



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

ΛΟΓΟΘΕΤΗΣ ΑΛΕΞΙΟΣ-ΑΝΤΩΝΙΟΣ – ICE 20390132  
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ II – ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2Η  
ΕΡΓΑΣΤΗΡΙΑΚΟ ΤΜΗΜΑ 11

## Ο ΚΩΔΙΚΑΣ ΟΛΟΚΛΗΡΩΜΕΝΟΣ

```
DROP DATABASE IF EXISTS CharteredAirlines;  
CREATE DATABASE CharteredAirlines;  
USE CharteredAirlines;
```

```
CREATE TABLE FLIGHTS(FNO INT(4)AUTO_INCREMENT,DEPARTURE  
VARCHAR(50),ARRIVAL VARCHAR(50),INTOREXT VARCHAR(3) ,SEATS INT NOT  
NULL,FREE INT DEFAULT(SEATS),DATE_OF_DEP DATE,PRIMARY KEY  
(FNO),CONSTRAINT INTEXT CHECK(INTOREXT='INT' OR  
INTOREXT='EXT'),CONSTRAINT DEPARR CHECK(DEPARTURE!=ARRIVAL));
```

```
CREATE TABLE CUSTOMERS(CID INT(4) AUTO_INCREMENT,NAME VARCHAR(50),SURNAME  
VARCHAR(50),NATIONALITY VARCHAR(50),DOB DATE,PRIMARY KEY(CID));
```

```
CREATE TABLE RESERVATIONS(RID INT(4) AUTO_INCREMENT,COST FLOAT,CID  
INT(4),FNO INT(4),DATE_OF_RES DATE DEFAULT (CURRENT_DATE),PRIMARY  
KEY(RID),FOREIGN KEY(CID) REFERENCES CUSTOMERS(CID),FOREIGN KEY(FNO)  
REFERENCES FLIGHTS(FNO));
```

```
ALTER TABLE CUSTOMERS AUTO_INCREMENT=100;  
ALTER TABLE FLIGHTS AUTO_INCREMENT=100;  
ALTER TABLE RESERVATIONS AUTO_INCREMENT=100;
```

```
DELIMITER //
```

```
CREATE TRIGGER NEW_RESERVATION  
BEFORE INSERT ON RESERVATIONS  
FOR EACH ROW  
BEGIN  
UPDATE FLIGHTS  
SET FREE = FREE - 1  
WHERE FNO=NEW.FNO;  
END//
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE TRIGGER DELETE_RESERVATION  
AFTER DELETE ON RESERVATIONS  
FOR EACH ROW  
BEGIN  
UPDATE FLIGHTS  
SET FREE = FREE + 1  
WHERE FNO=OLD.FNO;  
END//
```

```
DELIMITER ;
```

DELIMITER //

```
CREATE TRIGGER UPDATE_RESERVATION
AFTER UPDATE ON RESERVATIONS
FOR EACH ROW
BEGIN
UPDATE FLIGHTS
SET FREE = SEATS - (SELECT COUNT(*)
FROM RESERVATIONS
WHERE FID=NEW.FNO);
END//
```

DELIMITER ;

ALTER TABLE RESERVATIONS ADD CONSTRAINT UNIQUE KEY(CID,FNO);

```
INSERT INTO CUSTOMERS(NAME,SURNAME,NATIONALITY,DOB)
VALUES('ALEXIS','LOGOTHETIS','GR','2002-03-24');
INSERT INTO CUSTOMERS(NAME,SURNAME,NATIONALITY,DOB)
VALUES('DIMITRIS','DIMITRAKIS','IND','2002-05-02');
INSERT INTO CUSTOMERS(NAME,SURNAME,NATIONALITY,DOB)
VALUES('PANAGIOTHIS','PANAGIOTAKHS','CYP','1999-09-12');
INSERT INTO CUSTOMERS(NAME,SURNAME,NATIONALITY,DOB)
VALUES('DIMITRIS','LOGOTHETIS','GR','2000-10-21');
INSERT INTO CUSTOMERS(NAME,SURNAME,NATIONALITY,DOB)
VALUES('KWSTAS','KWSTAKHS','GR','2000-12-22');
```

```
INSERT INTO FLIGHTS(DEPARTURE,ARRIVAL,INTOREXT,SEATS,DATE_OF_DEP)
VALUES('ATH','ZTH','INT',250,"2023-12-30");
INSERT INTO FLIGHTS(DEPARTURE,ARRIVAL,INTOREXT,SEATS,DATE_OF_DEP)
VALUES('ENG','US','EXT',500,"2023-12-12");
INSERT INTO FLIGHTS(DEPARTURE,ARRIVAL,INTOREXT,SEATS,DATE_OF_DEP)
VALUES('ATH','ENG','EXT',100,"2023-4-25");
INSERT INTO FLIGHTS(DEPARTURE,ARRIVAL,INTOREXT,SEATS,DATE_OF_DEP)
VALUES('SKG','ATH','INT',350,"2022-12-30");
```

```
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(75,100,100);
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(150,100,101);
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(30,100,102);
```

```
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(75,101,100);
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(75,102,100);
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(30,103,102);
```

```
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(75,103,100);
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(150,104,101);
INSERT INTO RESERVATIONS(COST,CID,FNO) VALUES(15,104,103);
```

