

HY-360: Αρχεία και Βάσεις Δεδομένων

Χειμερινό Εξάμηνο 2024

Αναφορά εργασία μαθήματος



Κωνσταντίνιδης Γεώργιος csd3974

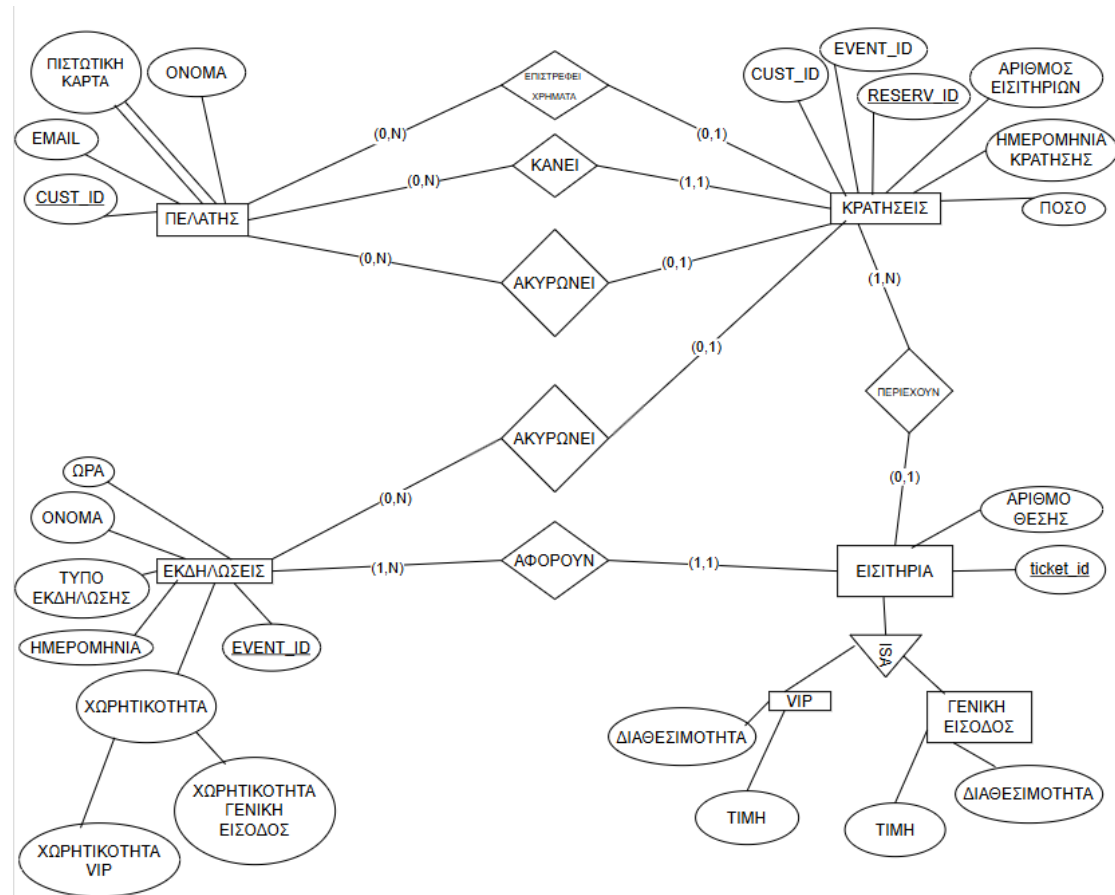
Καραγιάννης Ματθαίος csd5180

Παπαδάκης Γεώργιος csd4975

Δομή Α' Παραδοτέου

Φάση I: Εννοιολογική Μοντελοποίηση

Πλήρες διάγραμμα οντοτήτων-σχέσεων για την εταιρία



Τα γνωρίσματα (όνομα, τύπος) όλων των οντοτήτων και σχέσεων

- Όνομα_πελάτη->string
- Πιστωτική_καρτα->int
- Email ->string
- Cust_id->int
- Ποσό->int
- Ημερομηνία_κράτησης->string
- Αριθμός_VIP->int
- Αριθμός_γενικής->int
- Reserve_id->int
- Event_id->int
- Cust_id->int
- Αριθμό_θέσης->string

