

## Βάσεις Δεδομένων - Εργασία

### Database Driven Website - Moving On

Βασιλική Ειρήνη Μηλιούδη Σίσκου	XXXXXX
Θανάσης Τσιρίκας	XXXXXX
Παναγιώτα Νεφέλη Κουλιούμπα	XXXXXX

### Περίληψη

Ο σχεδιασμός της βάσης δεδομένων εστιάζει στην αποθήκευση πληροφοριών για μια μικρή μεταφορική εταιρεία. Μετά από ανάλυση απαιτήσεων, δημιουργήθηκαν τα διαγράμματα οντοτήτων-συσχετίσεων (Ο-Σ) και το σχεσιακό, τα οποία αποτέλεσαν τη βάση για την ανάπτυξη του κώδικα. Η εφαρμογή των παραπάνω θα έχει ως αποτέλεσμα τη δημιουργία ιστοσελίδας, η οποία θα ενσωματώνει μέρος της παραπάνω βάσης δεδομένων.

# Περιεχόμενα

Μέρος 1: Σχεδιασμός Βάσης Δεδομένων . . . . .	1
1.1 Διάγραμμα Οντοτήτων - Συσχετίσεων . . . . .	2
Μέρος 2: Ανάπτυξη Βάσης Δεδομένων . . . . .	4
2.1 Σχεσιακό Διάγραμμα . . . . .	4
2.2 Ερωτήματα MySQL . . . . .	5
Μέρος 3: Τελική Υλοποίηση Βάσης Δεδομένων	6
3.1 Άσκηση. . . . .	6
3.2 Κανονικοποίηση. . . . .	7
3.3 Ερωτήματα MySQL . . . . .	7
3.4 Ευρετήρια. . . . .	8
3.5 Όψεις. . . . .	9
3.6 Αποθηκευμένη Διαδικασία. . . . .	10
3.7 Αποθηκευμένη Συνάρτηση. . . . .	11
3.8 Συνναλαγή . . . . .	11
3.9 Σκανδάλη. . . . .	13
Κώδικας: Τελική Υλοποίηση. . . . .	14
Δημιουργία Βάσης Δεδομένων . . . . .	14
Εισαγωγή Δείγματος Δεδομένων . . . . .	17
Βιβλιογραφία . . . . .	22

# Μέρος 1

## Σχεδιασμός Βάσης Δεδομένων

Ο κάθε γραμματέας **διαχειρίζεται** (*manages*) οδηγούς και **καταγράφει/επεξεργάζεται** (*records*) παραγγελίες. Κάθε πελάτης **καταχωρεί** (*places*) τουλάχιστον μία παραγγελία και **πραγματοποιεί** (*makes*) την αντίστοιχη πληρωμή. Η παραγγελία **έχει** (*has*) την αντίστοιχη πληρωμή και μπορεί να **περάσει** (*is going into*) στο στάδιο της παράδοσης. Ένας οδηγός μπορεί να **παραδώσει** (*carries out*) την παραγγελία. Ένα φορτηγό μπορεί να **μεταφέρει** (*transfers*) μια παράδοση.

Οι βασικές οντότητες που θα περιλαμβάνονται στη βάση δεδομένων είναι:

Για κάθε γραμματέα (SECRETARY) θα αποθηκεύονται:

- αριθμός ταυτότητας (*ID*)
- όνομα (*first name*)
- επίθετο (*last name*)
- τηλέφωνο (*phone*)
- όνομα χρήστη (*username*)
- κωδικός πρόσβασης (*password*)

### Πρωτεύον Κλειδί: ID

Για κάθε οδηγό (DRIVER) θα αποθηκεύονται:

- αριθμός ταυτότητας (*ID*)
- όνομα (*first name*)
- επίθετο (*last name*)
- τηλέφωνο (*phone*)

### Πρωτεύον Κλειδί: ID

Για κάθε όχημα (TRUCK) θα αποθηκεύονται:

- αριθμός πινακίδας (*ID*)
- έτος αγοράς (*purchase year*)
- έτος παραγωγής (*production year*)
- χωρητικότητα (*capacity*)
- εταιρεία παραγωγής (*manufacturer*)

### Πρωτεύον Κλειδί: ID

Για κάθε πελάτη (CLIENT) θα αποθηκεύονται:

- αριθμός ταυτότητας (*ID*)
- όνομα (*first name*)

- επίθετο (*last name*)
- τηλέφωνο (*phone*)

**Πρωτεύον Κλειδί: ID**

Για κάθε πληρωμή (PAYMENT) θα αποθηκεύονται:

- αριθμός πληρωμής (*ID*)
- ημερομηνία (*date*)
- μέθοδος πληρωμής (*method*)
- ποσό πληρωμής (*amount*)

**Πρωτεύον Κλειδί: ID**

**Ξένο Κλειδί: orderID, clientID**

Για κάθε παραγγελία (ORDER) θα αποθηκεύονται:

- αριθμός παραγγελίας (*ID*)
- τιμή (*price*)
- κατάσταση (*status*)