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

```
SELECT FNO,COUNT(*)
FROM RESERVATIONS R
GROUP BY FNO;
```

```
DELETE FROM RESERVATIONS WHERE CID =(SELECT CID
FROM CUSTOMERS
WHERE NAME = 'ALEXIS');
```

```
SELECT FNO,COUNT(*)
FROM RESERVATIONS R
GROUP BY FNO;
```

```
SELECT C.NAME,C.SURNAME ,SUM(R.COST) AS 'COST'
FROM CUSTOMERS C,RESERVATIONS R
WHERE C.CID=R.CID
GROUP BY R.CID
ORDER BY SUM(R.COST) DESC;
```

```
DROP VIEW IF EXISTS NONUP_GROUP_CUST;
CREATE VIEW NONUP_GROUP_CUST(G_CID,COUNT_RSV,TOT_COST) AS
SELECT CID,COUNT(*),SUM(COST) FROM RESERVATIONS GROUP BY CID ORDER BY
SUM(COST) DESC;
```

```
DROP VIEW IF EXISTS FLIGHTS_IN_ATH;
CREATE VIEW
FLIGHTS_IN_ATH(V_FNO ,V_DEPARTURE ,V_ARRIVAL ,V_INTOREXT ,V_SEATS ,V_FREE
)
AS SELECT FNO,DEPARTURE,ARRIVAL,INTOREXT,SEATS,FREE FROM FLIGHTS WHERE
ARRIVAL='ATH' OR DEPARTURE='ATH' WITH CHECK OPTION;
```

```
DELIMITER //
```

```
CREATE FUNCTION CALC_DATE_DIFF( FLIGHT_DEP_DATE DATE,RES_DATE DATE)
RETURNS INT
DETERMINISTIC
BEGIN
```

```
RETURN DATEDIFF(FLIGHT_DEP_DATE,RES_DATE);  
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE TOTAL_PROFIT(IN FID INT)  
DETERMINISTIC  
BEGIN  
SELECT SUM(COST) AS "TOTAL PROFIT"  
FROM RESERVATIONS R  
WHERE FID=R.FNO;  
END //
```

```
DELIMITER ;
```

```
SELECT CALC_DATE_DIFF(F.DATE_OF_DEP,R.DATE_OF_RES) AS "DAYS LEFT" , R.RID  
FROM FLIGHTS F,RESERVATIONS R  
WHERE F.FNO=R.FNO;
```

```
CALL TOTAL_PROFIT(100);
```

```
SELECT * FROM FLIGHTS_IN_ATH;  
SELECT * FROM NONUP_GROUP_CUST;  
SELECT * FROM RESERVATIONS;  
SELECT * FROM FLIGHTS;
```

## ΕΡΩΤΗΜΑ ΠΡΩΤΟ

Δημιουργήστε trigger που θα ενημερώνει αυτόματα το πλήθος των ελεύθερων θέσεων, σε κάθε νέα κράτηση που θα εισάγεται στην ΒΔ. Δείξτε την λειτουργία του trigger με μία τέτοια εισαγωγή και τα αποτελέσματά της στο πλήθος ελεύθερων θέσεων.

```
CREATE TRIGGER NEW_RESERVATION
BEFORE INSERT ON RESERVATIONS
FOR EACH ROW
BEGIN
UPDATE FLIGHTS
SET FREE = FREE - 1
WHERE FNO=NEW.FNO;
END//

CREATE TRIGGER DELETE_RESERVATION
AFTER DELETE ON RESERVATIONS
FOR EACH ROW
BEGIN
UPDATE FLIGHTS
SET FREE = FREE + 1
WHERE FNO=OLD.FNO;
END//
```

```
DELETE FROM RESERVATIONS WHERE CID =(SELECT CID
FROM CUSTOMERS
WHERE NAME = 'ALEXIS');
```

