

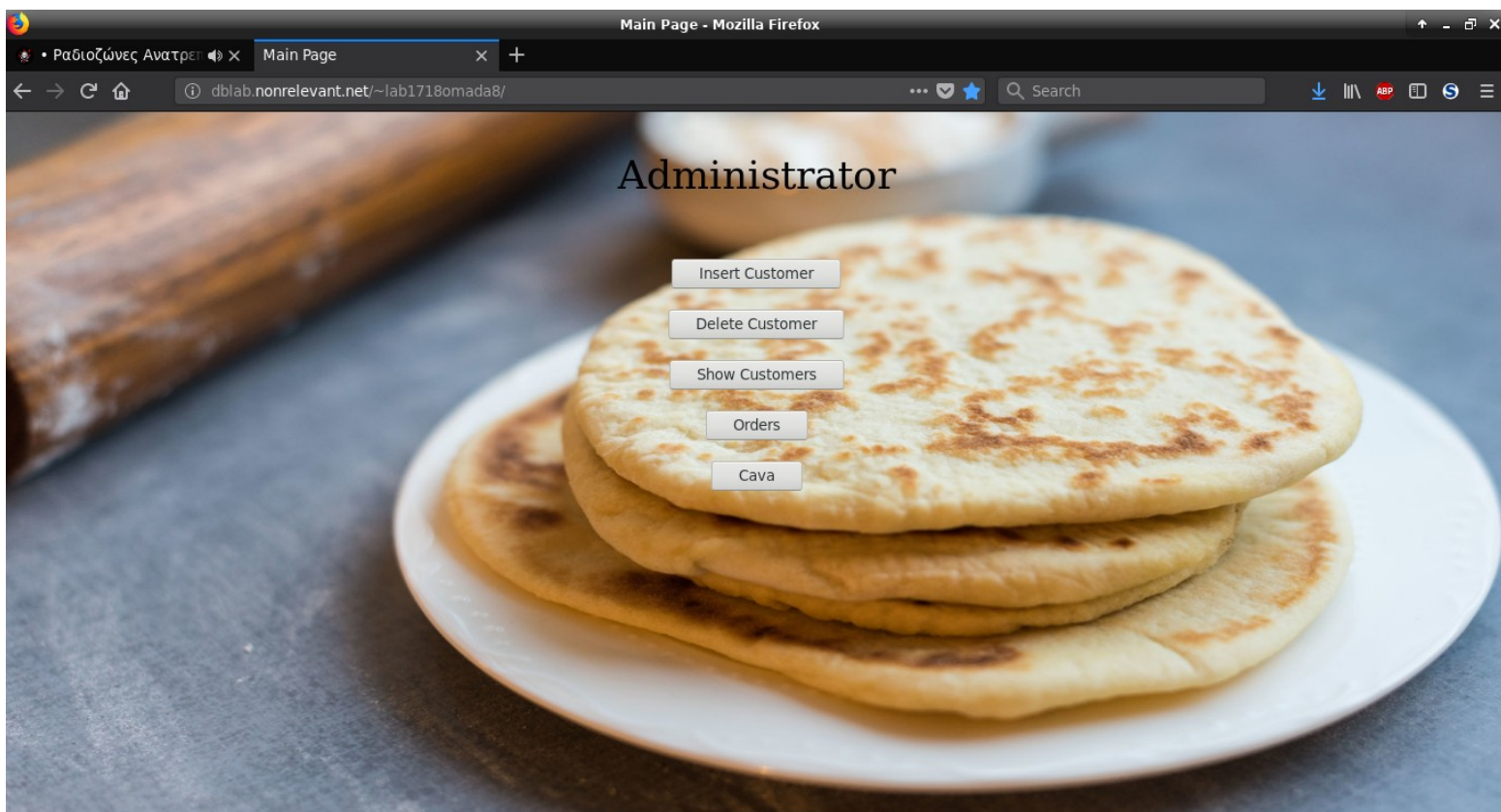
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΤΡΙΤΗ ΕΡΓΑΣΙΑ

ΔΙΑΧΕΙΡΙΣΗ ONLINE ΠΑΡΑΓΓΕΛΙΩΝ ΣΕ ΣΟΥΒΛΑΤΖΙΔΙΚΟ

ΟΜΑΔΑ 8 (lab1718omada8)