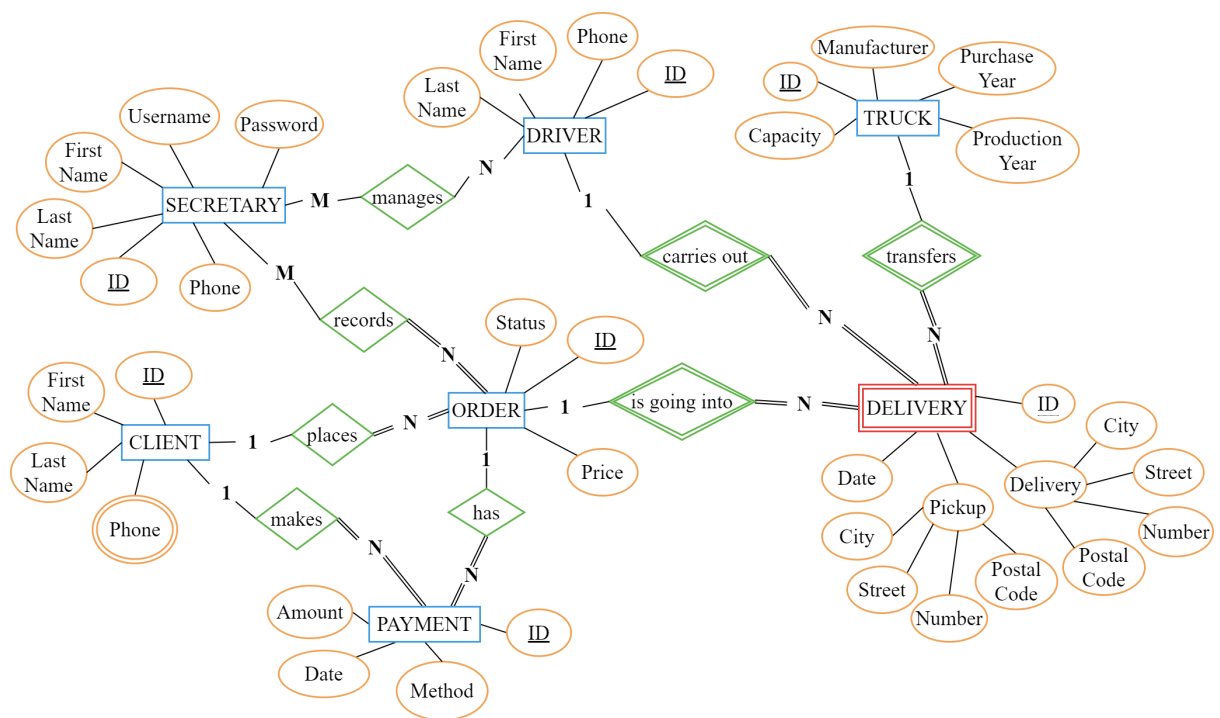
**Πρωτεύον Κλειδί: ID**

Για κάθε παράδοση (DELIVERY) θα αποθηκεύονται:

- αριθμός παράδοσης (*ID*)
- ημερομηνία παράδοσης (*date*)
- πληροφορίες παραλαβής (*pick up*) και παράδοσης (*delivery*)
- Τόσο η παράδοση (*delivery*) όσο και η παραλαβή (*pick up*) αποτελούνται από:
  - ταχυδρομικό κώδικα (*postal code*)
  - πόλη (*city*)
  - οδό (*street*)
  - αριθμό (*number*)

**Πρωτεύον Κλειδί: orderID, ID**

Η οντότητα DELIVERY εξετάται(μη ισχυρή οντότητα) από την οντότητα ORDER καθώς δεν υφίστανται χωρίς να έχει προηγηθεί η ύπαρξη μιας παραγγελίας. Η ιδιότητα phone του CLIENT είναι πλειότιμο, δηλαδή ένας πελάτης μπορεί να έχει παραπάνω από ένα τηλέφωνο. Τέλος, η ιδιότητα amount της PAYMENT είναι παραγόμενο μέγεθος της ιδιότητας price της ORDER.



Σχήμα 1.1: Διάγραμμα Οντοτήτων - Συσχετίσεων

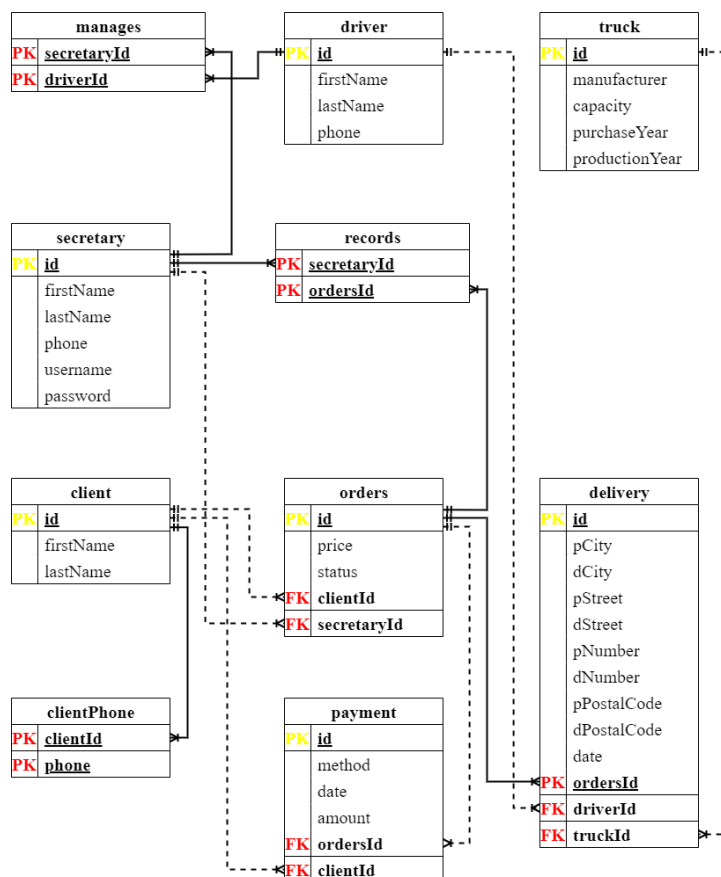
## Μέρος 2

# Ανάπτυξη Βάσης Δεδομένων

Οι πληροφορίες που δεν απεικονίζονται στο διάγραμμα είναι οι εξής:

Η ιδιότητα status μπορεί να πάρει τις τιμές 'Received', 'Cancel', 'Done' και η ιδιότητα method μπορεί να πάρει τις τιμές 'Card' και 'Cash'.

