

# ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ 2

## ΕΡΓΑΣΤΗΡΙΑΚΟ ΜΕΡΟΣ

ΟΜΑΔΕΣ Εργασίας τμήμα Τρίτης – Άσκηση 2

Χειμερινό Εξάμηνο 2020-2021, Π.Σ. ΠΑΔΑ

ΠΡΩΙΟΣ ΠΑΝΤΕΛΕΗΜΩΝ Α.Μ. 19390023 - ice18390023@uniwa.gr

ΛΥΚΟΥΔΗ ΔΕΣΠΟΙΝΑ Α.Μ. 18390151 - ice18390151@uniwa.gr

Υπεύθυνη Καθηγήτρια: Ζαχαρένια Γαροφαλάκη



## Περιεχόμενα

Αλλαγές στην προυπάρχουσα βάση .....	1
Προσθήκη στήλης τύπου DATE .....	1
Προσθήκη νέων δεδομένων στου πίνακες FLIGHTS και CUSTOMERS .....	2
Αλλαγή στις όψεις.....	2
Δημιουργία trigger (ερώτημα 1).....	3
Δημιουργία συνάρτησης (ερώτημα 2).....	5
Δημιουργία procedure (ερώτημα 3) .....	8

## Αλλαγές στην προυπάρχουσα βάση

Η προυπάρχουσα βάση, επειδή δεν ικανοποιούσε πλήρως της ανάγκες, διαμορφοποιήθηκε κατάλληλα.

### Προσθήκη στήλης τύπου DATE

Στην βάση CHARTERED\_AIRLINES, προσθέσαμε μια στήλη με τύπο DATE, στους πίνακες FLIGHTS και RESERVATIONS. Επίσης, προσθέσαμε και ημερωμινίες στις εγγραφές που υπάρχουν με τα ακόλουθα queries:

```
ALTER TABLE flights
ADD COLUMN DEPARTURE_DATE DATE;
```

```
ALTER TABLE reservations
ADD COLUMN RESERVATION_DATE DATE;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2021-2-1'
WHERE
    FLIGHT_NO = 1;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2021-2-10'
WHERE
    FLIGHT_NO = 7;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2020-12-26'
WHERE
    FLIGHT_NO = 2;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2021-7-5'
WHERE
    FLIGHT_NO = 3;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2021-4-12'
WHERE
    FLIGHT_NO = 4;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2021-4-18'
WHERE
    FLIGHT_NO = 5;
```

```
UPDATE FLIGHTS
SET
    DEPARTURE_DATE = '2020-12-20'
WHERE
    FLIGHT_NO = 6;
```

```

UPDATE RESERVATIONS
SET
    RESERVATION_DATE = '2020-12-20'
WHERE
    RESERVATION_NO = 5 OR RESERVATION_NO = 1;

```

```

UPDATE RESERVATIONS
SET
    RESERVATION_DATE = '2020-12-19'
WHERE
    RESERVATION_NO = 6 OR RESERVATION_NO = 2;

```

```

UPDATE RESERVATIONS
SET
    RESERVATION_DATE = '2020-11-5'
WHERE
    RESERVATION_NO = 7;

```

```

UPDATE RESERVATIONS
SET
    RESERVATION_DATE = '2020-8-29'
WHERE
    RESERVATION_NO = 4;

```

```

UPDATE RESERVATIONS
SET
    RESERVATION_DATE = '2020-8-10'
WHERE
    RESERVATION_NO = 3;

```

## Προσθήκη νέων δεδομένων στον πίνακες FLIGHTS και CUSTOMERS

Προσθέσαμε, δύο νέα δεδομένα στην βάση, με τα ονόματα 'DEFAULT' και primary key το 0, έτσι ώστε να τα χρησιμοποιούμε κατάλληλα αργότερα.

```

INSERT INTO FLIGHTS( FLIGHT_NO, DEPARTURE, ARRIVAL, FLIGHT_TYPE, SEATS,
FREE_SEATS, DEPARTURE_DATE)
VALUES (0, 'DEFAULT', 'DEFAULT', 'IN', 9999, 9999, '1999-07-09');

```

```

INSERT INTO CUSTOMERS( CUSTOMER_NO, FIRST_NAME, LAST_NAME, NATIONALITY,
BIRTH_DATE)
VALUES (0, 'DEFAULT', 'DEFAULT', 'DEFAULT', '1999-07-09');