27/1/18



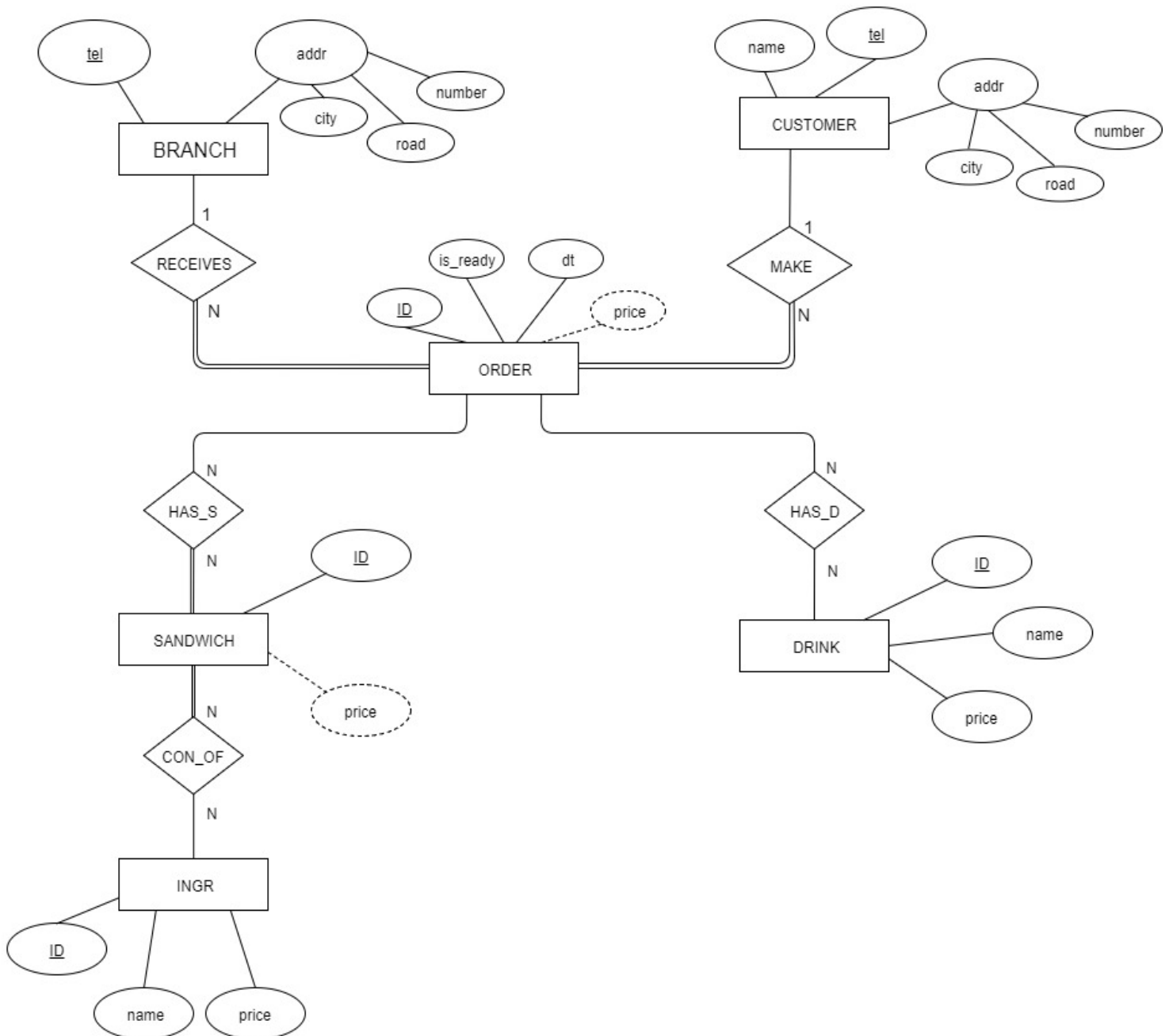
Διονύσιος- Βασίλειος Λυκιαρδόπουλος 56986