- Ticket_id->int
- Διαθεσιμότητα_γενική->boolean
- Τιμή_γενική->int
- Διαθεσιμότητα_vip->boolean
- Τιμή_vip->int
- Ώρα->string
- Όνομα->string
- Τύπος_εκδήλωσης->string
- Ημερομηνία->string
- Χωρητικότητα_vip->int
- Χωρητικότητα_γενική->int
- Χωρητικότητα->int
- Event_id->int

Τα πρωτεύοντα κλειδιά

- CUST_ID
- RESERV_ID
- TICKET_ID
- EVENT_ID

Επεξηγήσεις για τα μη προφανή γνωρίσματα και τις μη-προφανείς σχέσεις

Κατά την εκχώρηση ενός καινούργιου event αρχικοποιούνται όλα τα εισιτήρια, βάσει του αριθμού της χωρητικότητας της εκδήλωσης. Τα εισιτήρια αυτά, έχουν null στο reservation_id και TRUE στη boolean μεταβλητή “διαθεσιμότητα”. Συνεπώς, οι διαθέσιμες θέσεις για κάθε event προκύπτουν από το άθροισμα όλων των θέσεων με boolean “διαθεσιμότητα” να είναι TRUE. Κάθε πελάτης επιλέγει πόσα εισιτήρια θέλει να κλείσει (από κάθε τύπο εισιτηρίου: γενικής εισόδου ή vip), το σύστημα ελέγχει αν υπάρχουν τόσα διαθέσιμα εισιτήρια όσα ζητάει ο πελάτης και αν η αναζήτηση είναι επιτυχής τα εισιτήρια αυτά αποκτούν το reservation_id της κράτησης που έκανε ο πελάτης και “διαθεσιμότητα” ίση με FALSE.

Περιορισμούς πληθικότητας

Φαίνονται όλες οι πληθικότητες στο διάγραμμα οντοτήτων σχέσεων.

Τη μετάφραση του μοντέλου σας στο σχεσιακό μοντέλο

Πελάτης

<u>customer_id</u>	email	ονοματεπώνυμο
--------------------	-------	---------------

Πιστωτικές Κάρτες

<u>customer_id</u>	πιστωτική κάρτα
--------------------	-----------------

Κρατήσεις

<u>reservation_id</u>	customer_id	event_id	αριθμός εισιτηρίων γενικής εισόδου	αριθμός vip εισιτηρίων	ημερομηνία κράτησης	ποσό πληρωμής
-----------------------	-------------	----------	---------------------------------------------	------------------------------	------------------------	------------------

Εισιτήρια Γενικής Εισόδου

<u>ticket_id</u>	event_id	reservation_id	αριθμός θέσης	διαθεσιμότητα
------------------	----------	----------------	---------------	---------------

Εισιτήρια VIP

<u>ticket_id</u>	event_id	reservation_id	αριθμός θέσης	διαθεσιμότητα
------------------	----------	----------------	---------------	---------------

Εκδηλώσεις

<u>event_id</u>	όνομα	ημερομηνία	ώρα	τύπος εκδήλωσης	συνολική χωρητικότητα	vip χωρητικότητα	τιμή vip εισιτηρίου	χωρητικότητα γενικής εισόδου	τιμή εισιτηρίου γενικής εισόδου
-----------------	-------	------------	-----	--------------------	--------------------------	---------------------	------------------------	------------------------------------	------------------------------------

