

FINAL PROJECT

Ενδεικτικές Λύσεις Φάσης Α – Κανονικοποίηση και Δημιουργία Λογικού Σχήματος

1. Δημιουργία πίνακα campData και μαζική εισαγωγή δεδομένων από το αρχείο campData.txt

```
CREATE TABLE campData(  
    bookCode int NULL,  
    bookDt date NULL,  
    payCode int NULL,  
    payMethod char(2) NULL,  
    custCode int NULL,  
    custName varchar(30) NULL,  
    custSurname varchar (30) NULL,  
    custPhone varchar (20) NULL,  
    staffNo int NULL,  
    staffName varchar (30) NULL,  
    staffSurname varchar (30) NULL,  
    totalCost numeric(19, 2) NULL,  
    campCode char(3) NULL,  
    campName varchar (50) NULL,  
    numOfEmp int NULL,  
    empNo int NULL,  
    catCode char (1) NULL,  
    areaM2 int NULL,  
    unitCost numeric(4,2) NULL,  
    startDt date NULL,  
    endDt date NULL,  
    noPers int NULL,  
    costPerRental numeric(19, 2) NULL  
);  
  
SET DATEFORMAT dmy;  
BULK INSERT campData  
    FROM 'C:\data\campData.txt' -- Προσαρμόστε το Path  
    WITH (FIRSTROW = 2, FIELDTERMINATOR= ',', ROWTERMINATOR = '\n');
```

2. Κανονικοποίηση

1. Τα υπογραμμισμένα γνωρίσματα αποτελούν πρωτεύοντα κλειδιά
2. Το * μπροστά από ένα γνώρισμα δηλώνει ότι το συγκεκριμένο γνώρισμα είναι ξένο κλειδί.

Σημείωση: Στην εκφώνηση αναφέρεται ρητά ότι τα δύο γνωρίσματα **totalCost** και **costPerRental** πρέπει να τα αγνοήσετε. Τα γνωρίσματα αυτά είναι παραγόμενα και μπορούν να παραληφθούν. Ωστόσο μπορούν να συμπεριληφθούν για λόγους ταχύτητας εκτέλεσης των ερωτημάτων. Στην συγκεκριμένη λύση τα έχω συμπεριλάβει στην διαδικασία της κανονικοποίησης αλλά όχι στο τελικό σχήμα της βάσης. Τα γνωρίσματα αυτά επισημαίνονται με κίτρινο χρώμα.

(R) 1NF	(R1) 2NF	3NF	
<u>bookCode</u> bookDt payCode payMethod custCode custName custSurname custPhone staffNo staffName staffSurname totalCost <u>campCode</u> campName numOfEmp <u>empNo</u> catCode areaM2 unitCost <u>startDt</u> endDt noPers costPerRental	<u>bookCode</u> bookDt payCode payMethod custCode custName custSurname custPhone staffNo staffName staffSurname totalCost (R) <u>bookCode</u> <u>campCode</u> <u>empNo</u> <u>startDt</u> endDt noPers costPerRental (R2) <u>campCode</u> <u>empNo</u> catcode areaM2 unitCost (R3) <u>campCode</u> campName numOfEmp	BOOKING <u>bookCode</u> bookDt *payCode *custcode *staffNo totalCost PAYMENT payCode payMethod CUSTOMER custCode custName custSurname custPhone STAFF <u>staff No</u> staffName staffSurname CAMPING campCode campName numOfEmp	RENTAL * <u>bookCode</u> * <u>campCode</u> * <u>empNo</u> <u>startDt</u> endDt noPers costPerRental EMPLACEMENT * <u>campCode</u> <u>empNo</u> *catCode CATEGORY <u>catCode</u> areaM2 unitCost

Βήμα 1 (1NF)

Επιλέγουμε ως υποψήφιο κλειδί για την σχέση R τα γνωρίσματα (bookCode, campCode, empNo, startDt). Η σχέση μας βρίσκεται στην πρώτη κανονική μορφή διότι δεν υπάρχουν σύνθετα και επαναλαμβανόμενα γνωρίσματα.

Βήμα 2 (2NF)

Για να είναι η σχέση **R** στην δεύτερη κανονική μορφή πρέπει κάθε μη πρωτεύον γνώρισμα να είναι εξ ολοκλήρου εξαρτώμενο από το υποψήφιο κλειδί (**bookCode**, **campCode**, **empNo**, **StartDt**) και όχι από μέρος αυτού.

Διαπιστώνουμε ότι:

- Το γνώρισμα **bookCode** παρέχει πλήρη συναρτησιακή εξάρτηση στα μη πρωτεύοντα γνωρίσματα (**bookDt**, **payCode**, **payMethod**, **custCode**, **custName**, **custSurname**, **custPhone**, **StaffNo**, **StaffName**, **staffSurname**, **totalCost**).
- Ο συνδυασμός των γνωρισμάτων **campCode**, **empNo** παρέχουν πλήρη συναρτησιακή εξάρτηση στα μη πρωτεύοντα γνωρίσματα (**catCode**, **areaM2**, **unitCost**).
- Τέλος το γνώρισμα **campCode** παρέχει μια πλήρη συναρτησιακή εξάρτηση στα μη πρωτεύοντα γνωρίσματα **campName**, **numOfEmp**

Με βάση τα παραπάνω δημιουργούμε τις σχέσεις **R1**, **R2**, **R3** και τροποποιούμε κατάλληλα την σχέση **R** (βλέπε παραπάνω πίνακα στήλη **2NF**).