Ηλίας Παπαδέας 56989

Χριστόφορος Σπάρταλης 56785

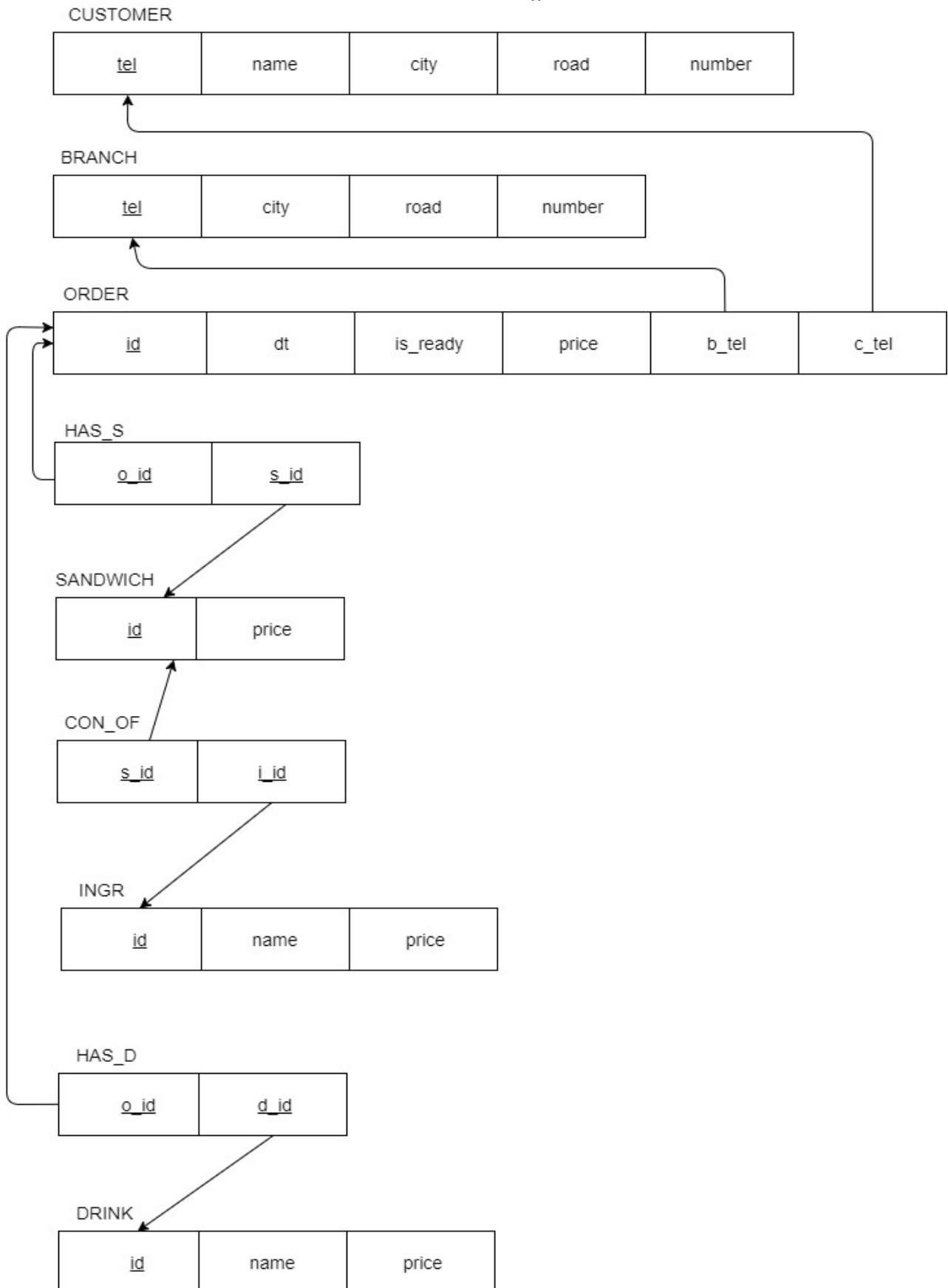
ΠΡΩΤΟ ΜΕΡΟΣ

Διάγραμμα οντοτήτων – συσχετίσεων



Επεξηγηματική παράγραφος: Τα γνωρίσματα price των οντοτήτων SANDWICH και ORDER είναι παραγόμενα. Επίσης, επειδή το όνομα της οντότητας ORDER είναι δεσμευμένη λέξη στην mysql. Στον κώδικα αναφέρεται ως ORDERING.

Σχεσιακό σχήμα



Διαχείριση online παραγγελίας

Ένας **πελάτης** με συγκεκριμένο τηλέφωνο, όνομα και διεύθυνση (πόλη, οδός, αριθμός) μπορεί να παραγγείλει σ' ένα από τα **υποκαταστήματα**, το οποίο επίσης έχει συγκεκριμένο τηλέφωνο και διεύθυνση (πόλη, οδός, αριθμός).

Η **παραγγελία** έχει συγκεκριμένο κωδικό, ημερομηνία, τιμή, καθώς και την πληροφορία αν είναι έτοιμη ή όχι.

Η **παραγγελία** μπορεί να περιέχει **σάντουιτς** ή **ποτό** ή και τα δύο. Το **ποτό** έχει συγκεκριμένο κωδικό, όνομα και τιμή. Το **σάντουιτς** έχει συγκεκριμένο κωδικό, τιμή και συνοδεύεται από **υλικά**. Τα **υλικά** αποτελούνται από μοναδικό κωδικό, όνομα και τιμή.

Μπορεί να υπάρξει **παραγγελία** χωρίς **σάντουιτς**, όμως δεν μπορεί να υπάρξει **σάντουιτς** χωρίς **παραγγελία**.

Το **σάντουιτς** είναι απαραίτητο να περιέχει τουλάχιστον ένα **υλικό**, όμως ένας **πελάτης** μπορεί να παραγγείλει σκέτο κάποιο **υλικό**. Μπορεί να υπάρξει **παραγγελία** χωρίς **ποτό** και το **ποτό** μπορεί να υπάρξει χωρίς **παραγγελία**.

Ο διαχειριστής (administrator) μπορεί να εισάγει ή να διαγράψει μια καταχώρηση πελάτη από το πελατολόγιο. Επίσης μπορεί να βλέπει όλες τις καταχωρήσεις στο πελατολόγιο και να γνωρίζει ανά πάσα στιγμή ποίοι είναι οι τρεις καλύτεροι πελάτες (δηλαδή αυτοί που κάνουν τη μεγαλύτερη κατανάλωση). Τέλος, έχει την εποπτεία όλων των παραγγελιών που έχουν γίνει σε όλα τα καταστήματα που διαθέτει η επιχείρηση.

ΔΕΥΤΕΡΟ ΜΕΡΟΣ

/*DATABASE dblab1718omada8_xs*/

```
create table CUSTOMER (  
  tel bigint(11) not null,  
  name varchar(40) not null,  
  city varchar(40),  
  road varchar(40) not null,  
  number int(4) not null,  
  primary key(tel));
```

```
create table BRANCH (  
  tel bigint(11) not null,  
  city varchar(40),  
  road varchar(40) not null,  
  number int(4) not null,
```

