

Proiectarea unei Baze de Date Aeroporturilor din România

Realizat de: Lăzăroiu Teodora-Bianca

Contents

1	Descrierea modelului	2
2	Constrângeri	2
3	Entități	3
4	Relații	4
5	Atribute	5
6	Diagrama Entitate-Relație	9
7	Diagrama Conceptuală	9
8	Schema relațională	10
9	Realizarea normalizărilor	10
9.1	Forma FN1	10
9.2	Forma FN2	11
9.3	Forma FN3	11
10	Crearea și inserarea datelor	12
11	Cereri SQL	16
11.1	Exercițiul 1	16
11.2	Exercițiul 2	16
11.3	Exercițiul 3	17
11.4	Exercițiul 4	18
11.5	Exercițiul 5	18
12	Actualizarea și suprimarea datelor	19
13	Secvențe pentru înregistrări	20
16	Cerere outer-join	20

1 Descrierea modelului

În acest paragraf vor fi prezentate descrierea modelului real al bazei de date ce urmează a fi proiectată, utilitățile acestuia dar și reguli sale de funcționare.

Modelul de date va gestiona aeroporturile din România și zborurile lor comerciale și interne, ceea ce înseamnă că, pentru crearea acestui model, se vor ignora zborurile în afara țării. Fiecare aeroport se află într-un oraș, iar orașul la rândul său va putea fi încadrat într-o regiune. De asemenea, la fiecare aeroport sunt angajați piloți și însoțitori de zbor care vor reprezenta echipajul necesar călătoriilor desfășurate. Un pilot sau însoțitor poate fi angajat la un singur aeroport. O companie aeriană deține și operează diferite modele de avioane. Se menționează că între aeroport și avioane nu există nicio legătură deoarece aeroporturile nu dețin propriile aeronave ci doar primesc o chirie plătită de firmele aeriene pentru a opera pe acel aeroport.

Un pasager își poate rezerva un loc pentru o călătorie iar, în cadrul unei rezervări, clientul trebuie să își aleagă clasa la care dorește să călătorească. Bineînțeles, anumite clase vor avea un preț adițional, separat de prețul biletului. Așadar pentru o călătorie se vor cunoaște aeroportul de plecare, cel de sosire, cei doi piloți și avionul pe care aceștia îl vor opera, data și ora decolării și a sosirii. Deoarece se gestionează doar curse interne nu există curse cu escală și sunt necesari exact doi piloți. Unui zbor pot să îi fie atașați un anumit număr de însoțitori de zbor, însă nu este obligatoriu. În cazul în care există cel puțin un însoțitor de zbor, pentru zborul respectiv va fi desemnat un șef de cabină. Acesta va fi unul dintre însoțitorii de zbor din acea călătorie.

În cadrul acestui model de date vor fi folosite date reale pentru gestionarea aeroporturilor, a orașelor și a regiunilor. Pentru o călătorie ce pleacă dintr-un anumit aeroport se vor folosi doar piloți și însoțitori de zbor din respectivul aeroport și se va neglija întoarcerea acestora la aeroportul de plecare printr-un alt zbor. Pretul unui bilet pentru o călătorie va fi separat de prețul adițional pentru o clasă de zbor superioară, așadar prețul unui bilet va fi identic pentru toți pasagerii unei călătorii.

2 Constrângeri

Pentru modelul ce urmează a fi proiectat există următoarele restricții de funcționare:

- Orice pilot sau însoțitor de zbor poate lucra pentru un singur aeroport
- Într-un oraș poate exista un singur aeroport
- Biletele pentru zboruri se pot achiziționa doar prin intermediul unei rezervări făcută personal, iar o persoană nu poate face o rezervare decât pentru el însuși
- O călătorie trebuie să se desfășoare între două aeroporturi distincte
- Orice zbor trebuie să aibă exact doi piloți distincti
- Data și ora decolării unui avion trebuie să fie mai mare decât data și ora aterizării
- Numărul de rezervări pentru o călătorie nu va depăși capacitatea avionului folosit
- Numărul de rezervări pentru o clasă nu va depăși capacitatea clasei din acel avion
- Șeful de cabină al unui zbor trebuie să fie un însoțitor de zbor pentru acea călătorie

3 Entități

Pentru baza de date ce gestionează aeroporturile din România vom avea entitățile: *AEROPORTURI*, *ORAȘE*, *REGIUNI*, *PILOȚI*, *ÎNȘOȚITORI*, *CĂLĂTORII*, *AVIOANE*, *CLASE*, *PASAGERI* și *COMPANII*. În această secțiune vor fi prezentate entitățile, dându-se o descriere a acestora și menționându-se pentru fiecare cheia primară.