Βήμα 3 (3NF)

Με απλά λόγια για να είναι μια σχέση στην τρίτη κανονική μορφή πρέπει κάθε μη πρωτεύον γνώρισμα να εξαρτάται μόνο από το κλειδί και όχι από άλλο μη πρωτεύον γνώρισμα. Εξετάζοντας τις σχέσεις της δεύτερης κανονικής μορφής διαπιστώνουμε ότι:

- **Σχέση R1:**
 - Το μη πρωτεύον γνώρισμα **payCode** παρέχει πλήρη συναρτησιακή εξάρτηση στο γνώρισμα **payMethod**.
 - Το μη πρωτεύον γνώρισμα **custCode** παρέχει πλήρη συναρτησιακή εξάρτηση στα γνωρίσματα (**custName**, **custSurname**, **custPhone**).
 - Το μη πρωτεύον γνώρισμα **staffNo** παρέχει πλήρη συναρτησιακή εξάρτηση στα γνωρίσματα (**staffName**, **staffSurname**).
- **Σχέση R2:**
 - Το μη πρωτεύον γνώρισμα **catCode** παρέχει πλήρη συναρτησιακή εξάρτηση στα γνωρίσματα (**areaM2**, **unitCost**).

Με βάση τα παραπάνω:

- δημιουργούμε τις σχέσεις **PAYMENT**, **CUSTOMER**, **STAFF**, και **CATEGORY** από την **R1** την οποία τροποποιούμε κατάλληλα την μετονομάζουμε σε **BOOKING**. (βλέπε παραπάνω πίνακα στήλη **3NF**).

- δημιουργούμε την σχέση **CATEGORY** από την **R2**, την οποία τροποποιούμε κατάλληλα και την μετονομάζουμε σε **EMPLACEMENT**.
- Τέλος μετονομάζουμε την σχέση **R3** σε **CAMPING** και την σχέση **R1** σε **RENTAL**

Η μετονομασία των σχέσεων διευκολύνει την κατανόηση του τελικού σχήματος.

3. Δημιουργία λογικού σχήματος

```
CREATE TABLE Staff
(
    staffNo int primary key,
    staffName varchar(30) NOT NULL,
    staffSurname varchar(30) NOT NULL
);

CREATE TABLE Payment
(
    payCode int primary key,
    payMethod varchar (20) NOT NULL
        constraint ck_pm check (payMethod='CC' OR payMethod='CH'
                                OR payMethod='CA')
);

CREATE TABLE Customer
(
    custCode int primary key,
    custName varchar(30) NOT NULL,
    custSurname varchar(30) NOT NULL,
    custPhone varchar(20) NOT NULL
);

CREATE TABLE Booking
(
    bookCode int primary key,
    bookDt date NOT NULL,
    payCode int NOT NULL,
    custCode int NOT NULL,
    staffNo int NOT NULL,

    Foreign Key (payCode) references payment(payCode),
    Foreign Key (custCode) references Customer(custCode),
    Foreign Key (staffNo) references Staff(staffNo)
);

CREATE TABLE Camping(
    campCode char(3) primary key,
    campName varchar(50) NOT NULL,
    numOfEmp int NOT NULL
);

CREATE TABLE Category
(
    catCode char(1) primary key,
    areaM2 int NOT NULL,
    unitCost numeric(4,2)not NULL
```

```

);

CREATE TABLE Emplacement
(
    campCode char(3) NOT NULL,
    empNo int NOT NULL,
    catCode char(1) NOT NULL,
    Primary Key (campCode,empNo),
    Foreign Key (campCode) references Camping (campCode),
    Foreign Key (catCode) references Category (catCode)
);

CREATE TABLE Rental(
    bookCode int NOT NULL,
    campCode char(3) NOT NULL,
    empNo int NOT NULL,
    startDt date NOT NULL,
    endDt date NOT NULL,
    noPers int NOT NULL,

    Primary Key (bookCode,campCode,empNo,startDt),
    Foreign Key (bookCode) references Booking (bookCode),
    Foreign Key (campCode,empNo) references Emplacement (campCode,empNo)
);

```

4. Διοχέτευση δεδομένων του πίνακα campData στους πίνακες της βάσης.

```

insert into Customer
select distinct custCode,custName,custSurname,custPhone
from campData;

```

```

insert into Staff
select distinct staffNo, staffName, staffSurname
from campData;

```

```

insert into Payment
select distinct payCode,payMethod
from campData;

```

```

insert into Camping
select distinct campCode,campName,numOfEmp
from campData;

```

```

insert into Category
select distinct catCode,areaM2,unitCost
from campData;

```

```

insert into Emplacement
select distinct campCode,empNo,catCode
from campData;

```

```
insert into Booking
select distinct bookCode,bookDt,payCode,custCode,staffNo
      from campData;
```

```
insert into Rental
select distinct bookCode,campCode,empNo,startDt,endDt,noPers
      from campData;
```