primary key (tel));

```
create table ORDERING (  
  id int not null AUTO_INCREMENT,  
  dt DATETIME,  
  is_ready bool,  
  price decimal(5,2) not null,  
  b_tel bigint(11) not null,  
  c_tel bigint(11) not null,  
  primary key(id),  
  foreign key(b_tel) references BRANCH(tel),  
  foreign key(c_tel) references CUSTOMER(tel));
```

```
create table SANDWICH (  
  id int not null AUTO_INCREMENT,  
  price decimal(5,2) not null,  
  primary key(id));
```

```
create table DRINK (  
  id int not null AUTO_INCREMENT,  
  name varchar(40) not null,  
  price decimal(5,2) not null,  
  primary key(id));
```

```
create table INGR (  
  id int not null AUTO_INCREMENT,  
  name varchar(40) not null,  
  price decimal(3,2) not null,  
  primary key(id));
```

```
create table CON_OF (  
  s_id int,  
  i_id int,  
  primary key (s_id, i_id),  
  foreign key(s_id) references SANDWICH(id),  
  foreign key(i_id) references INGR(id));
```

```
create table HAS_S (  
  o_id int,  
  s_id int,  
  primary key (o_id, s_id),  
  foreign key(s_id) references SANDWICH(id),
```

```
foreign key(o_id) references ORDERING(id));
```

```
create table HAS_D (  
  o_id int,  
  d_id int,  
  primary key (o_id, d_id),  
  foreign key(d_id) references DRINK(id),  
  foreign key(o_id) references ORDERING(id));
```

```
/*DATA dblab1718omada8_xs*/
```

```
SET FOREIGN_KEY_CHECKS=0;
```

```
#BRANCH
```

```
insert into BRANCH  
values (2541058698, 'Xanthi', 'Tsaldari', 58);
```

```
insert into BRANCH  
values (2541069852, 'Xanthi', 'Tsimiski', 103);
```

```
insert into BRANCH  
values (2510666777, 'Kavala', 'Burwnos', 5);
```

```
insert into BRANCH  
values (2535055999, 'Komotini', 'Ag. Gewrgiou', 33);
```

```
#CUSTOMER
```

```
insert into CUSTOMER  
values (6981713158, 'Xristoforos Spartalis', 'Xanthi', 'Karaoli', 6);
```

```
insert into CUSTOMER  
values (6988913688, 'Akis Lukiardopoulos', 'Xanthi', 'Karaoli', 15);
```

```
insert into CUSTOMER  
values (6987568998, 'Hlias Papadeas', 'Xanthi', 'Mprokoumi', 26);
```

```
insert into CUSTOMER  
values (2541078589, 'Xristoforos Spartalis', 'Xanthi', 'Karaoli', 6);
```

```
insert into CUSTOMER  
values (6985689636, 'Hriana Theofilou', 'Komotini', 'Perikleous', 19);
```

```
insert into CUSTOMER  
values (6988913699, 'Elena Papadopoulou', 'Komotini', 'Zaimi', 5);
```

```
insert into CUSTOMER  
values (2535066898, 'Hriana Theofilou', 'Komotini', 'Perikleous', 19);
```

```
insert into CUSTOMER  
values (6978562310, 'Akis Tsoxatzopoulos', 'Kavala', 'Xatzidaki', 67);
```

```
insert into CUSTOMER  
values (2510758699, 'Ntora Mpelogianni', 'Kavala', 'Papandreou', 100);
```

#INGR (MEAT)

```
insert into INGR (name, price)  
values ('kotompeikon', 1.50);
```

```
insert into INGR (name, price)  
values ('souvlaki kotopoulo', 1.30);
```

```
insert into INGR (name, price)  
values ('souvlaki xoirino', 1.30);
```

```
insert into INGR (name, price)  
values ('panseta', 2.30);
```

```
insert into INGR (name, price)  
values ('guros kotopoulo', 2.30);
```

```
insert into INGR (name, price)  
values ('guros xoirino', 2.30);
```

#INGR (WRAP)

```
insert into INGR (name, price)  
values ('pita', 0.00);
```

```
insert into INGR (name, price)
values ('psomaki', 0.00);
```

```
insert into INGR (name, price)
values ('lagana', 0.20);
```

```
insert into INGR (name, price)
values ('dipli pita', 0.20);
```

```
insert into INGR (name, price)
values ('kupriaki', 0.20);
```

```
insert into INGR (name, price)
values ('mpagketa', 0.30);
```

```
insert into INGR (name, price)
values ('tortigia', 0.30);
```

```
#INGR (INGR)
```

```
insert into INGR (name, price)
values ('tomata', 0.00);
```

```
insert into INGR (name, price)
values ('kremmudi', 0.00);
```

```
insert into INGR (name, price)
values ('laxano', 0.00);
```

```
insert into INGR (name, price)
values ('marouli', 0.00);
```

```
insert into INGR (name, price)
values ('patates', 0.00);
```

```
insert into INGR (name, price)
values ('piperia', 0.10);
```

```
insert into INGR (name, price)
values ('kefaloturi', 0.10);
```


#INGR (SAUCE)

```
insert into INGR (name, price)
values ('tzatziki', 0.20);
```

```
insert into INGR (name, price)
values ('turosalata', 0.20);
```

```
insert into INGR (name, price)
values ('paprika', 0.20);
```

```
insert into INGR (name, price)
values ('melitzanosalata', 0.20);
```

```
insert into INGR (name, price)
values ('sauce moustardas', 0.20);
```

```
insert into INGR (name, price)
values ('ouggareza', 0.20);
```

```
insert into INGR (name, price)
values ('giaourti', 0.20);
```

```
insert into INGR (name, price)
values ('sauce kantina', 0.30);
```