AEROPORTURI = ansambluri constituite din terenul, clădirile și instalațiile necesare decolării, aterizării, manevrării, adăpostirii și întreținerii avioanelor. Acestea pot avea ca și angajați piloți și însoțitori de zbor. Dintr-un aeroport pot pleca sau ajunge zboruri. Cheia primară a acestei entități este *cod_IATA*.

ORAȘE = forme complexe de așezări umane. În fiecare oraș își are locația un singur aeroport sau niciunul iar un oraș ca aparține mereu de o regiune. Cheia primară a acestei entități este *cod_poștal*.

REGIUNI = unități teritoriale întinse din cadrul României repartizate în funcție de coordonatele geografice, din care fac parte mai multe orașe. Cheia primară a acestei entități este *cod_regiune*.

PILOȚI = persoane fizice calificate care conduc o aeronavă și sunt angajați la un aeroport. Ei pilotează avioanele în cadrul unei călătorii. Cheia primară a acestei entități este *id_pilot*.

ÎNȘOȚITORI = persoane fizice calificate ce oferă pasagerilor măsuri de siguranță la bordul aeronavei și totodată deservesc pasagerii cu gustări și băuturi. Însoțitorii de zbor sunt angajați la un aeroport și pot fi atașați unui anumit zbor. Cheia primară a acestei entități este *id_însoțitor_de_zbor*.

CĂLĂTORII = deplasări aeriene desfășurate între două aeroporturi la care participă pasageri. Un bilet de călătorie necesită o rezervare a unui pasager. Pentru o călătorie sunt necesari exact doi piloți. Cheia primară a acestei entități este *cod_călătorie*.

AVIOANE = vehicule aeriene care se menține în aer datorită unor aripi și se deplasează cu ajutorul motoarelor. Un avion este deținut și operat de o singură companie aeriană. Un avion poate avea mai multe clase. Cheia primară a acestei entități este *cod_avion*.

CLASE = repartiții ale locurilor dintr-o aeronavă în funcție de facilitățile oferite clienților. Clasele superioare au un preț adițional iar fiecare pasager este obligat să își aleagă o clasă la rezervarea unei călătorii. Cheia primară a acestei entități este *cod_clasă*.

PASAGERI = persoane fizice care călătoresc folosind un avion. Pasageri trebuie să își facă rezervări pentru a putea călătorii cu avionul, iar în cadrul rezervărilor trebuie să își aleagă o clasă. Cheia primară a acestei entități este *id_pasager*.

COMPANII = firme ce dețin și operează aeronave oferind în același timp servicii de transport pentru persoane fizice. O companie poate să dețină mai multe avioane care vor fi operate pe aeroporturile din România. Cheia primară a acestei entități este *cod_companie*.

4 Relații

În continuare se vor prezenta relațiile dintre entitățile enumerate mai sus și se va da o descriere a acestora. Pentru fiecare relație se va preciza cardinalitatea minimă și maximă.

AEROPORTURI_se_aflăîn_ORAȘE = relație care leagă entitățile AEROPORTURI și ORAȘE, evidențiind legătura dintre acestea (un aeroport se află într-un singur oraș iar într-un oraș poate fi cel mult un aeroport). Relația are cardinalitatea minimă egală cu 0:1 și cea maximă egală cu 1:1.

REGIUNI_cuprind_ORAȘE = relație care leagă entitățile REGIUNI și ORAȘE, evidențiind legătura dintre acestea (o regiune poate cuprinde mai multe orașe iar un oraș se află într-o singură regiune). Relația are cardinalitatea minimă egală cu 1:0 și cea maximă egală cu 1:n.

PILOȚI_lucreează_la_AEROPORTURI = relație care leagă entitățile PILOȚI și AEROPORTURI, evidențiind legătura dintre acestea (un pilot lucrează la un singur aeroport iar un aeroport poate avea angajați mai mulți piloți). Relația are cardinalitatea minimă egală cu 0:1 și cea maximă egală cu n:1.