Τις εντολές της γλώσσας ορισμού δεδομένων για τις σχέσεις που προκύπτουν

```
CREATE TABLE CUST(  
    CUST_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    NAME VARCHAR(100) NOT NULL,  
    EMAIL VARCHAR(100) UNIQUE NOT NULL  
);
```

```
CREATE TABLE CARDS(  
    CUST_ID INT,  
    CARD_NUMBER VARCHAR(20) NOT NULL,  
    CARD_EXP_DATE TEXT NOT NULL,  
    CVV VARCHAR(4) NOT NULL,  
    FOREIGN KEY (CUST_ID) REFERENCES CUST(CUST_ID)  
);
```

```
CREATE TABLE EVENTS(  
    EVENT_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    NAME VARCHAR(100) NOT NULL,  
    EVENT_DATE DATE NOT NULL,  
    EVENT_TIME TIME NOT NULL,  
    EVENT_TYPE VARCHAR(50) NOT NULL,  
    TOTAL_CAPACITY INT NOT NULL,  
    VIP_CAPACITY INT NOT NULL,  
    GENERAL_CAPACITY INT NOT NULL  
);
```

```
CREATE TABLE RESERVATIONS(  
    RESERV_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    CUST_ID INT,  
    EVENT_ID INT,  
    NUMBER_OF_TICKETS_GEN INT NOT NULL,  
    NUMBER_OF_TICKETS_VIP INT NOT NULL,  
    RESERV_DATE DATE NOT NULL,  
    RESERV_COST DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (CUST_ID) REFERENCES CUST(CUST_ID),  
    FOREIGN KEY (EVENT_ID) REFERENCES EVENTS(EVENT_ID)  
);
```

```
CREATE TABLE VIP_TICKETS(  
    VIPTICKETS_ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    RESERV_ID INT,  
    EVENT_ID INT,  
    SEAT_NO VARCHAR(10) NOT NULL,  
    TICKET_COST DECIMAL(10, 2) NOT NULL,
```

```

    AVAILABILITY BOOLEAN NOT NULL,

    FOREIGN KEY (RESERV_ID) REFERENCES RESERVATIONS(RESERV_ID),

    FOREIGN KEY (EVENT_ID) REFERENCES EVENTS(EVENT_ID)

);

```

```

CREATE TABLE GENERAL_TICKETS(

    GENERALTICKETS_ID INTEGER PRIMARY KEY AUTOINCREMENT,

    RESERV_ID INT,

    EVENT_ID INT,

    SEAT_NO VARCHAR(10) NOT NULL,

    TICKET_COST DECIMAL(10, 2) NOT NULL,

    AVAILABILITY BOOLEAN NOT NULL,

    FOREIGN KEY (RESERV_ID) REFERENCES RESERVATIONS(RESERV_ID),

    FOREIGN KEY (EVENT_ID) REFERENCES EVENTS(EVENT_ID)

);

```

Περιορισμούς ακεραιότητας και συναρτησιακές εξαρτήσεις

CUSTOMER_ID → EMAIL NAMESURNAME

CUSTOMER_ID → CREDIT_CARD

RESERVATION_ID → CUSTOMER_ID EVENT_ID GENERAL_TICKET_COUNT

VIP_TICKET_COUNT DATE TOTAL_COST

TICKET_ID → EVENT_ID RESERVATION_ID SEAT_NUM PRICE AVAILABILITY

EVENT_ID → NAMEDATETIMETYPETOTAL_CAPACITY VIP_CAPACITY

GENERAL_CAPACITY

Καθορισμό κλειδιών των σχέσεων βάσει των συναρτησιακών εξαρτήσεων

CUSTOMER_ID

RESERVATION_ID

TICKET_ID

EVENT_ID

Μετατροπή του μοντέλου σε τρίτη κανονική μορφή με διατήρηση των συναρτησιακών εξαρτήσεων και χωρίς απώλεια πληροφορίας

Αποτελεί τρίτη κανονική μορφή, από τις συναρτησιακές εξαρτήσεις.

