

REFERENCES `pkdd99\_star`.`account` (`account\_id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `date\_fk`

FOREIGN KEY (`date`)

REFERENCES `pkdd99\_star`.`date` (`SK\_Day`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `status\_fk`

FOREIGN KEY (`status\_id`)

REFERENCES `pkdd99\_star`.`status` (`SK\_status`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

CREATE INDEX `account\_loan\_fk\_idx` ON `pkdd99\_star`.`loan` (`account\_id` ASC);

CREATE INDEX `date\_loan\_fk\_idx` ON `pkdd99\_star`.`loan` (`date` ASC);

CREATE INDEX `status\_loan\_fk\_idx` ON `pkdd99\_star`.`loan` (`status\_id` ASC);

-- ----------------------

-- Table `pkdd99\_star`.`payment\_reason`

-- ----------------------

DROP TABLE IF EXISTS `pkdd99\_star`.`payment\_reason` ;

CREATE TABLE IF NOT EXISTS `pkdd99\_star`.`payment\_reason` (

`SK\_reason` INT NOT NULL,

`reason` VARCHAR(50) NOT NULL,

`All` VARCHAR(5) NOT NULL,

PRIMARY KEY (`SK\_reason`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

-- ----------------------

-- Table `pkdd99\_star`.`orders`

-- ----------------------

DROP TABLE IF EXISTS `pkdd99\_star`.`orders` ;

CREATE TABLE IF NOT EXISTS `pkdd99\_star`.`orders` (

`order\_id` INT NOT NULL,

`account\_id` INT NOT NULL,

`bank\_to` VARCHAR(15) NOT NULL,

`account\_to` INT NOT NULL,

`amount` DOUBLE NOT NULL,

`reason\_id` INT NOT NULL,

PRIMARY KEY (`order\_id`),

CONSTRAINT `account\_fk2`

FOREIGN KEY (`account\_id`)

REFERENCES `pkdd99\_star`.`account` (`account\_id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `reason\_fk`

FOREIGN KEY (`reason\_id`)

REFERENCES `pkdd99\_star`.`payment\_reason` (`SK\_reason`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

CREATE INDEX `account\_orders\_fk\_idx` ON `pkdd99\_star`.`orders` (`account\_id` ASC);

CREATE INDEX `reason\_orders\_fk\_idx` ON `pkdd99\_star`.`orders` (`reason\_id` ASC);

SET SQL\_MODE=@OLD\_SQL\_MODE;

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

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

Section-break (continuous)

Πρακτικά, μπορείτε και αναλυτική λίστα των εντολών κατασκευής πινάκων, αλλά οπωσδήποτε αρχίστε με ένα workbench screenshot.

## Σχεσιακό σχήμα σε φυσικο επίπεδο

Όταν θα έχετε στήσει και ρυθμίσει τη βάση δεδομένων σας, εδώ καταγράφονται και οι ρυθμίσεις σε φυσικό επίπεδο. Ενδεικτικά:

### Ρύθμιση των παραμέτρων του dbms

storage engine,memory allocation (of various kinds), …

### Ρύθμιση του φυσικού σχήματος της βάσης δεδομένων

ορισμός πιθανών ευρετηρίων (indexes), όψεων (views) που είναι υλοποιημένες ή μη, αλλαγές στο σχήμα των πινάκων για λόγους απόδοσης, κλπ. Τεκμηριώστε τα παραπάνω με βάση τα πλάνα από τα ερωτήματα που καθυστερούν ή με βάση την εσωτερική δομή του κώδικα και της δυσκολίας συγγραφής του.

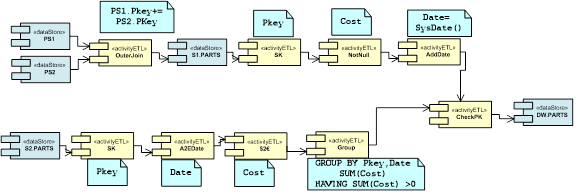
### Ρύθμιση ασφάλειας

ορισμός δικαιωμάτων καταχώρησης ή ανάκτησης δεδομένων σε διαφορετικούς ρόλους και χρήστες του συστήματος.