ÎNSOȚITORI_lucreează_la_AEROPORTURI = relație care leagă entitățile ÎNSOȚITORI și AEROPORTURI, evidențiind legătura dintre acestea (însoțitorii lucrează la un aeroport și la un aeroport pot lucra mai mulți însoțitori). Relația are cardinalitatea minimă egală cu 0:1 și cea maximă egală cu n:1.

ÎNSOȚITORI_însoțesc_CĂLĂTORII = relație care leagă entitățile ÎNSOȚITORI și CĂLĂTORII, evidențiind legătura dintre acestea (însoțitorii pot însoții pasagerii în timpul călătoriilor însă pentru o călătorie nu sunt neapărat necesari însoțitori de zbor). Relația are cardinalitatea minimă egală cu 0:0 și cea maximă egală cu n:n.

CĂLĂTORII_au_PILOȚI = relație care leagă entitățile CĂLĂTORII și PILOȚI, evidențiind legătura dintre acestea (pentru o călătorie este nevoie de exact doi piloți). Relația are cardinalitatea minimă egală cu 0:2 și cea maximă egală cu n:2.

CĂLĂTORII_se_desfășoarăîntre_AEROPORTURI = relație care leagă entitățile CĂLĂTORII și AEROPORTURI, evidențiind legătura dintre acestea (o călătorie se desfășoară între două aeroporturi). Relația are cardinalitatea minimă egală cu 0:2 și cea maximă egală cu n:2.

AVIOANE_se_folosesc_la_CĂLĂTORII = relație care leagă entitățile AVIOANE și CĂLĂTORII, evidențiind legătura dintre acestea (pentru o călătorie este folosit un avion). Relația are cardinalitatea minimă egală cu 0:1 și cea maximă egală cu 1:n.

AVIOANE_au_CLASE = relație care leagă entitățile AVIOANE și CLASE, evidențiind legătura dintre acestea (avioanele au cel puțin o clasă). Relația are cardinalitatea minimă egală cu 0:1 și cea maximă egală cu n:n.

AVIOANE_deținute_de_COMPANII = relație care leagă entitățile AVIOANE și COMPANII, evidențiind legătura dintre acestea (o companie poate deține mai multe avioane). Relația are cardinalitatea minimă egală cu 0:1 și cea maximă egală cu n:1.

PASAGERII_rezervă_CLASE_pentru_CĂLĂTORII = relație de tip 3 care leagă entitățile PASAGERII, CLASE și CĂLĂTORII, evidențiind legătura dintre acestea (pasagerii pot rezerva o anumită clasă din avion pentru o călătorie). Denumirea relației va fi *rezervă*.

5 Atribute

Pentru fiecare entitate a modelului de date se vor prezenta atributele sale și descrierea acestora, incluzând tipul de date și eventualele constrângeri, valori implicite sau valori posibile.

- Entitatea *AEROPORTURI* are ca atribute:

cod_IATA = variabilă de tip caractere, de lungime 3, care reprezintă un identificator de locație unic pentru fiecare aeroport, dat de Asociația Internațională de Transport Aerian. Valorile vor reprezenta codurile celor 15 aeroporturi publice și comerciale din România.

cod_poștal = variabilă de tip caractere, de lungime 6, care reprezintă un identificator unic pentru un anumit oraș sau o anumită zonă.

nume = variabilă de tip caractere, de lungime maxima 55, care reprezintă numele aeroportului

adresă = variabilă de tip caractere, de lungime maxima 50, care reprezintă strada și numărul aeroportului

telefon = variabilă de tip caractere, de lungime maxima 10, care reprezintă numărul de telefon al aeroportului

- Entitatea *ORAȘE* are ca atribute:

cod_poștal = variabilă de tip caractere, de lungime 6, care reprezintă un identificator unic pentru un anumit oraș sau o anumită zonă.

nume = variabilă de tip caractere, de lungime maxima 15, care reprezintă numele orașului. Acestea vor fi: Arad, Bacău, Baia Mare, București, Cluj-Napoca, Constanța, Craiova, Iași, Oradea, Satu Mare, Sibiu, Suceava, Târgu Mureș, Timișoara, Tulcea.

cod_aeroport = variabilă de tip caractere, de lungime 3, care reprezintă codul IATA al aeroportului.

cod_regiune = variabilă de tip caractere, de lungime 3, care reprezintă un cod unic specific unei anumite regiuni din România.