#DRINK

```
insert into DRINK (name, price)
values ('xwris', 0.00);
```

```
insert into DRINK (name, price)
values ('nero 500ml', 0.50);
```

```
insert into DRINK (name, price)
values ('coca cola 330ml', 1.20);
```

```
insert into DRINK (name, price)
values ('portokalada 330ml', 1.20);
```

```
insert into DRINK (name, price)
values ('lemonada 330ml', 1.20);
```

```
insert into DRINK (name, price)
values ('soda 330ml', 1.20);
```

```
insert into DRINK (name, price)
values ('amstel 330ml', 1.00);
```

```
insert into DRINK (name, price)
values ('vergina 330ml', 0.80);
```

#ORDERING

```
insert into ORDERING (dt, is_ready, price, b_tel, c_tel)
values ('2017-11-29 20:50:46', 1, 2.70, 2541058698, 2541078589);
```

```
insert into ORDERING (dt, is_ready, price, b_tel, c_tel)
values ('2017-11-29 21:05:32', 1, 2.90, 2541058698, 6987568998);
```

```
insert into ORDERING (dt, is_ready, price, b_tel, c_tel)
values ('2017-11-29 21:30:19', 1, 2.30, 2541058698, 6988913688);
```

```
insert into ORDERING (dt, is_ready, price, b_tel, c_tel)
values ('2017-11-30 13:02:30', 1, 2.30, 2541058698, 6981713158);
```

```
insert into ORDERING (dt, is_ready, price, b_tel, c_tel)
values ('2017-11-30 18:15:27', 0, 2.40, 2535055999, 6988913699);
```

```
insert into ORDERING (dt, is_ready, price, b_tel, c_tel)
values ('2017-11-30 18:20:46', 0, 7.20, 2541058698, 6981713158);
```

#SANDWICH

```
insert into SANDWICH (price) values (2.70);
```

```
insert into SANDWICH (price) values (1.70);
```

```
insert into SANDWICH (price) values (2.30);
```

insert into SANDWICH (price) values (2.30);

insert into SANDWICH (price) values (2.40);

insert into SANDWICH (price) values (2.30);

insert into SANDWICH (price) values (2.40);

#CON_OF

insert into CON_OF values (1, 5);

insert into CON_OF values (1, 9);

insert into CON_OF values (1, 18);

insert into CON_OF values (1, 21);

insert into CON_OF values (2, 1);

insert into CON_OF values (2, 10);

insert into CON_OF values (2, 18);

insert into CON_OF values (3, 6);

insert into CON_OF values (3, 7);

insert into CON_OF values (3, 8);

insert into CON_OF values (4, 6);

insert into CON_OF values (4, 7);

insert into CON_OF values (4, 8);

insert into CON_OF values (5, 5);

insert into CON_OF values (5, 7);

insert into CON_OF values (5, 18);

insert into CON_OF values (5, 20);

insert into CON_OF values (6, 6);

insert into CON_OF values (6, 7);

insert into CON_OF values (6, 18);

insert into CON_OF values (7, 6);

insert into CON_OF values (7, 7);

insert into CON_OF values (7, 18);

insert into CON_OF values (7, 20);

#HAS_S

```
insert into HAS_S (o_id, s_id) values (1, 1);
insert into HAS_S (o_id, s_id) values (2, 2);
insert into HAS_S (o_id, s_id) values (3, 3);
insert into HAS_S (o_id, s_id) values (4, 4);
insert into HAS_S (o_id, s_id) values (5, 5);
insert into HAS_S (o_id, s_id) values (6, 6);
insert into HAS_S (o_id, s_id) values (6, 7);
```

#HAS_D

```
insert into HAS_D (o_id, d_id) values (1, 1);
insert into HAS_D (o_id, d_id) values (2, 3);
insert into HAS_D (o_id, d_id) values (3, 1);
insert into HAS_D (o_id, d_id) values (4, 1);
insert into HAS_D (o_id, d_id) values (5, 1);
insert into HAS_D (o_id, d_id) values (6, 3);
insert into HAS_D (o_id, d_id) values (6, 5);
```

SET FOREIGN_KEY_CHECKS=1;

commit;

/*DATA MINING dblab1718omada8_xs*/

/*1.The customers that are located in Xanthi*/

```
SELECT * FROM CUSTOMER
WHERE city='Xanthi';
```

/*2.The id and the price of the orders that have to be prepared in the branch of Xanthi and where to be delivered*/

```
SELECT o.id, o.price, c.city, c.road, c.number
FROM ORDERING o, BRANCH b, CUSTOMER c
WHERE o.b_tel=b.tel AND o.c_tel=c.tel
AND o.is_ready=0 AND b.city='Xanthi';
```

/*3.Order the orders according to the date and time that they have been made (most recent first)*/

```
SELECT * FROM ORDERING  
ORDER BY dt DESC;
```

/*4.The ingredients of the sandwiches that have to be prepared*/

```
SELECT o.id, s.id, i.name  
FROM ORDERING o, HAS_S h, SANDWICH s, INGR i, CON_OF c  
WHERE o.id=h.o_id AND s.id=h.s_id  
AND s.id=c.s_id AND i.id=c.i_id  
AND o.is_ready=0;
```