Περιγραφή των ερωτήσεων προς τη βάση δεδομένων με SQL

1)

SELECT

E.NAME AS Event_Name,

E.TOTAL_CAPACITY- (

IFNULL(SUM(R.NUMBER_OF_TICKETS_GEN), 0) + IFNULL(SUM(R.NUMBER_OF_TICKETS_VIP), 0)

) AS Available_Seats,

IFNULL(SUM(R.NUMBER_OF_TICKETS_GEN), 0) + IFNULL(SUM(R.NUMBER_OF_TICKETS_VIP), 0) AS
Reserved_Seats

FROM

EVENTS E

LEFT JOIN

RESERVATIONS R ON E.EVENT_ID = R.EVENT_ID

GROUPBY

E.EVENT_ID;

2)

SELECT E.NAME, SUM(R.RESERV_COST)

FROM EVENTS E

JOIN RESERVATIONS R ON E.EVENT_ID = R.EVENT_ID

GROUPBYE.NAME;

3)

SELECT E.NAME

FROM RESERVATIONS R

```
JOIN EVENTS E ON E.EVENT_ID = R.EVENT_ID
```

```
GROUP BY E.NAME
```

```
ORDER BY COUNT(R.RESERV_ID) DESC
```

```
LIMIT 1;
```

4)

```
SELECT
```

```
    E.EVENT_ID,
```

```
    E.NAME AS EVENT_NAME,
```

```
    SUM(R.RESERV_COST) AS TOTAL_REVENUE
```

```
FROM
```

```
    EVENTS E
```

```
JOIN
```

```
    RESERVATIONS R ON E.EVENT_ID = R.EVENT_ID
```

```
WHERE
```

```
    R.RESERV_DATE BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'
```

```
GROUP BY
```

```
    E.EVENT_ID, E.NAME
```

```
ORDER BY
```

```
    TOTAL_REVENUE DESC
```

```
LIMIT 1;
```

5)

```
SELECT
```

```
    RESERVATIONS.RESERV_ID,
```

```
    CUST.NAME AS CUSTOMER_NAME,
```

```
    EVENTS.NAME AS EVENT_NAME,
```

```
    RESERVATIONS.NUMBER_OF_TICKETS_GEN,
```

```
    RESERVATIONS.NUMBER_OF_TICKETS_VIP,
```

```
    RESERVATIONS.RESERV_COST,
```

```
    RESERVATIONS.RESERV_DATE
```

```
FROM
```

```
    RESERVATIONS
```


JOIN

CUST ON RESERVATIONS.CUST_ID = CUST.CUST_ID

JOIN

EVENTS ON RESERVATIONS.EVENT_ID = EVENTS.EVENT_ID

WHERE

RESERVATIONS.RESERV_DATE BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'

ORDER BY

RESERVATIONS.RESERV_DATE;

6)