- Entitatea *REGIUNI* are ca atribute:

cod_regiune = variabilă de tip caractere, de lungime 3, care reprezintă un cod unic specific unei anumite regiuni din România.

nume = variabilă de tip caractere, de lungime maxima 15, care reprezintă numele regiunii. Acestea vor fi: Banat, Bucovina, Crișana, Dobrogea, Maramureș, Moldova, Muntenia, Oltenia și Transilvania.

număr_aeroporturi = variabilă de tip întreg, de lungime maxima 2, care reprezintă numărul total de aeroporturi din acea regiune.

- Entitatea *PILOȚI* are ca atribute:

id_pilot = variabilă de tip întreg, de lungime maximă 4, care reprezintă un cod unic pentru un pilot

cod_aeroport = variabilă de tip caractere, de lungime 3, care reprezintă codul IATA al aeroportului unde lucrează un pilot

nume = variabilă de tip caractere, de lungime maxima 30, care reprezintă numele pilotului

prenume = variabilă de tip caractere, de lungime maxima 30, care reprezintă prenumele pilotului

telefon = variabilă de tip caractere, de lungime maxima 10, care reprezintă numărul de telefon al pilotului

data_angajării = variabilă de tip dată calendaristică, care reprezintă data la care a fost angajat pilotul

salariu = variabilă de tip întreg, de lungime maximă 6, care reprezintă salariul anual exprimat în lei al unui pilot.

- Entitatea *ÎNȘOȚITORI* are ca attribute:

id_însoțitor_de_zbor = variabilă de tip întreg, de lungime maximă 4, care reprezintă un cod unic pentru un însoțitor de zbor

cod_aeroport = variabilă de tip caractere, de lungime 3, care reprezintă codul IATA al aeroportului la care lucrează un însoțitor de zbor

nume = variabilă de tip caractere, de lungime maxima 30, care reprezintă numele însoțitorului de zbor

prenume = variabilă de tip caractere, de lungime maxima 30, care reprezintă prenumele însoțitorului de zbor

telefon = variabilă de tip caractere, de lungime maxima 10, care reprezintă numărul de telefon al însoțitorului de zbor

data_angajării = variabilă de tip dată calendaristică, care reprezintă data la care a fost angajat însoțitorul de zbor

salariu = variabilă de tip întreg, de lungime maximă 6, care reprezintă salariul anual exprimat în lei al unui însoțitor de zbor

- Entitatea *CĂLĂTORII* are ca attribute:

cod_călătorie = variabilă de tip întreg, de lungime maximă 6, care reprezintă un cod unic pentru identificarea unui zbor

cod_aeroport_plecare = variabilă de tip caractere, de lungime 3, care reprezintă codul IATA al aeroportului din care începe o călătorie

cod_aeroport_sosire = variabilă de tip caractere, de lungime 3, care reprezintă codul IATA al aeroportului în care se termină o călătorie. Acest atribut trebuie să fie diferit de *cod_aeroport_plecare*

id_pilot_principal = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic al primului pilot din avion.

id_pilot_secundar = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic al celui de al doilea pilot din avion. Acest atribut trebuie să fie diferit de *id_pilot_principal*

cod_avion = variabilă de tip întreg, de lungime maximă 4, care reprezintă un codul unic al avionului folosit pentru călătorie

data_plecării = variabilă de tip dată calendaristică, care reprezintă data și ora la care decolează avionul

data_sosirii = variabilă de tip dată calendaristică, care reprezintă data și ora la care atelizează avionul. Data sosirii trebuie sa fie mai mare decât data plecării.

- Entitatea *AVIOANE* are ca attribute:

cod_avion = variabilă de tip întreg, de lungime maximă 4, care reprezintă un cod unic pentru identificarea unui avion

model = variabilă de tip caractere, de lungime maxima 30, care reprezintă denumirea modelului de avion

cod_companie = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic al companiei ce deține avionul

capacitate = variabilă de tip întreg, de lungime maximă 3, care reprezintă numărul total de locuri destinate pasagerilor din interiorul aeronavei

viteză_de_croazieră = variabilă de tip întreg, de lungime maximă 4, care reprezintă viteza cu care se deplasează avionul când se află la altitudinea de croazieră exprimată în km/h