/*5.The parts of the orders and their price in the branch of Xanthi (first the ones that aren't ready and most recent)*/

```
SELECT o.id, o.is_ready, o.price, s.id, s.price, d.id, d.price  
FROM ORDERING o, HAS_S hs, SANDWICH s, HAS_D hd, DRINK d,  
BRANCH b  
WHERE o.id=hs.o_id AND s.id=hs.s_id  
AND o.id=hd.o_id AND d.id=hd.d_id  
AND o.b_tel=b.tel  
AND b.city='Xanthi'  
ORDER BY is_ready, dt DESC;
```

ΤΡΙΤΟ ΜΕΡΟΣ

ΕΡΩΤΗΜΑ 1

#(1.a)

/*Εμφανίζει το όνομα και το τηλέφωνο του πελάτη καθώς και το πότε πραγματοποιήθηκε η τελευταία του παραγγελία. Επίσης εμφανίζει και τις καταχωρήσεις στο πελατολόγιο που είναι τελείως ανενεργές. Σε μια δεύτερη περίπτωση, εμφανίζει μόνο όσες καταχωρήσεις από το πελατολόγιο έχουν παραγγείλει έστω και μία φορά.*/

/*1η περίπτωση (left outer join)*/

```
SELECT name, tel, max(dt) as teleutaia_paraggelia  
FROM (CUSTOMER LEFT OUTER JOIN ORDERING ON tel=c_tel)
```

```
GROUP BY tel
ORDER BY teleutaia_paraggelia desc;
```

```
/*2η περίπτωση (right outer join ή join)*/
SELECT name, tel, max(dt) as teleutaia_paraggelia
FROM (CUSTOMER RIGHT OUTER JOIN ORDERING ON tel=c_tel)
GROUP BY tel
ORDER BY teleutaia_paraggelia desc;
```

#(1.β)

/*Εμφανίζει τα ονόματα των 3 πρώτων πελατών που έχουν κάνει τη μεγαλύτερη συνολική κατανάλωση, ταξινομημένα σε φθίνουσα σειρά με βάση τη συνολική κατανάλωση*/

```
SELECT name, sum(price) as sunolikh_katanalwsh
FROM CUSTOMER, ORDERING
WHERE tel=c_tel
GROUP BY name
ORDER BY sunolikh_katanalwsh desc limit 3;
```

/*Εμφανίζει το μέσο κόστος παραγγελίας (μέχρι 2 δεκαδικά) που γίνεται σε κάθε υποκατάστημα. Εμφανίζει και τα καταστήματα στα οποία δεν έχει γίνει κάποια παραγγελία ακόμα.*/

```
SELECT concat (city, ' ', road, ' ', number) as katasthma,
truncate (avg(price), 2) as meso_kostos_paraggelias
FROM (BRANCH LEFT OUTER JOIN ORDERING ON tel=b_tel)
GROUP BY katasthma;
```

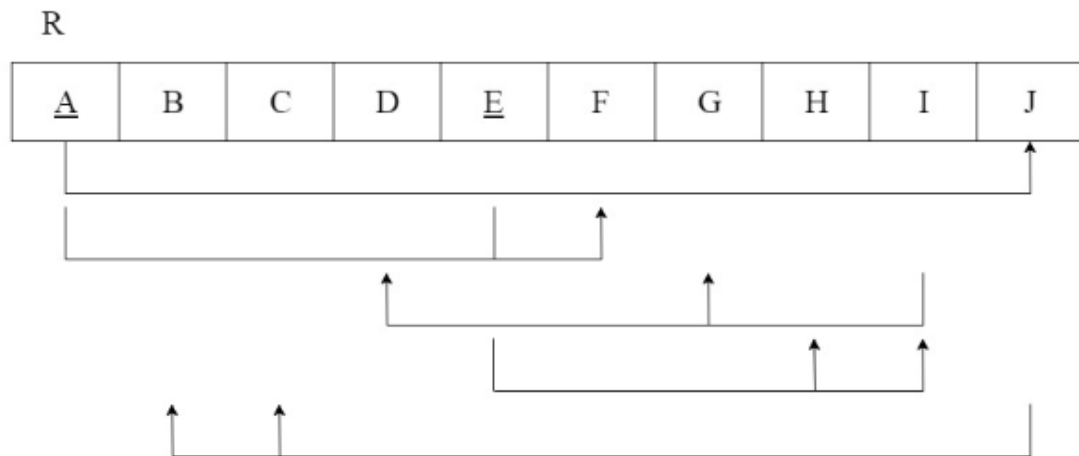
#(1.γ)

/*Εμφανίζει τα ονόματα των πελατών που έχουν παραγγείλει έστω και μία φορά στο μαγαζί και τον συνολικό αριθμό των παραγγελιών, ομαδοποιημένα κατά όνομα και ταξινομημένα κατά φθίνουσα σειρά με βάση τον αριθμό παραγγελίας.*/

```
SELECT name, COUNT(*) as sunolikes_paraggelies
FROM CUSTOMER, ORDERING
WHERE tel=c_tel
GROUP BY name
ORDER BY sunolikes_paraggelies desc;
```