### 2.1 Σχεσιακό Διάγραμμα



Σχήμα 2.1: Σχεσιακό Διάγραμμα

## 2.2 Ερωτήματα MySQL

```

1  ----- DATA ANALYSIS -----
2
3  -- 1.) List the following details of each employee:
4  ----- employee number, last name, first name, sex, and salary.
5
6  SELECT concat(client.firstName, ' ', client.lastName)
7  AS 'Client Name', orders.id AS 'Expected Orders'
8  FROM client, orders
9  WHERE orders.clientId = client.id
10         AND orders.status = 'Received';
11
12 SELECT concat(client.firstName, ' ', client.lastName)
13 AS 'Client Name', payment.method
14 FROM client, payment
15 ORDER BY payment.method DESC;
16
17 SELECT *
18 FROM client
19     JOIN payment ON client.id = payment.clientId;
20
21 SELECT concat(client.firstName, ' ', client.lastName)
22 AS 'Client Name', payment.method, payment.amount, payment.date
23 FROM client
24 JOIN payment ON client.id = payment.clientId
25 WHERE payment.date BETWEEN '2023-05-05' AND '2023-09-09'
26 ORDER BY DATE;
27
28 SELECT orders.id AS 'Orders'
29 FROM orders
30 WHERE status = 'Received'
31 UNION
32 SELECT delivery.id AS 'Deliveries'
33 FROM delivery
34 WHERE dCity = 'Xanthi';

```

## Μέρος 3

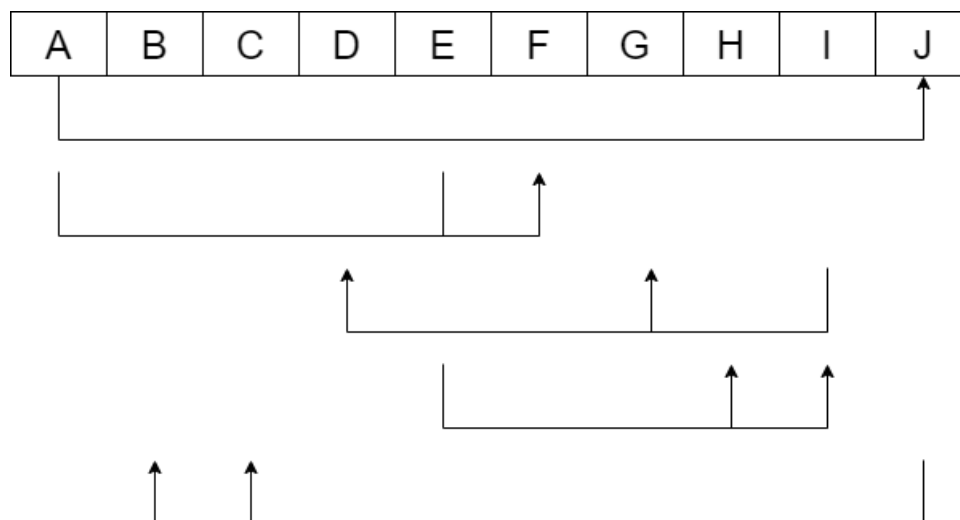
# Τελική Υλοποίηση Βάσης Δεδομένων

### 3.1 Άσκηση

Καθολική Σχέση:  $P = A, B, C, D, E, F, G, H, I, J$

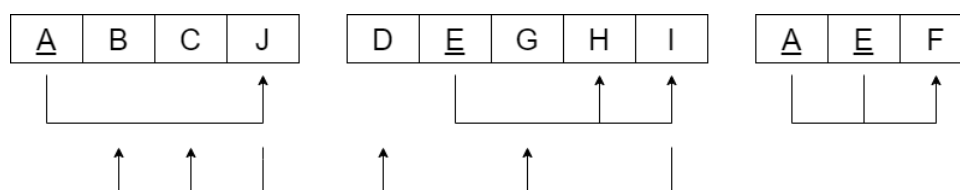
Σύνολο Συναρτησιακών Εξαρτήσεων:  $\{\{A\} \rightarrow \{J\}, \{E, A\} \rightarrow \{F\}, \{I\} \rightarrow \{G, D\}, \{E\} \rightarrow \{I, H\}, \{J\} \rightarrow \{C, B\}\}$

1.



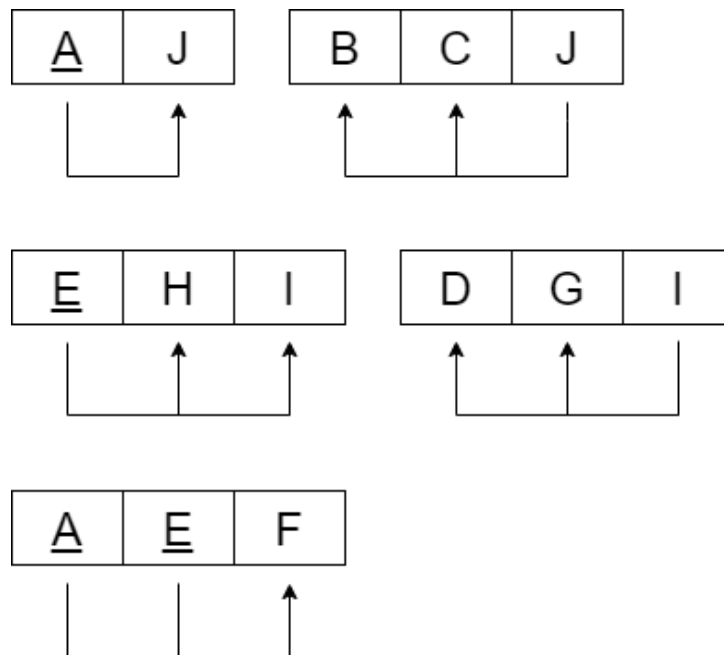
Εφαρμόζουμε το θεώρημα που λέει ότι η κλειστότητα του κλειστού καθορίζει όλα τα γνώ-  
ρίσματα. Τα  $B, C, D, F, G, H, I$  και  $J$  απορρίπτονται γιατί εξαρτώνται από κάποιο άλλο  
γνώρισμα και επομένως αντί για αυτά μπορούμε να χρησιμοποιήσουμε το γνώρισμα από το  
οποίο εξαρτώνται. Τελικά προκύπτει ότι το κλειδί της  $P$  θα είναι το:  $\{A, E\}^+$ .

2. Να σπάσετε την  $R$  σε σχέσεις που να βρίσκονται σε 2NF



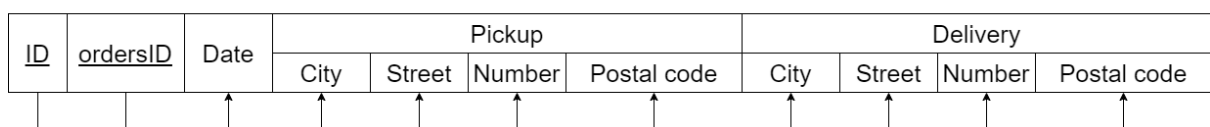
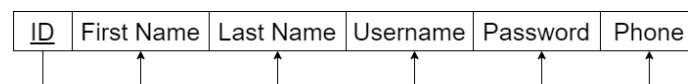


3. στη συνέχεια σε 3NF.