- Entitatea *CLASE* are ca attribute:

cod_clasă = variabilă de tip întreg, de lungime maximă 2, care reprezintă un cod unic pentru identificarea unui clase

nume = variabilă de tip caractere, de lungime maxima 20, care reprezintă denumirea clasei de zbor

preț_aditional = variabilă de tip întreg, de lungime maximă 4, care reprezintă prețul adițional exprimat în lei care trebuie plătit pentru a rezerva un loc la acea clasă

facilitați = variabilă de tip caractere, de lungime maxima 60, care reprezintă o listă a facilităților oferite de acea clasă

- Entitatea *PASAGERI* are ca attribute:

id_pasager = variabilă de tip întreg, de lungime maximă 6, care reprezintă un cod unic pentru identificarea unui pasager

nume = variabilă de tip caractere, de lungime maxima 30, care reprezintă numele pasagerului

prenume = variabilă de tip caractere, de lungime maxima 30, care reprezintă prenumele pasagerului

email = variabilă de tip caractere, de lungime maxima 30, care reprezintă adresa de email a pasagerului

telefon = variabilă de tip caractere, de lungime maxima 10, care reprezintă numărul de telefon al pasagerului

data_nașterii = variabilă de tip dată calendaristică, care reprezintă data de naștere a pasagerului

- Entitatea *COMPANII* are ca atribute:

cod_companie = variabilă de tip întreg, de lungime maximă 4, care reprezintă un cod unic al companiei aeriene

nume = variabilă de tip caractere, de lungime maximă 30, care reprezintă denumirea companiei aeriene

- Relația *AVIOANE_au_CLASE* are ca atribute:

cod_clasă = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic pentru identificarea unui clase

cod_avion = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic pentru identificarea unui avion

capacitate = variabilă de tip întreg, de lungime maximă 3, care reprezintă numărul total de locuri de la o anumită clasă, într-un anumit avion

- Relația *PASAGERII_rezervă_CLASE_pentru_CĂLĂTORII*, numită *rezervă*, are ca atribute:

cod_clasă = variabilă de tip întreg, de lungime maximă 2, care reprezintă codul unic pentru identificarea unui clase

cod_călătorie = variabilă de tip întreg, de lungime maximă 6, care reprezintă codul unic pentru identificarea unui zbor

id_pasager = variabilă de tip întreg, de lungime maximă 6, care reprezintă codul unic pentru identificarea unui pasager

cod_rezervare = variabilă de tip întreg, de lungime maximă 8, care reprezintă un cod unic prin care se identifică o rezervare

pret_bilet = variabilă de tip întreg, de lungime maximă 4, care reprezintă prețul unui bilet ,exprimat în lei, plătit de un pasager pentru o anumită călătorie, la o clasă aleasă de acesta

- Relația *ÎNȘOȚITORII_însoțesc_CĂLĂTORII*, numită *însoțesc*, are ca atribute:

id_însoțitor_de_zbor = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic pentru un însoțitor de zbor

cod_călătorie = variabilă de tip întreg, de lungime maximă 6, care reprezintă codul unic pentru identificarea unui zbor

șef_cabină = variabilă de tip întreg, de lungime maximă 4, care reprezintă codul unic al însoțitorului de zbor ce este desemnat șef de cabină pentru acel zbor și va fi responsabil pentru restul însoțitorilor de zbor