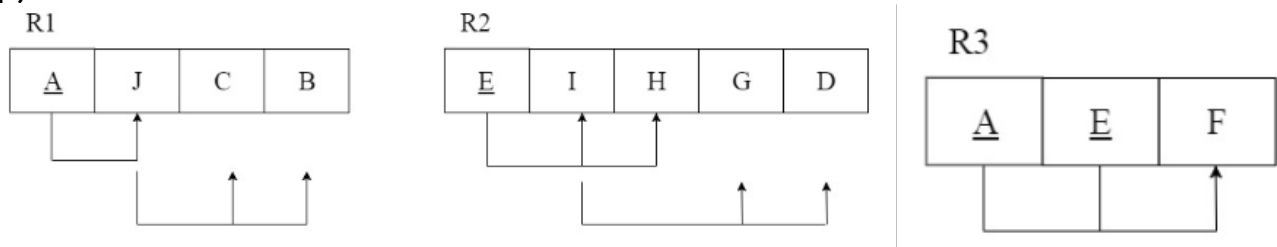
ΕΡΩΤΗΜΑ 2

α)



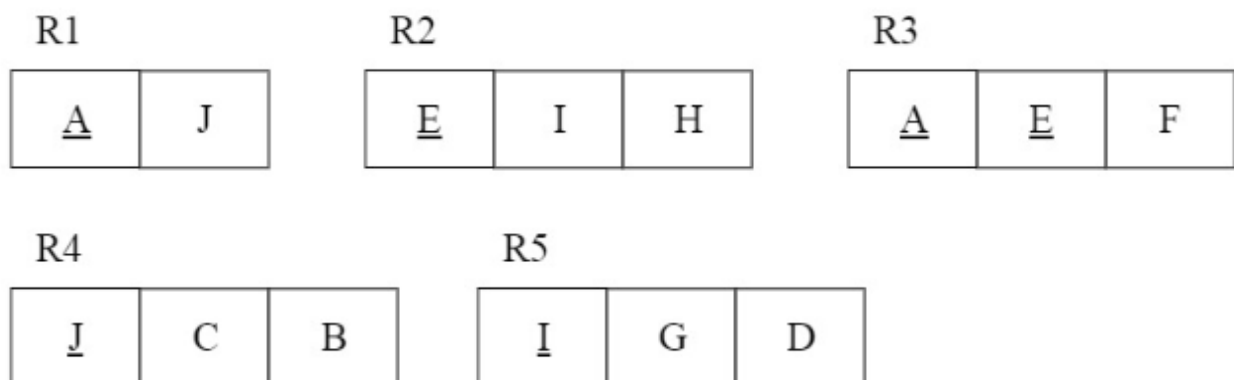
Υποψήφιο Κλειδί: {A, E}⁺ = {A, E, J, C, B, I, H, G, D, F}
 Άρα επαληθεύτηκε η κλειστότητα.

β)



R1 (A, J) R2 (E, I, H, G, D) R3 (A, E, F)

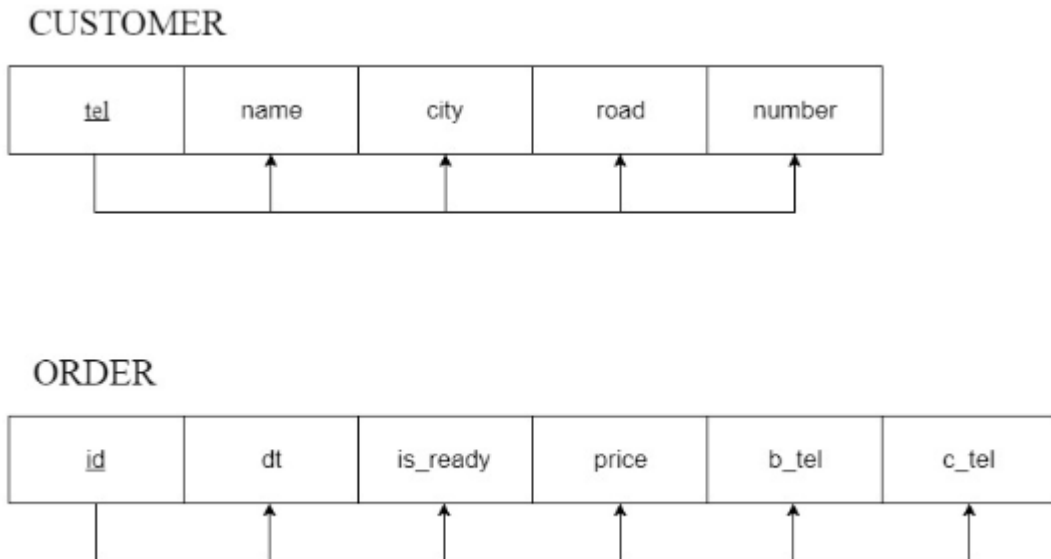
γ)



R1 (A, J) R2 (E, I, H) R3 (A, E, F) R4 (J, C, B) R5(I, G, D)

ΕΡΩΤΗΜΑ 3

α)



Τα κλειδιά που διαλέξαμε συμπίπτουν με αυτά που διαλέξαμε από τις Σ.Ε.

β) Οι δύο πίνακες είναι σε 2NF, αφού έχουν μόνο ένα κλειδί.

γ) Οι πίνακες δεν μπορούν να «σπάσουν» περαιτέρω. Είναι ήδη σε 3NF.