## 3.2 Κανονικοποίηση

Οι δύο μεγαλύτεροι σε αριθμό γνωρισμάτων πίνακες της βάσης είναι SECRETARY και DELIVERY. Το σχήμα υποδεικνύει ότι τα κλειδιά που επιλέξαμε συμπίπτουν με αυτά που προκύπτουν από τις Σ.Ε. Οι πίνακες είναι σε 2NF και 3NF καθώς κάθε χαρακτηριστικό τους που δεν είναι κλειδί είναι συναρτησιακά εξαρτώμενο από το πρωτεύον κλειδί.



## 3.3 Ερωτήματα MySQL

```

1 -----# Queries 2-----
2 USE omada;
3
4 -- Both use JOIN with different results --
5 -- #1 Total number of orders that have delivery
6 SELECT (SELECT COUNT(*) FROM orders INNER JOIN delivery
7         ON delivery.ordersID = orders.id) AS "Total Number Rows",

```

```

8 orders.id AS "Order ID", delivery.id AS "Delivery ID",
9 status AS Status, price AS Price,
10 concat(dCity, ", ", dStreet, ", ", dNumber, ", ", dPostalCode)
11 AS "Delivery Destination",
12 concat(pCity, ", ", pStreet, ", ", pNumber, ", ", pPostalCode)
13 AS "Pick-Up Destination"
14 FROM orders
15     INNER JOIN delivery ON delivery.ordersID = orders.id
16 ORDER BY CAST(orders.id AS SIGNED) ASC, date DESC,
17 pCity ASC, dCity ASC;
18
19 -- #1 Total number of orders with and without delivery
20 SELECT (SELECT COUNT(*) FROM orders LEFT JOIN delivery
21 ON delivery.ordersID = orders.id) AS "Total Number Rows",
22 orders.id AS "Order ID", delivery.id AS "Delivery ID",
23 status AS Status, price AS Price,
24 concat(dCity, ", ", dStreet, ", ", dNumber, ", ", dPostalCode)
25 AS "Delivery Destination",
26 concat(pCity, ", ", pStreet, ", ", pNumber, ", ", pPostalCode)
27 AS "Pick-Up Destination"
28 FROM orders
29     LEFT JOIN delivery ON delivery.ordersID = orders.id;
30
31 -- #2 Average price of orders
32 SELECT avg(price) AS "Average Price"
33 FROM orders;
34
35 -- #3 Most recent delivery date
36 SELECT MAX(date) AS "Average date"
37 FROM delivery;
38
39 -- #4 Group by method in payment entity
40 SELECT method AS Method, count(*) AS "Total Payments"
41 FROM payment
42 GROUP BY method
43 ORDER BY method ASC, count(*) DESC;
44
45 -- #5 Group by status in orders entity and using having command
46 SELECT status, count(*) AS "Total Orders"
47 FROM orders
48 GROUP BY status
49 HAVING avg(price) > 300 AND status ="Received";

```

### 3.4 Ευρετήρια

Η δημιουργία ευρετηρίου για τα ονόματα της/του γραμματέα, της/του πελάτη και της/του οδηγού πιθανόν να είναι χρήσιμα για γρήγορη και αποτελεσματική αναζήτηση σε κάποια διεπαφή.

Η δημιουργία ευρετηρίου για τις διευθύνσεις παραλαβής και παράδοσης μπορεί να διευκολύνει

το προσωπικό της εταιρείας, όπως οι οδηγοί και οι γραμματείς, στον εντοπισμό και τη διαχείριση των παραδόσεων.

```

1  ----- CREATE INDEXES -----
2  USE omada8;
3
4  ----- #1 create secretary full name index -----
5  CREATE INDEX secretary_fullName_idk
6  ON secretary(firstName, lastName);
7
8  ----- #2 create driver full name index -----
9  CREATE INDEX driver_fullName_idk ON driver(firstName, lastName);
10
11 ----- #3 create client full name index -----
12 CREATE INDEX client_fullName_idk ON client(firstName, lastName);
13
14 ----- #4 create delivery address index -----
15 CREATE INDEX delivery_dAddress_idk
16 ON delivery(dCity, dStreet, dNumber);
17
18 ----- #5 create pick-up address index -----
19 CREATE INDEX delivery_pAddress_idk
20 ON delivery(pCity, pStreet, pNumber);

```

### 3.5 Όψεις

Η πρώτη όψη δημιουργεί έναν εικονικό πίνακα που εμφανίζει τον αριθμό παραγγελίας, τον αριθμό παράδοσης, το όνομα του πελάτη, την διεύθυνση παραλαβής, την διεύθυνση παράδοσης, τον οδηγό που θα πραγματοποιήσει την παράδοση καθώς και τον αριθμό πινακίδας του εκάστοτε φορτηγού και ταξινομείτε με βάση τον αριθμό παραγγελίας. Θα μπορούσε να χρησιμοποιηθεί στην σελίδα της διεπαφής για να εμφανίσει τις σημαντικότερες πληροφορίες της παραγγελίας.

Η δεύτερη όψη δημιουργεί έναν εικονικό πίνακα που εμφανίζει τον αριθμό παράδοσης και την ημερομηνία αποστολής που παραδόθηκαν τον τελευταίο χρόνο ταξινομημένα με βάση την ημερομηνία σε φθίνουσα σειρά. Θα μπορούσε να χρησιμοποιηθεί σε κάποια σελίδα της διεπαφής για να εμφανίσει τις πρόσφατες παραδόσεις.

```

1  -----# Views -----
2  USE omada8;
3
4  -- #1 create orders table with order by order id
5  CREATE OR REPLACE VIEW orders_table_view AS
6  SELECT orders.id AS "Order ID",
7  delivery.id AS "Delivery ID",
8  concat(client.firstName, ' ', client.lastName) AS "Client Name",
9  concat(pCity, ", ", pStreet, ", ", pNumber) AS "Pick-Up Dest",
10 concat(dCity, ", ", dStreet, ", ", dNumber) AS "Delivery Dest",
11 concat(driver.firstName, ' ', driver.lastName)
12 AS "Driver Full Name", truck.id AS "Truck ID"