6 Diagrama Entitate-Relație

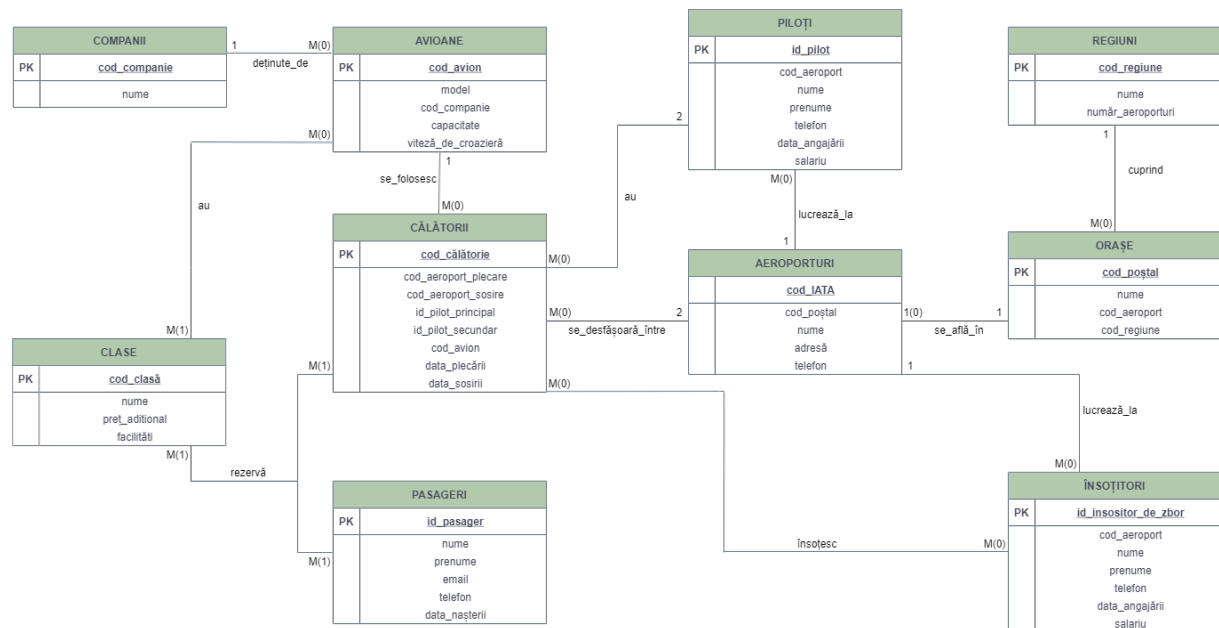


Figure 1: Diagrama Entitate-Relație

7 Diagrama Conceptuală

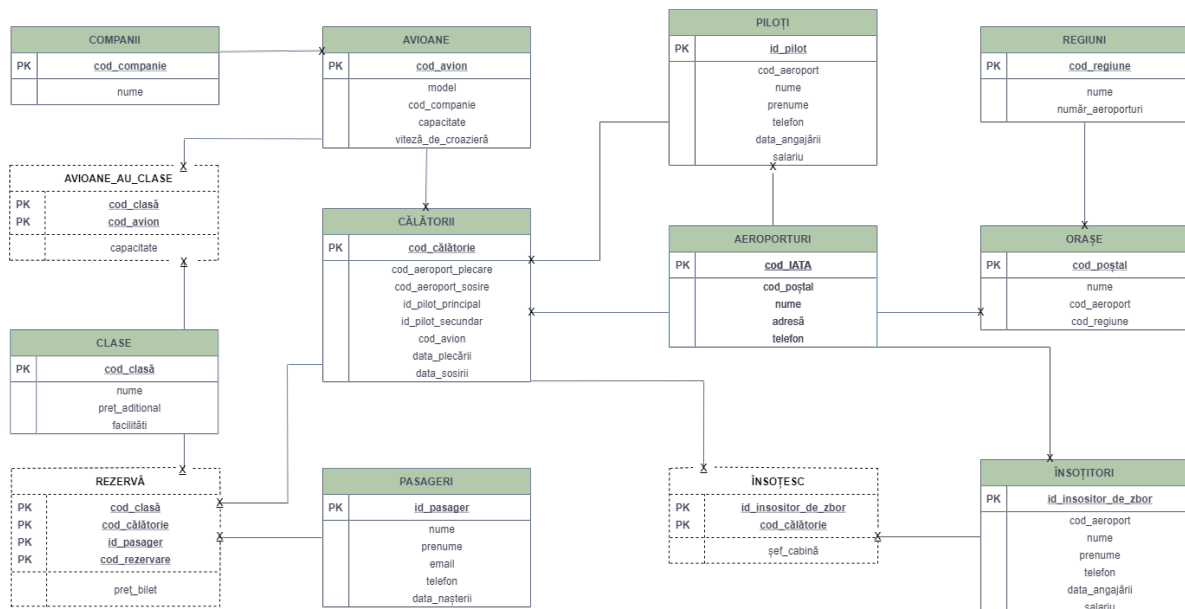


Figure 2: Diagrama Conceptuală

8 Schema relațională

În continuare urmează să fie enumerate schemele relaționale corespunzătoare diagramei conceptuale de la punctul 7, evidențiindu-se cheia primară prin numele atributului urmat de caracterul "#". În aceste scheme se vor enumera toate atributele entităților modelului de date.