```

## Αλλαγή στις όψεις

Στην όψη A\_VIEW, προστέθηκε μία νέα στήλη, η οποία χρησιμοποιεί την συνάρτηση που δημιουργήσαμε στο ερώτημα 2. Επιπλέον, προστέθηκε προστέθηκε μια WHERE για να μην επιστρέφει τα της πτήσης που έχουν FLIGHT\_NO ίσο με το 0.

```

ALTER VIEW A_VIEW( RESERVATION_NO, DEPARTURE, ARRIVAL, FIRST_NAME, LAST_NAME,
COST, RESERVED_BEFORE)
AS SELECT RESERVATION_NO, DEPARTURE, ARRIVAL, FIRST_NAME, LAST_NAME, COST,
DIFF_DATE_BASED_ON_RESERV_SECOND_OPTION(RESERVATION_NO)
FROM RESERVATIONS
JOIN CUSTOMERS ON CUSTOMERS.CUSTOMER_NO = RESERVATIONS.CUSTOMER_NO

```

```

JOIN FLIGHTS ON FLIGHTS.FLIGHT_NO = RESERVATIONS.FLIGHT_NO
WHERE RESERVATIONS.FLIGHT_NO != 0
ORDER BY cost DESC;

```

Στην όψη TOTAL\_FLIGHTS, προσθέσαμε επίσης μία WHERE, για να μην εμφανίζονται πτήσης με FLIGHT\_NO ίσο με το 0.

```

ALTER VIEW TOTAL_FLIGHTS( FLIGHT_NO, TOTAL_RESERVATIONS, REMAINING_SEATS)
AS SELECT RESERVATIONS.FLIGHT_NO, count(RESERVATIONS.FLIGHT_NO), FLIGHTS.SEATS-
count(RESERVATIONS.FLIGHT_NO)
FROM RESERVATIONS
JOIN FLIGHTS ON RESERVATIONS.FLIGHT_NO = FLIGHTS.FLIGHT_NO
WHERE
RESERVATIONS.FLIGHT_NO != 0
GROUP BY FLIGHT_NO;

```

## Δημιουργία trigger (ερώτημα 1)

Δημιουργίσαμε τρεις triggers. Οι δύο ήταν απαραίτητη, διότι επειδή έχουμε unsigned τιμές, εμφανιζόταν error στον πίνακα όταν εκχωρούνταν παραπάνω εγγραφές από ότι ελεύθερες θέσεις σε μια πτήση.

Ο trigger SET\_FREE\_SEATS\_EQUAL\_TO\_SEATS, δημιουργήθηκε απλώς για να εκχωρεί στην στήλη με τις ελεύθερες θέσεις, την τιμή των θέσεων, εφόσον δεν υπάρχει τιμή στις ελεύθερες θέσεις. Επίσης, αν οι συνολικές θέσεις είναι NULL, τότε εκχωρεί 0. Μελλοντικά, πρέπει να δημιουργήσουμε δυο trigger, οι οποίοι σε κάποιο update, θα ελέγχει αν έγινε αλλαγή στις συνολικές θέσεις. Ο before update, θα βάζει την τιμή των συνολικών θέσεων, σε μια νέα στήλη που θα πρέπει να δημιουργήσουμε και ο after update, θα ελέγχει αν είναι διαφορετικοί οι αριθμοί και τότε θα εκχωρεί την τιμή των συνολικών θέσεων στις ελεύθερες θέσεις μείων τα reservations που έχουν γίνει για το συγκεκριμένος flight\_no απο των πίνακα TOTAL\_FLIGHTS με το query:

```

SELECT TOTAL_RESERVATIONS FROM TOTAL_FLIGHTS WHERE FLIGHT_NO = NEW.FLIGHT_NO;

```

και μετά θα αλλάζει την τιμή της στήλης που άλλαξε ο trigger before update.