```

```

13 FROM delivery
14     INNER JOIN orders ON delivery.ordersId = orders.id
15     INNER JOIN driver ON delivery.driverId = driver.id
16     INNER JOIN truck ON delivery.truckID = truck.id
17     INNER JOIN client ON orders.clientID = client.id
18 ORDER BY CAST(orders.id AS SIGNED) ASC;
19
20 -- To show the results of the view
21 SELECT * FROM orders_table_view;
22
23 -----
24
25 -- #2 create recent deliveries table order by date
26 CREATE or REPLACE VIEW recent_deliveries_table_view AS
27 SELECT delivery.id AS 'Recent Deliveries', date
28 FROM delivery
29 WHERE date BETWEEN NOW() - INTERVAL 12 MONTH AND NOW()
30 ORDER BY delivery.date DESC;
31
32 -- To show the results of the view
33 SELECT * FROM recent_deliveries_table_view;

```

### 3.6 Αποθηκευμένη Διαδικασία

Η συγκεκριμένη αποθηκευμένη διαδικασία ανανεώνει την κατάσταση προϋπάρχουσας παραγγελίας.

```

1 ----- Procedure -----
2 USE omada8;
3
4 -- Task: Create a procedure to update existing order status
5 -- in the orders table
6 DELIMITER !!
7 DROP PROCEDURE IF EXISTS UpdateOrderStatus;
8 CREATE PROCEDURE UpdateOrderStatus
9 (IN orderIdParam VARCHAR(10),
10 IN orderStatusParam enum('Received', 'Done', 'Cancel'))
11 BEGIN
12     SET @orderId := (SELECT id
13                     FROM orders
14                     WHERE orders.id = orderIdParam);
15     IF @orderId THEN
16         UPDATE orders SET status = orderStatusParam
17         WHERE id = orderIdParam;
18         SELECT concat("Order ", orderIdParam, ' is updated.')
19         AS Confirmation;
20     ELSE
21         SELECT
22         CONCAT("There is no order with id ", orderIdParam)

```

```

23         AS "Confirmation";
24     END IF;
25 END !!
26 DELIMITER ;
27
28 -- To show the results of the procedure
29 CALL UpdateOrderStatus(11, "Done");
30
31 -- To verify tables are accurate after procedure
32 SELECT status FROM orders WHERE id = 11;

```

### 3.7 Αποθηκευμένη Συνάρτηση

Η συγκεκριμένη αποθηκευμένη συνάρτηση καταμετρά τον συνολικό αριθμό των παραδόσεων που έχει πραγματοποιήσει ένας οδηγός.

```

1  ----- Procedure -----
2  USE omada8;
3
4  -- Task: Create a function to retrieve the total number of
5  -- deliveries for a specific driver
6  DELIMITER !
7  DROP FUNCTION IF EXISTS numberOfDeliveries;
8  CREATE FUNCTION numberOfDeliveries(driverLastName VARCHAR(30))
9  RETURNS INT
10 DETERMINISTIC
11 BEGIN
12     DECLARE deliveryCount INT;
13
14     SELECT COUNT(*)
15     INTO deliveryCount
16     FROM delivery
17     INNER JOIN driver ON delivery.driverId = driver.id
18     WHERE driverLastName = driver.lastName;
19
20     RETURN deliveryCount;
21 END !
22 DELIMITER ;
23
24 -- To show the results of the procedure
25 SELECT numberOfDeliveries("Aggelidis")
26 AS "The Total Number of Deliveries";

```

### 3.8 Συνναλαγή

```

1  ----- Transaction -----
2

```

```

3 DROP PROCEDURE IF EXISTS test;
4
5 DELIMITER $$
6 CREATE PROCEDURE test()
7 BEGIN
8     DECLARE err TINYINT DEFAULT 0;
9     DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET err = 1;
10
11     START TRANSACTION;
12
13     INSERT INTO orders
14         (id, price, status, secretaryID, clientID)
15     VALUES
16         (41, 300, 'Received', 6, 7);
17
18     SAVEPOINT point1;
19
20     INSERT INTO delivery
21         (id, pCity, pStreet, pNumber, pPostalCode, dCity,
22         dStreet, dNumber, dPostalCode, date, ordersID,
23         driverID, truckID)
24     VALUES
25         (20, 'Komotini', 'Agios Dimitriou Street', 28, 69100,
26         'Alexandroupoli', 'Old Patras Street', 75, 69131,
27         '2023-06-21', 41, 1, 'ZFK-5678');
28
29     IF err = 1 THEN
30         ROLLBACK TO SAVEPOINT point1;
31         SELECT 'An error occurred' AS message;
32     ELSE
33         COMMIT;
34         SELECT 'OK' AS message;
35     END IF;
36
37 END$$
38 DELIMITER ;
39
40 CALL test();
41
42 -----
43 #verify tables are accurate
44 SELECT * FROM delivery WHERE id=20;
45
46 #verify tables are accurate
47 SELECT * FROM orders WHERE id=41;

```

### 3.9 Σκανδάλη

Η δημιουργία αυτού του trigger ανανεώνει το κόστος(amount) της πληρωμής όταν πραγματοποιείται εισαγωγή της παραγγελίας και πιο συγκεκριμένα η τιμή(price). Πρακτικά, αυτό σημαίνει ότι ο πελάτης καταχωρεί ένα ποσό προκαταβολής, το οποίο προστίθεται στο συνολικό κόστος της παραγγελίας.

```

1  ----- Trigger -----
2  USE omada8;
3
4  DELIMITER //
5  CREATE TRIGGER updateOrderPrice
6  AFTER INSERT ON payment
7  FOR EACH ROW
8  BEGIN
9      UPDATE orders
10     SET price = price + NEW.amount
11     WHERE id = NEW.orderId;
12 END//
13 DELIMITER ;
14
15 ----- #verify tables are accurate -----
16 INSERT INTO orders (id, secretaryID, clientID)
17 VALUES (18, 9, 4);
18
19 INSERT INTO payment (id, amount, date, clientID, orderID)
20 VALUES (18, 50.00, '2023-01-12', 2, 18);
21
22 SELECT * FROM orders WHERE id = 18;