SELECT

E.NAME AS Event_Name,

COALESCE(SUM(CASE WHEN V.AVAILABILITY = FALSE THEN V.TICKET_COST ELSE 0 END), 0) AS

VIP_Revenue,

COALESCE(SUM(CASE WHEN G.AVAILABILITY = FALSE THEN G.TICKET_COST ELSE 0 END), 0) AS

General_Revenue,

COALESCE(SUM(CASE WHEN V.AVAILABILITY = FALSE THEN V.TICKET_COST ELSE 0 END), 0) +

COALESCE(SUM(CASE WHEN G.AVAILABILITY = FALSE THEN G.TICKET_COST ELSE 0 END), 0) AS

Total_Revenue

FROM

EVENTS E

LEFT JOIN

RESERVATIONS R ON E.EVENT_ID = R.EVENT_ID

LEFT JOIN

VIP_TICKETS V ON R.RESERV_ID = V.RESERV_ID

LEFT JOIN

GENERAL_TICKETS G ON R.RESERV_ID = G.RESERV_ID

GROUPBY

E.EVENT_ID

UNION

SELECT

'Total' AS Event_Name,

```

        COALESCE(SUM(CASE WHEN V.AVAILABILITY = FALSE THEN V.TICKET_COST ELSE 0 END), 0) AS
VIP_Revenue,

        COALESCE(SUM(CASE WHEN G.AVAILABILITY = FALSE THEN G.TICKET_COST ELSE 0 END), 0) AS
General_Revenue,

        COALESCE(SUM(CASE WHEN V.AVAILABILITY = FALSE THEN V.TICKET_COST ELSE 0 END), 0) +
        COALESCE(SUM(CASE WHEN G.AVAILABILITY = FALSE THEN G.TICKET_COST ELSE 0 END), 0) AS
Total_Revenue

FROM

    EVENTS E

LEFT JOIN

    RESERVATIONS R ON E.EVENT_ID = R.EVENT_ID

LEFT JOIN

    VIP_TICKETS V ON R.RESERV_ID = V.RESERV_ID

LEFT JOIN

    GENERAL_TICKETS G ON R.RESERV_ID = G.RESERV_ID;

```

Περιγραφή σε ψευδοκώδικα των διαδικασιών

1)

Διαδικασία Εγγραφή νέου πελάτη: Καταχώριση των απαραίτητων στοιχείων πελάτη.

- Δημιούργησε μια νέα καταχώρηση στον πίνακα **CUSTOMERS** με τα εξής στοιχεία:
CUSTOMER_ID: Καθορίζεται αυτόματα από την βάση
NAME: Το όνομα του πελάτη
SURNAME: Το επίθετο του πελάτη
EMAIL: Το email του πελάτη
- Για κάθε πιστωτική κάρτα που θέλει να εισάγει ο πελάτης δημιούργησε μια νέα καταχώρηση στον πίνακα **CREDIT_CARDS** με τα εξής στοιχεία:
CUSTOMER_ID: Το ίδιο CUSTOMER_ID απο τον πίνακα CUSTOMERS
CREDIT_CARD: Τα στοιχεία της πιστωτικής κάρτας

2)

Δημιουργία νέας εκδήλωσης: Εισαγωγή νέας εκδήλωσης από το σύστημα.

- Δημιούργησε μια νέα καταχώρηση στον πίνακα **EVENTS** με τα εξής στοιχεία:

EVENT_ID: Καθορίζεται αυτόματα από την βάση.

NAME: Το όνομα της εκδήλωσης

DATE: Ημερομηνία που λαμβάνει χώρα η εκδήλωση.

TIME: Ώρα που λαμβάνει χώρα η εκδήλωση.

TYPE: Ο τύπος της εκδήλωσης

TOTAL_CAPACITY: Η συνολική χωρητικότητα της εκδήλωσης.

VIP_CAPACITY: Η χωρητικότητα για VIP εισιτήρια.

GENERAL_CAPACITY: Η χωρητικότητα για εισιτήρια γενικής εισόδου.

- Με βάση τις χωρητικότητες κάθε εισόδου, δημιούργησε στον αντίστοιχο πίνακα τα εισιτήρια της εκδήλωσης.

Συγκεκριμένα, στον πίνακα **VIP_TICKETS** δημιούργησε όσες καταχωρήσεις όσο και η τιμή του **VIP_CAPACITY** με τα παρακάτω στοιχεία:

TICKET_ID: Καθορίζεται αυτόματα από την βάση.

EVENT_ID: Το ίδιο **EVENT_ID** από τον πίνακα **EVENTS**

RESERVATION_ID: Αρχικοποίησε με την τιμή **NULL**

SEAT_NUMBER: Καθορίζεται αυτόματα από την βάση.

PRICE: Η τιμή του κάθε εισιτηρίου.

AVAILABILITY: Αρχικοποίησε με την τιμή 1.

Στον πίνακα **GENERAL_TICKETS** δημιούργησε όσες καταχωρήσεις όσο και η τιμή του **GENERAL_CAPACITY** με τα παρακάτω στοιχεία:

TICKET_ID: Καθορίζεται αυτόματα από την βάση.