Ο trigger SET\_FREE\_SEATS\_EQUAL\_TO\_SEATS, υλοποιείται ως εξής:

```

DELIMITER //
CREATE TRIGGER SET_FREE_SEATS_EQUAL_TO_SEATS
BEFORE INSERT ON FLIGHTS
FOR EACH ROW
BEGIN
    SET NEW.FREE_SEATS = IFNULL(NEW.FREE_SEATS, IFNULL(NEW.SEATS, 0));
END//
DELIMITER ;

```

Και δύο παραδείγματα για τον έλεγχο της σωστή λειτουργίας του:

```

INSERT INTO FLIGHTS (FLIGHT_NO, DEPARTURE, ARRIVAL, FLIGHT_TYPE, SEATS,
DEPARTURE_DATE)
VALUES (8, 'Nikos Kazantzakis', 'Eleutherios Venizelos', 'IN', 300, '2021-07-
20');

```

```

INSERT INTO FLIGHTS (FLIGHT_NO, DEPARTURE, ARRIVAL, FLIGHT_TYPE, SEATS,
FREE_SEATS, DEPARTURE_DATE)
VALUES (9, 'Nikos Kazantzakis', 'Makedonia', 'IN', 300, 250, '2021-01-5');
SELECT * FROM FLIGHTS;

```

Ο trigger CHECK\_AVAILABILITY, ελέγχει την διαθεσιμότητα και βάση αυτής, αλλάζει τις τιμές στις στήλες FLIGHT\_NO, CUSTOMER\_NO, COST σε 0 και την στήλη RESERVATION\_DATE σε 1999-07-09.

```
DELIMITER //
CREATE TRIGGER CHECK_AVAILABILITY
BEFORE INSERT ON RESERVATIONS
FOR EACH ROW
BEGIN
IF ( (SELECT REMAINING_SEATS
        FROM TOTAL_FLIGHTS
        WHERE TOTAL_FLIGHTS.FLIGHT_NO = NEW.FLIGHT_NO) <= 0 ) THEN # Αν ήταν
το free_seats των flights τότε θα έπρεπε να ήταν > 0
SET
        NEW.FLIGHT_NO = 0, NEW.CUSTOMER_NO = 0, NEW.COST = 0,
NEW.RESERVATION_DATE = '1999-7-9';
END IF;
END
//
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS CHECK_AVAILABILITY;
```

Ο trigger SEATS\_AVAILABILITY, μειώνει κατά μια ελεύθερη θέση από τον πίνακα FLIGHTS, αν το FLIGHT\_NO δεν είναι 0.

```
DELIMITER //
CREATE TRIGGER SEATS_AVAILABILITY
AFTER INSERT ON RESERVATIONS
FOR EACH ROW
BEGIN
IF ( (IFNULL((SELECT REMAINING_SEATS
        FROM TOTAL_FLIGHTS
        WHERE TOTAL_FLIGHTS.FLIGHT_NO = NEW.FLIGHT_NO), -1) >= 0) # Αν ήταν
το free_seats των flights τότε θα έπρεπε να ήταν > 0
AND
        NEW.CUSTOMER_NO > 0) THEN
UPDATE FLIGHTS
SET
        FREE_SEATS = FREE_SEATS-1
WHERE
        FLIGHTS.FLIGHT_NO = NEW.FLIGHT_NO;
END IF;
END//
DELIMITER ;
```

Εισάγωντας, με το εξής ερώτημα νέες εγγραφές, θα παρατηρήσουμε πως και οι δυο triggers δουλεύουν ορθά.

```
INSERT INTO RESERVATIONS ( RESERVATION_NO, CUSTOMER_NO, FLIGHT_NO, COST,
RESERVATION_DATE)
VALUES (8, 1, 4, 50, '2020-12-20'),
(9, 2, 4, 50, '2020-12-20'),
(10, 3, 4, 50, '2020-12-20'),
(11, 4, 4, 50, '2020-12-20');

SELECT * FROM FLIGHTS;
SELECT * FROM RESERVATIONS;
```

Στον πίνακα FLIGHTS, οι ελεύθερες θέσεις έχουν μειωθεί στο 0 και στον πίνακα RESERVATIONS, η τελευταία εκχώρησε με RESERVATION\_NO ίσο με 11, πλέον έχει αλλάξει. Δώστε έμφαση στο FLIGHT\_NO για 4 και στα RESERVATION\_NO 8,9,10 και 11.