ΕΡΩΤΗΜΑ 4

#(4.a)

/*Ανανεώνονται 5 πίνακες ταυτόχρονα. Ο CON_OF, ο οποίος συνδέει τα υλικά με το σάντουιτς το οποίο συνθέτουν. Ο SANDWICH, ο οποίος περιέχει την πληροφορία για την τιμή του προηγούμενου σάντουιτς. Ο HAS_S που συνδέει το/τα σάντουιτς με την παραγγελία. Ο HAS_D, που συνδέει το/τα ποτά με την παραγγελία. Ο ORDERING που περιέχει τις πληροφορίες της παραγγελίας.*/

DELIMITER |

start transaction;

SET FOREIGN_KEY_CHECKS=0;

insert into CON_OF(s_id, i_id) values (8, 6);

insert into CON_OF(s_id, i_id) values (8, 11);

insert into CON_OF(s_id, i_id) values (8, 18);

insert into CON_OF(s_id, i_id) values (8, 20);

insert into CON_OF(s_id, i_id) values (8, 25);

insert into SANDWICH(price) values (2.80);

insert into HAS_S(o_id, s_id) values (7, 8);

insert into HAS_D(o_id, d_id) values (7, 8);

insert into ORDERING (dt, is_ready, price, b_tel, c_tel)

values ('2017-11-30 19:01:39', 0, 3.60, 2541058698, 6988913688);

SET FOREIGN_KEY_CHECKS=1;


```
commit|  
DELIMITER ;
```

#(4.b)

/*Ευρετήριο για τα ονοματεπώνυμα των πελατών, έτσι ώστε όταν θέλουμε να εντοπίσουμε το ονοματεπώνυμο ενός πελάτη (π.χ. για να του προσφέρουμε μία καλύτερη τιμή ή κάποιο δώρο) να πέρνει λιγότερο χρόνο.*/*

```
CREATE INDEX cust_name ON CUSTOMER (name);
```

/*Ευρετήριο για τα ονόματα των υλικών, έτσι ώστε όταν θέλουμε να εντοπίσουμε κάποιο υλικό (π.χ. για να το διαγράψουμε ή να ανανεώσουμε την τιμή του) να πέρνει λιγότερο χρόνο.*/*

```
CREATE INDEX ingr_name ON INGR (name);
```

#(4.c)

/*Εμφανίζει συγκεντρωμένα τις απαραίτητες πληροφορίες για το ποια παραγγελία πρέπει να ετοιμαστεί που πρέπει να αποσταλεί και σε ποιόν πελάτη (χρησιμοποιείται συχνά στο site)*/*

```
CREATE VIEW order_info_admin as  
SELECT id, dt, price, concat(road, ' ', number, ' ', city) as address, c_tel,  
prosfora(price), is_ready  
FROM ORDERING, CUSTOMER  
WHERE c_tel=tel  
ORDER BY dt desc;
```

/*Η όψη kaluteroi_pelates περιέχει τα ονόματα των 3 πρώτων πελατών που έχουν κάνει τη μεγαλύτερη συνολική κατανάλωση, ταξινομημένα σε φθίνουσα σειρά με βάση τη συνολική κατανάλωση*/*

```
CREATE VIEW kaluteroi_pelates as  
SELECT name, sum(price) as sunolikh_katanalwsh  
FROM CUSTOMER, ORDERING  
WHERE tel=c_tel  
GROUP BY name  
ORDER BY sunolikh_katanalwsh desc limit 3;
```

#(4.d)

/*Stored Procedure: Προσθέτει ή διαγράφει καταχωρήσεις στον πίνακα DRINK (υλοποιείται και στην ιστοσελίδα)*/*

```
DELIMITER |  
CREATE PROCEDURE kava  
(IN action_in varchar(10),  
IN name_in varchar(40),  
IN price_in decimal(5,2))
```

```

BEGIN
  IF action_in='DELETE' THEN
    DELETE FROM DRINK WHERE name=name_in;
  ELSEIF action_in='INSERT' THEN
    INSERT INTO DRINK (name, price) VALUES (name_in, price_in);
  END IF;
END|
DELIMITER ;

```

```

CALL kava('INSERT', 'sourwth', 0.8);
CALL kava('DELETE', 'sourwth', 0.8);

```

/*Stored Function: Εμφανίζει τι extra δώρο θα περιέχεται σε κάποια παραγγελία ανάλογα με το κόστος της (υλοποιείται και στην ιστοσελίδα)*/

```

DELIMITER |
CREATE FUNCTION prosfora (price decimal(5,2))
RETURNS varchar(20) DETERMINISTIC
BEGIN
  DECLARE extra varchar(20);
  IF price >= 5 THEN
    SET extra="coca cola 500ml";
  ELSEIF price >= 2.5 THEN
    SET extra="coca cola 330ml";
  ELSE
    SET extra="---";
  END IF;
  RETURN (extra);
END|
DELIMITER ;

```

```

SELECT id, price, prosfora(price) FROM ORDERING;

```

#(4.e)

/*Όταν αλλάζει τηλέφωνο ένας πελάτης, ανανεώνεται και ο πίνακας των παραγγελιών (ORDERING)*/

```

DELIMITER |
CREATE TRIGGER change_tel
AFTER UPDATE ON CUSTOMER
FOR EACH ROW
BEGIN
  UPDATE ORDERING
  SET c_tel = NEW.tel
  WHERE c_tel = OLD.tel;
END|
DELIMITER ;

```