# αρχιτεκτονικη Λογισμικού

## αρχιτεκτονικη και δομη ETL

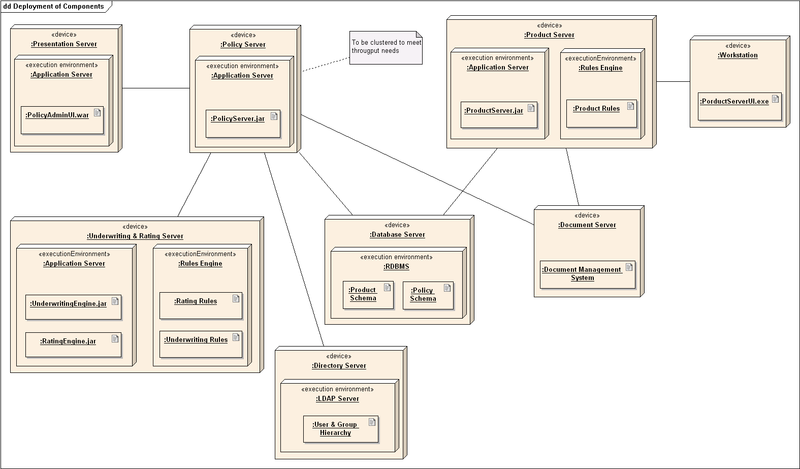
Αρχικά πρέπει να φέρουμε τα δεδομένα μέσα στη βάση μας για περαιτέρω επεξεργασία. Εδώ καταγράφεται η αρχιτεκτονική της ETL διαδικασίας (είτε μέσω εργαλείου, είτε μέσω των όποιων scripts προεπεξεργασίας και φόρτωσης δεδομένων φτιάξετε). **Είναι σημαντικό η διαδικασία να καταγραφεί με ακρίβεια στις λεπτομέρειες**. Μπορείτε να χρησιμοποιήσετε UML-based / BPMN /ETL-specific formalisms για τη διαγραμματική απεικόνιση. Δείτε τις σχετικές οδηγίες στο συνοδευτικό κείμενο στο web site του μαθήματος.

Σχήμα 2.1Παράδειγμα τεκμηρίωσης των μετασχηματισμών ETL με ένα UML component diagram

Λογικά, για ότι είναι αυτοματοποιημένο, αρκεί να πείτε τι ρυθμίσεις χρειάζονται. Αν έχετε όμως παρεμβάσεις που γίνονται manually, πρέπει να καταγραφούν επίσης οι λεπτομέρειες.

## Διαγράμματα ΠΑΚΕΤΩΝ / υποσυστημάτων kentρικησ εφαρμογησ

Το διάγραμμα για τα υποσυστήματα / πακέτα του λογισμικού που κατασκευάσατε ως κεντρική εφαρμογή επερώτησης. Ο στόχος είναι να φανεί η high-level αρχιτεκτονική του συστήματος, χωρίς λεπτομέρειες των επί μέρους κλάσεων. Κάποιος πολύ σύντομος σχολιασμός επίσης.



Σχήμα 2.2Deployment diagram of a system (from Wikipedia: https://commons.wikimedia.org/wiki/File:Deployment\_Diagram.PNG)

## Διαγράμμα(τα) Κλάσεων κεντρικησ εφαρμογησ

Αν η ανάπτυξη γίνει αντικειμενοστρεφώς, εδώ μπαίνουν τα διαγράμματα κλάσεων + ο σχολιασμός της κεντρικής εφαρμογής. Αλλιώς μπαίνουν διαγράμματα που διευκολύνουν την κατανόηση της εσωτερικής αρχιτεκτονικής του λογισμικού (π.χ., component/ deployment diagrams / …)

# Υποδείγματα ερωτήσεων και απαντήσεων

Screen dumps από τα σημαντικά τμήματα του λογισμικού.

Σαν ένα σύντομο manual…

# λοιπά σχόλια

Ότι άλλα σχόλια υπάρχουν