Πίνακας 1 FLIGHTS

FLIGHT_NO	DEPARTURE	ARRIVAL	FLIGHT_TYPE	SEATS	FREE_SEATS	DEPARTURE_DATE
0	DEFAULT	DEFAULT	IN	9999	9999	1999-07-09
1	Eleutherios Venizelos	South Andros	OUT	300	299	2021-02-01
2	Eleutherios Venizelos	Manchester	OUT	300	300	2020-12-26
3	Eleutherios Venizelos	Nikos Kazantzakis	IN	300	300	2021-07-05
4	Eleutherios Venizelos	Makedonia	IN	3	0	2021-04-12
5	Makedonia	Eleutherios Venizelos	IN	3	3	2021-04-18
6	Makedonia	Nikos Kazantzakis	IN	3	3	2020-12-20
7	South Andros	Eleutherios Venizelos	OUT	300	300	2021-02-10
8	Nikos Kazantzakis	Eleutherios Venizelos	IN	300	300	2021-07-20
9	Nikos Kazantzakis	Makedonia	IN	300	250	2021-01-05
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Πίνακας 2 RESERVATIONS

	RESERVATION_NO	CUSTOMER_NO	FLIGHT_NO	COST	RESERVATION_DATE
▶	1	1	1	1000.00	2020-12-20
	2	2	1	600.00	2020-12-19
	3	3	3	25.00	2020-08-10
	4	4	3	27.30	2020-08-29
	5	1	7	1300.00	2020-12-20
	6	2	7	800.00	2020-12-19
	7	1	3	48.73	2020-11-05
	8	1	4	50.00	2020-12-20
	9	2	4	50.00	2020-12-20
	10	3	4	50.00	2020-12-20
	11	0	0	0.00	1999-07-09
*	NULL	NULL	NULL	NULL	NULL

## Δημιουργία συνάρτησης (ερώτημα 2)

Δημιουργήσαμε 2 συναρτήσεις, εκ των οποίων, η μία επιστρέφει μόνο ημέρες και η άλλη επιστρέφει χρόνια/μήνες/ημέρες.

Η συνάρτηση DIFF\_DATE\_BASED\_ON\_RESERV, επιστρέφει έναν αριθμό ακολοθούμενο απο το μήνυμα 'ημέρες' και δέχεται ως παράμετρο τον αριθμό της κράτησης.

```
DELIMITER //
CREATE FUNCTION DIFF_DATE_BASED_ON_RESERV( RESERV_NO INT(10) )
RETURNS VARCHAR(12)
BEGIN
    DECLARE DATE1 DATE;
    DECLARE DATE2 DATE;
```

```

SET DATE1 = (SELECT DEPARTURE_DATE
FROM FLIGHTS
JOIN RESERVATIONS ON FLIGHTS.FLIGHT_NO = RESERVATIONS.FLIGHT_NO
WHERE RESERVATION_NO = RESERV_NO);

SET DATE2 = (SELECT RESERVATION_DATE
FROM FLIGHTS
JOIN RESERVATIONS ON FLIGHTS.FLIGHT_NO = RESERVATIONS.FLIGHT_NO
WHERE RESERVATION_NO = RESERV_NO);

RETURN CONCAT( DATEDIFF( DATE1, DATE2), " ημέρες");
END//
DELIMITER ;

```

Η συνάρτηση DIFF\_DATE\_BASED\_ON\_RESERV\_SECOND\_OPTION επιστρέφει τα χρόνια αν υπάρχουν, μήνες αν υπάρχουν και ημέρες.

```

DELIMITER //
CREATE FUNCTION DIFF_DATE_BASED_ON_RESERV_SECOND_OPTION( RESERV_NO INT(10) )
RETURNS VARCHAR(255)
BEGIN
    DECLARE DATE1 DATE;
    DECLARE DATE2 DATE;
    DECLARE NUM_OF_DATES INT(6);
    DECLARE TMP INT(6);
    DECLARE RETURN_VALUE VARCHAR(255) DEFAULT "";

    SET DATE1 = (SELECT DEPARTURE_DATE
FROM FLIGHTS
JOIN RESERVATIONS ON FLIGHTS.FLIGHT_NO = RESERVATIONS.FLIGHT_NO
WHERE RESERVATION_NO = RESERV_NO);

    SET DATE2 = (SELECT RESERVATION_DATE
FROM FLIGHTS
JOIN RESERVATIONS ON FLIGHTS.FLIGHT_NO = RESERVATIONS.FLIGHT_NO
WHERE RESERVATION_NO = RESERV_NO);

    SET NUM_OF_DATES = DATEDIFF( DATE1, DATE2);

    IF (NUM_OF_DATES >= 365) THEN

        IF (SELECT LENGTH(RETURN_VALUE) > 0) THEN
            SET RETURN_VALUE = CONCAT(RETURN_VALUE, ", ");
        END IF;

        SET TMP = (SELECT NUM_OF_DATES DIV 365 );

        SET NUM_OF_DATES = NUM_OF_DATES - 365 * TMP;

        IF ( TMP > 1) THEN
            SET RETURN_VALUE = CONCAT(RETURN_VALUE, TMP, " χρόνια");
        ELSE
            SET RETURN_VALUE = CONCAT(RETURN_VALUE, TMP, " χρόνο");
        END IF;
    END IF;

    IF (NUM_OF_DATES >= 30) THEN

        IF (SELECT LENGTH(RETURN_VALUE) > 0) THEN