```

# Κώδικας

## Τελική Υλοποίηση

### Δημιουργία Βάσης Δεδομένων

```
1  ----- SQL DATABASE ----- Omada8
2
3  ----- CREATE DATABASE -----
4
5  CREATE DATABASE omada8
6      CHARSET 'utf8mb4' COLLATE 'utf8mb4_unicode_ci';
7
8  USE omada8;
9
10 ----- CREATE TABLES -----
11
12 ----- #1 create secretary table [ENTITY] -----
13 CREATE TABLE secretary(
14     id VARCHAR(50) NOT NULL UNIQUE,
15     firstName VARCHAR(50) NOT NULL,
16     lastName VARCHAR(50) NOT NULL,
17     phone BIGINT NULL DEFAULT NULL,
18     username VARCHAR(50) NOT NULL,
19     password VARCHAR(200) NOT NULL,
20
21     CONSTRAINT pk_secretary PRIMARY KEY(id)
22 );
23
24 ----- #2 create driver table [ENTITY] -----
25 CREATE TABLE driver(
26     id VARCHAR(50) NOT NULL UNIQUE,
27     firstName VARCHAR(50) NOT NULL,
28     lastName VARCHAR(50) NOT NULL,
29     phone BIGINT DEFAULT NULL,
30
31     CONSTRAINT pk_driver_id PRIMARY KEY(id)
32 );
33
34 ----- #3 create truck table [ENTITY] -----
35 CREATE TABLE truck(
```



```

36     id VARCHAR(50) NOT NULL UNIQUE,
37     manufacturer VARCHAR(50) DEFAULT NULL,
38     capacity VARCHAR(10) DEFAULT NULL,
39     purchaseYear YEAR DEFAULT NULL,
40     productionYear YEAR DEFAULT NULL,
41
42     CONSTRAINT pk_truck_id PRIMARY KEY(id)
43 );
44
45 ----- #4 create client table [ENTITY] -----
46 CREATE TABLE client(
47     id VARCHAR(50) NOT NULL UNIQUE,
48     firstName VARCHAR(50) NOT NULL,
49     lastName VARCHAR(50) NOT NULL,
50
51     CONSTRAINT pk_client_id PRIMARY KEY(id)
52 );
53
54 ----- #5 create clientPhone table [RELATIONSHIP] -----
55 CREATE TABLE clientphone(
56     clientID VARCHAR(50) NOT NULL,
57     phone BIGINT NOT NULL UNIQUE,
58
59     CONSTRAINT pk_clientPhone PRIMARY KEY(clientID, phone),
60     CONSTRAINT fk_clientPhone_clientId FOREIGN KEY(clientID)
61     REFERENCES client(id) ON DELETE CASCADE ON UPDATE CASCADE
62 );
63
64 ----- #6 create orders table [ENTITY] -----
65 CREATE TABLE orders(
66     id VARCHAR(50) NOT NULL UNIQUE,
67     price DECIMAL(5, 2) NOT NULL DEFAULT 0.00,
68     status ENUM('Received', 'Done', 'Cancel') NOT NULL
69     DEFAULT 'Received',
70
71     secretaryID VARCHAR(50) NOT NULL,
72     clientID VARCHAR(50) NOT NULL,
73
74     CONSTRAINT pk_orders PRIMARY KEY(id),
75     CONSTRAINT fk_orders_secretaryID FOREIGN KEY(secretaryID)
76     REFERENCES secretary(id) ON DELETE NO ACTION ON UPDATE
77     CASCADE,
78     CONSTRAINT fk_orders_clientID FOREIGN KEY(clientID)
79     REFERENCES client(id) ON DELETE CASCADE ON UPDATE CASCADE
80 );
81
82
83 ----- #7 create payment table [ENTITY] -----
84 CREATE TABLE payment(
85     id VARCHAR(50) NOT NULL UNIQUE,
86     amount DECIMAL(5, 2) NOT NULL,

```

```

87     method ENUM('Cash', 'Card') NOT NULL DEFAULT 'Cash',
88     date DATE NOT NULL,
89
90     clientID VARCHAR(50) NOT NULL,
91     orderID VARCHAR(50) NOT NULL UNIQUE,
92
93     CONSTRAINT pk_payment PRIMARY KEY(id),
94     CONSTRAINT fk_payment_clientID FOREIGN KEY(clientID)
95     REFERENCES client(id) ON DELETE CASCADE ON UPDATE CASCADE,
96     CONSTRAINT fk_payment_orderID FOREIGN KEY(orderID)
97     REFERENCES orders(id) ON DELETE CASCADE ON UPDATE CASCADE
98 );
99
100 ----- #8 create delivery table [ENTITY] -----
101 CREATE TABLE delivery(
102     id VARCHAR(50) NOT NULL UNIQUE,
103     date DATE NOT NULL,
104     pCity VARCHAR(50) NOT NULL,
105     pStreet VARCHAR(50) NOT NULL,
106     pNumber BIGINT NOT NULL,
107     pPostalCode BIGINT NULL DEFAULT NULL,
108     dCity VARCHAR(50) NOT NULL,
109     dStreet VARCHAR(50) NOT NULL,
110     dNumber BIGINT NOT NULL,
111     dPostalCode BIGINT NULL DEFAULT NULL,
112
113     ordersID VARCHAR(50) NOT NULL UNIQUE,
114     driverID VARCHAR(50) NOT NULL,
115     truckID VARCHAR(50) NOT NULL,
116
117     CONSTRAINT pk_delivery PRIMARY KEY(id, ordersID),
118     CONSTRAINT fk_delivery_ordersID FOREIGN KEY(ordersID)
119     REFERENCES orders(id) ON DELETE CASCADE ON UPDATE CASCADE,
120     CONSTRAINT fk_delivery_driverID FOREIGN KEY(driverID)
121     REFERENCES driver(id) ON DELETE NO ACTION ON UPDATE CASCADE,
122     CONSTRAINT fk_delivery_truckID FOREIGN KEY(truckID)
123     REFERENCES truck(id) ON DELETE NO ACTION ON UPDATE CASCADE
124 );
125
126 ----- #9 create manages table [RELATIONSHIP] -----
127 CREATE TABLE manages(
128     secretaryID VARCHAR(50) NOT NULL,
129     driverID VARCHAR(50) NOT NULL,
130
131     CONSTRAINT pk_manages PRIMARY KEY (secretaryID, driverID),
132     CONSTRAINT fk_manages_secretaryId FOREIGN KEY(secretaryID)
133     REFERENCES secretary(id) ON DELETE NO ACTION ON UPDATE
134     CASCADE,
135     CONSTRAINT fk_manages_driverId FOREIGN KEY(driverID)
136     REFERENCES driver(id) ON DELETE NO ACTION ON UPDATE CASCADE
137 );