EVENT_ID: Το ίδιο **EVENT_ID** από τον πίνακα **EVENTS**

RESERVATION_ID: Αρχικοποίησε με την τιμή **NULL**

SEAT_NUMBER: Καθορίζεται αυτόματα από την βάση.

PRICE: Η τιμή του κάθε εισιτηρίου.

AVAILABILITY: Αρχικοποίησε με την τιμή 1.

3)

Διαδικασία Αναζήτηση_Διαθέσιμων_Θέσεων(Εκδήλωση_ID):

1. Ανάκτησε την εκδήλωση με το συγκεκριμένο EVENT_ID από τον πίνακα EVENTS.
2. Ανάκτησε τα διαθέσιμα VIP και General εισιτήρια για την εκδήλωση:
 - Ανάκτησε τα δεδομένα από τους πίνακες VIP_TICKETS και GENERAL_TICKETS.
3. Για κάθε τύπο εισιτηρίου:
 - Αν το εισιτήριο είναι διαθέσιμο (AVAILABILITY = 1), εμφάνισε τις πληροφορίες.
4. Αν δεν υπάρχουν διαθέσιμα εισιτήρια, επιστρέφει μήνυμα "Δεν υπάρχουν διαθέσιμες θέσεις".

4)

Κράτηση εισιτηρίων: Επιλογή από τα διαθέσιμα εισιτήρια, καταχώρηση κράτησης και ολοκλήρωση πληρωμής.

Συναρτηση ΚράτησηΕισιτηρίων(customer_id, ev_id, αριθμός_εισιτηρίων_γενικής, αριθμός_vip_εισιτηρίων){

 διαθέσιμα_γενικής= Βρες_Εισιτήρια_event(ev_id, τύπος_θέσης=γενική, TRUE); //επιστρέφει τον αριθμό

των διαθέσιμων εισιτηρίων γενικής εισόδου με διαθεσιμότητα=TRUE

 διαθέσιμα_vip= Βρες_Εισιτήρια_event(ev_id, τύπος_θέσης=vip, TRUE);

 AN((διαθέσιμα_γενικής<αριθμός_εισιτηρίων_γενικής) || (διαθέσιμα_vip<αριθμός_vip_εισιτηρίων)){

 ΓΡΑΨΕ "ΑΝΕΠΑΡΚΗΣ ΔΙΑΘΕΣΙΜΟΤΗΤΑ!!";

 return-1;

 }

 κόστος_κράτησης= (αριθμός_εισιτηρίων_γενικής * Τιμή_εισιτηρίου(ev_id, τύπος_θέσης=γενική) +

(αριθμός_vip_εισιτηρίων * Τιμή_εισιτηρίου(ev_id, τύπος_θέσης=vip)));

 new_reserv_id= Βρες_τελευταίο_reservation_id() +1;

 εισ=διαθέσιμα_γενικής;

 ΓΙΑ i ΑΠΟ 1 ΕΩΣ ΚΑΙ αριθμός_εισιτηρίων_γενικής{

 εισ->διαθεσιμότητα= FALSE;

 εισ->reservation_id= new_reserv_id;

 εισ= εισ->next;

 }

 εισ= διαθέσιμα_vip;

 ΓΙΑ i ΑΠΟ 1 ΕΩΣ ΚΑΙ αριθμός_vip_εισιτηρίων{

 εισ->διαθεσιμότητα= FALSE;

 εισ->reservation_id= new_reserv_id;

 εισ= εισ->next;

 }

 ΠΡΟΣΘΕΣΕ_ΣΤΟΝ_ΠΙΝΑΚΑ(RESERVATIONS, RESERV_ID= new_reserv_id, CUST_ID= customer_id,

EVENT_ID= ev_id, NUMBER_OF_TICKETS_GEN= αριθμός_εισιτηρίων_γενικής, NUMBER_OF_TICKETS_VIP=