```



```

        SET RETURN_VALUE = CONCAT(RETURN_VALUE, ", ");
    END IF;

    SET TMP = (SELECT NUM_OF_DATES DIV 30 );

    SET NUM_OF_DATES = NUM_OF_DATES - 30 * TMP;

    IF ( TMP > 1) THEN
        SET RETURN_VALUE = CONCAT(RETURN_VALUE, TMP, " μήνες");
    ELSE
        SET RETURN_VALUE = CONCAT(RETURN_VALUE, TMP, " μήνα");
    END IF;
END IF;

IF (NUM_OF_DATES >= 1 OR LENGTH(RETURN_VALUE) = 0) THEN

    IF (SELECT LENGTH(RETURN_VALUE) > 0) THEN
        SET RETURN_VALUE = CONCAT(RETURN_VALUE, ", ");
    END IF;

    IF ( NUM_OF_DATES = 1) THEN
        SET RETURN_VALUE = CONCAT(RETURN_VALUE, NUM_OF_DATES, " ημέρα");
    ELSE
        SET RETURN_VALUE = CONCAT(RETURN_VALUE, NUM_OF_DATES, " ημέρες");
    END IF;
END IF;

RETURN RETURN_VALUE;
END//
DELIMITER ;

```

και με το ακόλουθο query, ελέγχουμε την ορθότητα της δεύτερης συνάρτησης

```

SELECT RESERVATION_NO, DIFF_DATE_BASED_ON_RESERV_SECOND_OPTION( RESERVATION_NO)
AS 'Διαφορά ημερών'
FROM RESERVATIONS
WHERE FLIGHT_NO != 0;

```

Πίνακας 3 Αποτελέσματα του query με την χρήση της συνάρτησης.

RESERVATION_NO	Διαφορά ημερών
1	1 μήνα, 13 ημέρες
2	1 μήνα, 14 ημέρες
3	10 μήνες, 29 ημέρες
4	10 μήνες, 10 ημέρες
7	8 μήνες, 2 ημέρες
8	3 μήνες, 23 ημέρες
9	3 μήνες, 23 ημέρες
10	3 μήνες, 23 ημέρες
5	1 μήνα, 22 ημέρες
6	1 μήνα, 23 ημέρες

### Δημιουργία procedure (ερώτημα 3)

Η procedure, δέχεται 2 παραμέτρους, όπου την πρώτη την χρησιμοποιεί για να αναζητήσει βάση της τιμής αυτής και στην δεύτερη, αλλάζει την τιμή με το αποτέλεσμα. Η υλοποίηση της procedure είναι η εξής:

```
DELIMITER //
```

```
CREATE PROCEDURE RETURN_PROFIT_FROM_FLIGHT( IN FLIGHT_ID INT(10), OUT PROFIT
```

```
INT(10))
```

```
BEGIN
```

```
    SET PROFIT = (SELECT SUM(COST) FROM A_VIEW WHERE FLIGHT_ID = FLIGHT_NO);
```

```
END//
```

```
DELIMITER ;
```

Ελέγχουμε την ορθή λειτουργία της procedure

```
SET @FLIGHT_ID = 1;
```

```
SET @PROFIT = 0;
```

```
CALL RETURN_PROFIT_FROM_FLIGHT( @FLIGHT_ID, @PROFIT);
```

```
SELECT @FLIGHT_ID, @PROFIT;
```

```
SELECT SUM(COST) FROM RESERVATIONS WHERE FLIGHT_NO = @FLIGHT_ID;
```

Πίνακας 4 Αποτελέσματα του query *SELECT @FLIGHT\_ID, @PROFIT*

@FLIGHT_ID	@PROFIT
1	1600