```

```

138
139 ----- #10 create records table [RELATIONSHIP] -----
140 CREATE TABLE records(
141     secretaryID VARCHAR(50) NOT NULL,
142     ordersID VARCHAR(50) NOT NULL,
143
144     CONSTRAINT pk_records PRIMARY KEY (secretaryID, ordersID),
145     CONSTRAINT fk_records_secretary FOREIGN KEY(secretaryID)
146     REFERENCES secretary(id) ON DELETE NO ACTION ON UPDATE
147     CASCADE,
148     CONSTRAINT fk_records_ordersId FOREIGN KEY(ordersID)
149     REFERENCES orders(id) ON DELETE NO ACTION ON UPDATE CASCADE
150 );

```

## Εισαγωγή Δεδομένων

```

1 ----- INSERT MOCK DATA IN DATABASE -----
2 USE omada8;
3
4 ----- #1 insert into secretary values -----
5 INSERT INTO secretary (id, firstName, lastName, phone,
6 username, password)
7 VALUES
8 (1, 'Vasiliki', 'Miliou', 6950236484,
9 'Penelope12', 'hgjhdghj'),
10 (2, 'Eirini', 'Fyssa', 6978543242,
11 'NICK23', 'hewurhw' ),
12 (3, 'Panagiota', 'Papadopoulou', 6945426754,
13 'CHASEKLOM', 'JIIEJhiuh'),
14 (4, 'Nefeli', 'Papadaki', 6932401905,
15 '12JENNIFER', 'bhuhdw'),
16 (5, 'Thanasis', 'Athanasίου', 6924164371,
17 'J10Hthhr2NY', 'gwd2328'),
18 (6, 'Kostas', 'Dimitriou', 6947548910,
19 'BETTEsc', '121ewdewFHf2'),
20 (7, 'Giannis', 'Kontou', 6995847463,
21 'MOSasxTEL', 'BCHDxsbsb2335'),
22 (8, 'Andreas', 'Panagiotou', 6954874263,
23 'MATTHss12EW', '123sajdh26'),
24 (9, 'Eleni', 'Tsironi', 6926154487,
25 'JOEdwe', 'cdjc216emkm'),
26 (10, 'Philippos', 'Michailidis', 6978514263,
27 'CHRISgvrge1TIAN', 'GABLEjcjje');
28
29 ----- #2 insert into driver values -----
30 INSERT INTO driver(id, firstName, lastName, phone)
31 VALUES
32 (1, 'Pyrrhos', 'Aggelidis', '6971234567'),
33 (2, 'Epameinondas', 'Zafeiropoulos', '6949274527'),

```

```

34 (3, 'Kassandra', 'Kyriakidi', '6970013686'),
35 (4, 'Katerina', 'Dimitriou', '6980536831'),
36 (5, 'Giannis', 'Karabatos', '6949435810'),
37 (6, 'Petros', 'Kontos', '6975542021'),
38 (7, 'Petros', 'Kontos', '6975542021');
39
40 ----- #3 insert into truck values -----
41 INSERT INTO truck(id, manufacturer, capacity,
42 productionYear, purchaseYear)
43 VALUES
44 ('ZFK-5678', 'Mercedes', '10 tons', '1999', '2009'),
45 ('YKD-8899', 'Scania', '8 tons', '2008', '2010'),
46 ('VRI-8753', 'Scania', '10 tons', '1999', '2010'),
47 ('VRI-7753', 'Scania', '10 tons', '1999', '2010'),
48 ('KOR-9981', 'Volvo', '10 tons', '1998', '2013'),
49 ('KOR-9991', 'Volvo', '10 tons', '1998', '2013'),
50 ('PLO-2266', 'Scania', '12 tons', '2001', '2015'),
51 ('KEO-7185', 'Volvo', '8 tons', '2005', '2012');
52
53 ----- #4 insert into client values -----
54 INSERT INTO client(id, firstName, lastName)
55 VALUES
56 (1, 'Giorgos', 'Kontogiannis'),
57 (2, 'Eleni', 'Markopoulou'),
58 (3, 'Nikos', 'Dimitrakopoulos'),
59 (4, 'Anastasia', 'Georgiou'),
60 (5, 'Dimitris', 'Karabasis'),
61 (6, 'Aikaterini', 'Michailidou'),
62 (7, 'Pavlos', 'Katsoulis'),
63 (8, 'Sofia', 'Liberopoulou'),
64 (9, 'Kostas', 'Papanikolaou'),
65 (10, 'Maria', 'Panagiotou'),
66 (11, 'Maria', 'Panagiotou'),
67 (12, 'Maria', 'Panagiotou');
68
69 ----- #5 insert into client phones values -----
70 INSERT INTO clientPhone(clientId, phone)
71 VALUES
72 (1, 6810203041),
73 (1, 6452013687),
74 (2, 6958525654),
75 (3, 6958524565),
76 (4, 6945452564),
77 (4, 6845210395),
78 (5, 6898653212),
79 (6, 6484858689),
80 (7, 6959842364),
81 (8, 6465615253),
82 (8, 6987452034),
83 (8, 6954203956),
84 (9, 6845124793),