αριθμός_vip_εισιτηρίων, RESERV_DATE= Date(Σήμερα), RESERV_COST= κόστος_κράτησης);

```

        ΓΡΑΨΕ “Η ΚΡΑΤΗΣΗ ΚΑΤΑΧΩΡΗΘΗΚΕ ΕΠΙΤΥΧΩΣ!”;

        return new_reserv_id;

    }

```

5)

Ακύρωση κράτησης: Διαγραφή της κράτησης, με όρους επιστροφής χρημάτων ή χρέωση ακύρωσης.

```

Συνάρτηση ΑκύρωσηΚράτησης(reserv_id_tokill){

    εισιτήρια_γενικής= Βρες_Εισιτήρια_reserv(reserv_id_tokill, τύπος_θέσης=γενική, FALSE);

    εισιτήρια_vip= Βρες_Εισιτήρια_reserv(reserv_id_tokill, τύπος_θέσης=vip, FALSE);

    ΑΝ((εισιτήρια_γενικής==NULL)&&(εισιτήρια_vip==NULL)){

        ΓΡΑΨΕ “Η ΚΡΑΤΗΣΗ ΔΕΝ ΒΡΕΘΗΚΕ!!!”;

        return -1;

    }

    ΟΣΟ(εισιτήρια_γενικής->next != NULL) ΕΠΑΝΕΛΑΒΕ{

        εισ= εισιτήρια_γενικής;

        εισ->διαθεσιμότητα= TRUE;

        εισ->reservation_id= NULL;

        εισ= εισ->next;

    }

    ΑΝ(εισιτήρια_γενικής != NULL){

        εισ= εισιτήρια_γενικής;

        εισ->διαθεσιμότητα= TRUE;

        εισ->reservation_id= NULL;

    }

    ΟΣΟ(εισιτήρια_vip->next != NULL) ΕΠΑΝΕΛΑΒΕ{

        εισ= εισιτήρια_vip;

        εισ->διαθεσιμότητα= TRUE;

        εισ->reservation_id= NULL;

        εισ= εισ->next;

    }

    ΑΝ(εισιτήρια_vip != NULL){

        εισ= εισιτήρια_vip;

        εισ->διαθεσιμότητα= TRUE;

        εισ->reservation_id= NULL;

    }

    ΑΦΑΙΡΕΣΕ_ΑΠΟ_ΤΟΝ_ΠΙΝΑΚΑ(RESERVATIONS, RESERV_ID= reserv_id_tokill);

    ΓΡΑΨΕ “Η ΚΡΑΤΗΣΗ ΑΚΥΡΩΘΗΚΕ ΜΕ ΕΠΙΤΥΧΙΑ!”;
}

```

```
        return 0;  
    }
```

6)

Διαδικασία Ακύρωση_Εκδήλωσης(Εκδήλωση_ID):

1. Ανάκτησε όλες τις κρατήσεις (RESERVATIONS) για την εκδήλωση από τον πίνακα RESERVATIONS με το συγκεκριμένο EVENT_ID.
2. Για κάθε κράτηση:
 - Ανάκτησε τον πελάτη (CUST_ID) και το κόστος της κράτησης (RESERV_COST).
 - Υπολόγισε το ποσό επιστροφής χρημάτων και ενημέρωσε τον πελάτη:
 - Επιστροφή χρημάτων στον πελάτη μέσω πληρωμής.
 - Σημείωσε την επιστροφή χρημάτων στην βάση δεδομένων
3. Διαγραφή ή ακύρωση των εισιτηρίων για την εκδήλωση:
 - Διαγραφή ή ενημέρωση των VIP_TICKETS και GENERAL_TICKETS για την εκδήλωση ώστε να επιστραφούν οι θέσεις.
4. Ενημέρωσε το σύστημα ότι η εκδήλωση έχει ακυρωθεί και όλες οι κρατήσεις για αυτήν δεν ισχύουν πια.