	CID	NAME	SURNAME	NATIONALITY	DOB
▶	100	ALEXIS	LOGOTHETIS	GR	2002-03-24

## BEFORE DELETE

	RID	COST	CID	FNO	DATE_OF_RES
▶	100	75	100	100	2022-12-29
	101	150	100	101	2022-12-29
	102	30	100	102	2022-12-29
	103	75	101	100	2022-12-29
	104	75	102	100	2022-12-29
	105	30	103	102	2022-12-29
	106	75	103	100	2022-12-29
	107	150	104	101	2022-12-29
	108	15	104	103	2022-12-29
•	NULL	NULL	NULL	NULL	NULL

## AFTER DELETE

	RID	COST	CID	FNO	DATE_OF_RES
▶	103	75	101	100	2022-12-29
	104	75	102	100	2022-12-29
	105	30	103	102	2022-12-29
	106	75	103	100	2022-12-29
	107	150	104	101	2022-12-29
	108	15	104	103	2022-12-29
✱	NULL	NULL	NULL	NULL	NULL

[illegible][illegible]

## ΕΡΩΤΗΜΑ ΔΕΥΤΕΡΟ

Δημιουργήστε μια function η οποία θα επιστρέφει τη διάρκεια ισχύος μίας κράτησης, λαμβάνοντας υπόψιν την ημερομηνία της κράτησης και την ημερομηνία αναχώρησης της πτήσης (π.χ. 6 μήνες).

Θεωρώντας ότι ως default ημερομηνία κράτησης την ημερομηνία την οποία έγινε η κράτηση

```
,DATE_OF_RES DATE DEFAULT (CURRENT_DATE).
```

```
CREATE FUNCTION CALC_DATE_DIFF( FLIGHT_DEP_DATE DATE,RES_DATE DATE)
RETURNS INT
DETERMINISTIC
) BEGIN
RETURN DATEDIFF(FLIGHT_DEP_DATE,RES_DATE);
- END //

DELIMITER ;
```

```
SELECT CALC_DATE_DIFF(F.DATE_OF_DEP,R.DATE_OF_RES) AS "DAYS LEFT" , R.RID
FROM FLIGHTS F,RESERVATIONS R
WHERE F.FNO=R.FNO;
```

	DAYS LEFT	RID
▶	366	103
	366	104
	117	105
	366	106
	348	107
	1	108

## FLIGHTS

	FNO	DEPARTURE	ARRIVAL	INTOREXT	SEATS	FREE	DATE_OF_DEP
▶	100	ATH	ZTH	INT	250	247	2023-12-30
	101	ENG	US	EXT	500	499	2023-12-12
	102	ATH	ENG	EXT	100	99	2023-04-25
	103	SKG	ATH	INT	350	349	2022-12-30
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## RESERVATIONS

	RID	COST	CID	FNO	DATE_OF_RES
▶	103	75	101	100	2022-12-29
	104	75	102	100	2022-12-29
	105	30	103	102	2022-12-29
	106	75	103	100	2022-12-29
	107	150	104	101	2022-12-29
	108	15	104	103	2022-12-29
*	NULL	NULL	NULL	NULL	NULL

### ΕΡΩΤΗΜΑ ΤΡΙΤΟ

Δημιουργήστε μία procedure που θα δέχεται σαν είσοδο τον κωδικό μίας πτήσης και θα επιστρέφει τα συνολικά έσοδα από κρατήσεις αυτής της πτήσης για την εταιρία.

```
DELIMITER //
```

```
CREATE PROCEDURE TOTAL_PROFIT(IN FID INT)
```

```
DETERMINISTIC
```

```
3 BEGIN
```

```
SELECT SUM(COST) AS "TOTAL PROFIT"
```

```
FROM RESERVATIONS R
```

```
WHERE FID=R.FNO;
```

```
~ END //
```

```
DELIMITER ;
```

```
CALL TOTAL_PROFIT(100);
```

	RID	COST	CID	FNO	DATE_OF_RES
►	103	75	101	100	2022-12-31
	104	75	102	100	2022-12-31
	106	75	103	100	2022-12-31
*	NULL	NULL	NULL	NULL	NULL

	TOTAL PROFIT
►	225