```

```

85 (10, 6456751520);
86
87 ----- #6 insert into orders values -----
88 INSERT INTO orders(id, price, status, secretaryID, clientID)
89 VALUES
90 (1, 200, 'Received', 1, 1),
91 (2, 450, 'Received', 1, 1),
92 (3, 200, 'Done', 2, 3),
93 (4, 300, 'Done', 3, 4),
94 (5, 350, 'Done', 4, 5),
95 (6, 400, 'Received', 5, 6),
96 (7, 300, 'Received', 6, 7),
97 (8, 250, 'Received', 7, 8),
98 (9, 200, 'Done', 8, 9),
99 (10, 500, 'Received', 9, 10),
100 (11, 200, 'Done', 9, 5),
101 (12, 350, 'Done', 10, 5);
102
103 ----- #7 insert into payment values -----
104 INSERT INTO payment(id, amount, method, date, clientID, orderID)
105 VALUES
106 (1, '250', 'Cash', '2023-05-21', 1, 1),
107 (2, '250', 'Cash', '2019-05-21', 1, 3),
108 (3, '250', 'Cash', '2019-05-21', 2, 2),
109 (4, '250', 'Cash', '2019-05-21', 3, 12),
110 (5, '350', 'Card', '2023-02-07', 3, 4),
111 (6, '350', 'Card', '2023-11-30', 4, 5),
112 (7, '200', 'Card', '2023-09-03', 5, 6),
113 (8, '300', 'Cash', '2023-12-19', 6, 7),
114 (10, '400', 'Cash', '2023-09-17', 8, 8),
115 (11, '400', 'Cash', '2023-09-17', 9, 10),
116 (12, '400', 'Cash', '2023-09-17', 8, 11);
117
118 ----- #8 insert into delivery values -----
119 INSERT INTO delivery(id, pCity, pStreet, pNumber, pPostalCode,
120 dCity, dStreet, dNumber, dPostalCode,
121 date, ordersID, driverID, truckID)
122 VALUES
123 (1, 'Komotini', 'Agios Dimitriou Street', 28, 69100,
124 'Alexandroupoli', 'Old Patras Street', 75, 69131,
125 '2023-06-21', 1, 1, 'ZFK-5678'),
126 (2, 'Thessaloniki', 'Egnatias Street', 56, 54625,
127 'Patra', 'Agios Nikolaos Street', 120, 26444,
128 '2023-04-12', 2, 1, 'YKD-8899'),
129 (3, 'Larisa', 'Ethnikis Antistasis Street', 9, 41222,
130 'Xanthi', 'Dimitrakopoulou Street', 8, 67100,
131 '2023-02-28', 3, 2, 'ZFK-5678'),
132 (4, 'Thessaloniki', 'Egnatias Street', 56, 54625,
133 'Patra', 'Agios Nikolaos Street', 120, 26444,
134 '2023-04-12', 4, 2, 'YKD-8899'),
135 (5, 'Tripoli', 'Leonidou Street', 45, 22131,

```

```

136 'Thessaloniki', 'Demokratias Street', 42, 54631,
137 '2023-01-26', 5, 7, 'PLO-2266'),
138 (6, 'Kalamata', 'Papaflessa Street', 20, 24100,
139 'Larisa', 'Achilleos Street', 52, 41335,
140 '2023-06-16', 6, 7, 'KEO-7185'),
141 (7, 'Kastoria', 'Megalous Alexandrou Street', 5, 52100,
142 'Kozani', 'Papanastasiou Street', 10, 50123,
143 '2023-02-22', 7, 1, 'ZFK-5678'),
144 (8, 'Serres', 'Aristotelous Street', 6, 62122,
145 'Arta', 'Ioanni Karydi Street', 28, 47100,
146 '2023-03-14', 8, 2, 'ZFK-5678'),
147 (9, 'Corinth', 'Ismenis Street', 18, 20100,
148 'Lamia', 'Mesologgiou Street', 23, 35122,
149 '2023-04-19', 9, 3, 'ZFK-5678'),
150 (10, 'Argos', 'Kapodistriou Street', 12, 21200,
151 'Nafplio', 'Ioanninon Street', 40, 21100,
152 '2023-02-10', 10, 3, 'ZFK-5678'),
153 (11, 'Corinth', 'Ismenis Street', 18, 20100,
154 'Lamia', 'Mesologgiou Street', 23, 35122,
155 '2023-04-19', 11, 4, 'ZFK-5678'),
156 (12, 'Corinth', 'Ismenis Street', 18, 20100,
157 'Lamia', 'Mesologgiou Street', 23, 35122,
158 '2023-04-19', 12, 2, 'ZFK-5678');
159
160 ----- #9 insert into manages values -----
161 INSERT INTO manages(secretaryId, driverId)
162 VALUES
163 (2, 4),
164 (2, 5),
165 (2, 2),
166 (3, 3),
167 (4, 4),
168 (1, 1),
169 (6, 3),
170 (5, 5),
171 (3, 1);
172
173 ----- #10 insert into records values -----
174 INSERT INTO records(secretaryId, ordersId)
175 VALUES
176 (1, 1),
177 (1, 7),
178 (1, 4),
179 (2, 2),
180 (2, 1),
181 (2, 7),
182 (2, 9),
183 (3, 3),
184 (3, 7),
185 (4, 4),
186 (4, 2),

```

```

187 (5, 4),
188 (5, 6),
189 (5, 5),
190 (6, 5),
191 (6, 6),
192 (7, 7),
193 (8, 8),
194 (9, 9),
195 (9, 10);
196
197 COMMIT;
198
199 -----
200 -----
201
202 USE omada8;
203
204 ----- verify tables are accurate -----
205 SELECT count(*) FROM secretary;
206 SELECT * FROM secretary;
207
208 SELECT count(*) FROM driver;
209 SELECT * FROM driver;
210
211 SELECT count(*) FROM truck;
212 SELECT * FROM truck;
213
214 SELECT count(*) FROM client;
215 SELECT * FROM client;
216
217 SELECT count(*) FROM clientPhone;
218 SELECT * FROM clientPhone;
219
220 SELECT count(*) FROM orders;
221 SELECT * FROM orders;
222
223 SELECT count(*) FROM payment;
224 SELECT * FROM payment;
225
226 SELECT count(*) FROM delivery;
227 SELECT * FROM delivery;
228
229 SELECT count(*) FROM manages;
230 SELECT * FROM manages;
231
232 SELECT count(*) FROM records;
233 SELECT * FROM records;

```

# Βιβλιογραφία

- [1] R. Elmasri and S. B. Navathe. *Fundamentals of database systems*. Addison-Wesley, 2011.