- *AEROPORTURI* (cod_IATA#, cod_poștal, nume, adresă, telefon)
- *ORAȘE* (cod_poștal#, nume, cod_aeroport, cod_regiune)
- *REGIUNI* (cod_regiuni#, nume, număr_aeroporturi)
- *PILOȚI* (id_pilot#, cod_aeroport, nume, prenume, telefon, data_angajării, salariu)
- *ÎNȘOȚITORI* (id_însoțitor_de_zbor#, cod_aeroport, nume, prenume, data_angajării, salariu)
- *CĂLĂTORII* (cod_călătorie#, cod_aeroport_plecare, cod_aeroport_sosire, id_pilot_principal, id_pilot_secundar, cod_avion, data_plecării, data_sosirii)
- *AVIOANE* (cod_avion#, model, cod_companie, capacitate, viteză_de_croazieră)
- *CLASE* (cod_clasă#, nume, preț_aditional, facilități)
- *PASAGERI* (id_pasager#, nume, prenume, email, telefon, data_nașterii)
- *COMPANII* (cod_companie#, nume)
- *AVIOANE_AU_CLASE* (cod_clasă#, cod_avion#, capacitate)
- *REZERVĂ* (cod_clasă#, cod_călătorie#, id_pasager#, cod_rezervare#, preț_bilet)
- *ÎNȘOȚESC* (id_însoțitor_de_zbor#, cod_călătorie, șef_cabină)

9 Realizarea normalizărilor

9.1 Forma FN1

Diagrama realizată se află în forma normală FN1 deoarece fiecărui atribut al modelului de date îi corespunde o valoare indivizibilă. Un exemplu de formă NON-FN1 ar putea apărea în tabelul *CĂLĂTORII*, astfel:

#COD_CALATORIE	COD_AEROPORTURI	ID_PILOTI
1001	OTP, CLJ	120, 130
1002	OTP, IAS	120, 130

Se observă că în exemplul de mai sus coloanele *cod_aeroporturi* și *id_piloți* se pot despărți fiecare în două coloane distincte. Modelul corect în forma FN1, ce urmează să fie implementat, este următorul:

#COD_CALATORIE	COD_AEROPORT_PLECAR	COD_AEROPORT_SOSIRE	ID_PILOT_PRINCIPAL	ID_PILOT_SECUNDAR
1001	OTP	CLJ	120	130
1002	OTP	IAS	120	130

9.2 Forma FN2

Modelul de date se află în forma normală FN2 deoarece se află în forma FN1 și fiecărui atribut care nu participă la cheia primară este dependent de întreaga cheie primară. Un exemplu de formă NON-FN2 ar putea apărea în tabelul *AVIOANE_AU_CLASE*, astfel:

#COD_CLASA	#COD_AVION	CAPACITATE	CAPACITATE_AVION
10	1000	100	149
20	1000	49	149

Se poate observa dependența funcțională dintre cheile primare *cod_clasă*, *cod_avion* și atributul *capacitate*, însă atributul *capacitate_avion* este dependent doar de *cod_avion*, ceea ce plasează relația în afara formei FN2. Modelul corect în forma FN2, ce urmează să fie implementat, este următorul:

#COD_CLASA	#COD_AVION	CAPACITATE
10	1000	100
20	1000	49

9.3 Forma FN3

Baza de date se află în forma normală FN3 deoarece se află în forma FN2 și fiecărui atribut care nu este cheie, depinde de cheie, de întreaga cheie și numai de cheie. Un exemplu de formă NON-FN3 poate apărea în tabelul *ORAȘE*, astfel:

#COD_POSTAL	NUME	COD_AEROPORT	COD_REGIUNE	ADRESA_AEROPORT
075150	Bucuresti	OTP	MUN	Calea Bucurestilor
400397	Cluj	CLJ	TRA	Strada Traian Vuia

Observăm că atributul *adresă_aeroport* nu este dependent de *cod_poștal* deoarece nu are legătură cu acesta, ceea ce face ca tabelul să nu fie în FN3. Modelul corect în forma FN3, ce urmează să fie implementat, este următorul:

#COD_POSTAL	NUME	COD_AEROPORT	COD_REGIUNE
075150	Bucuresti	OTP	MUN
400397	Cluj	CLJ	TRA

10 Crearea și inserarea datelor

Mai jos vor fi ilustrate printscreen-uri din aplicația SQL Developer după crearea bazei de date și a constrângerilor sale și a inserării datelor din cele 13 tabele

AEROPORTURI:

	COD_IATA	COD_POSTAL	NUME	ADRESA	TELEFON
1	OTP	75150	Henri Coanda International Airport	Calea Bucurestilor 224E	0212041000
2	CLJ	400397	Avram Iancu International Airport	Strada Traian Vuia 149-151	0264307500
3	IAS	700750	Iasi International Airport	Strada Moara de Vant 34	0232271590
4	CND	907195	Mihail Kogalniceanu International Airport	Strada Tudor Vladimirescu	0241255100
5	TSR	307200	Traian Vuia International Airport	Strada Aeroportului 2	0256386089
6	OMR	410223	Oradea International Airport	Calea Aradului 80	0259416082
7	CRA	200621	Craiova International Airport	Calea Bucuresti 325	0754022508
8	SBZ	550052	Sibiu International Airport	Soseaua Alba Iulia 73	0269253135

ORAȘE:

	COD_POSTAL	NUME	COD_AEROPORT	COD_REGIUNE
1	75150	Bucuresti	OTP	MUN
2	400397	Cluj	CLJ	TRA
3	700750	Iasi	IAS	MOL
4	907195	Constanta	CND	DOB
5	307200	Timisoara	TSR	BAN
6	410223	Oradea	OMR	CRI
7	200621	Craiova	CRA	OLT
8	550052	Sibiu	SBZ	TRA
9	110001	Pitesti	(null)	MUN

REGIUNI:

	COD_REGIUNE	NUME	NUMAR_AEROPORTURI
1	MUN	Muntenia	1
2	TRA	Transilvania	2
3	MOL	Moldova	1
4	DOB	Dobrogea	1
5	BAN	Banat	1
6	OLT	Oltenia	1
7	CRI	Crisana	1
8	MAR	Maramures	0

PILOȚI:

	ID_...	COD_AEROPORT	NUME	PRENUME	TELEFON	DATA_ANGAJARII	SALARIU
1	120	OTP	Popescu	Mihai	0723467897	04-MAY-18	100000
2	130	OTP	Dumitru	Andrei	0773567890	07-SEP-18	150000
3	140	TSR	Marinescu	Adrian	0793857856	05-FEB-16	550000
4	150	TSR	Coman	Stefania	0775852235	05-DEC-20	200000
5	160	CLJ	Florescu	Mihai	0788451775	07-DEC-10	500000
6	170	CLJ	Moise	Horia	0788449075	23-JUN-15	350000
7	180	CLJ	Stoica	Irina	0702390175	15-JAN-21	180000
8	190	IAS	Naum	Florin	0712345175	15-NOV-19	250000
9	200	IAS	Olteanu	Paul	0712743275	01-JAN-14	550000
10	210	CND	Avram	Daniel	0766320275	01-JUN-20	222000
11	220	CND	Stefan	George	0766320275	30-AUG-13	355000
12	230	CND	Stancu	Elena	0733332146	01-MAY-21	99000
13	240	CRA	Mihailescu	Cosmin	0754999146	01-MAY-18	340000
14	250	CRA	Olteanu	Valentin	0765412146	17-APR-10	450000

CĂLĂTORII:

	✧ COD_CALATORIE	✧ COD_AEROPORT_PLECARE	✧ COD_AEROPORT_SOSIRE	✧ ID_PILOT_PRINCIPAL	✧ ID_PILOT_SECUNDAR	✧ COD_AVION	✧ DATA_PLECĂRII	✧ DATA_SOSIRII
1	1001	OTP	CLJ	120	130	1006	01-JUN-21	01-JUN-21
2	1002	OTP	IAS	120	130	1002	02-JUN-21	02-JUN-21
3	1003	OTP	TSR	120	130	1000	03-JUN-21	03-JUN-21
4	1004	OTP	TSR	120	130	1010	04-JUN-21	04-JUN-21
5	1005	CLJ	IAS	160	180	1003	27-MAY-21	27-MAY-21
6	1006	CLJ	CND	160	170	1008	30-MAY-21	31-MAY-21
7	1007	IAS	CRA	190	200	1009	04-JUN-21	04-JUN-21
8	1008	IAS	OTP	190	200	1002	05-JUN-21	05-JUN-21
9	1009	CRA	IAS	240	250	1010	04-JUN-21	04-JUN-21
10	1010	CRA	CLJ	240	250	1001	08-JUN-21	08-JUN-21
11	1011	TSR	OTP	140	150	1005	30-MAY-21	30-MAY-21
12	1012	TSR	CND	140	150	1007	28-MAY-21	28-MAY-21
13	1013	OTP	OMR	120	130	1003	11-JUN-21	11-JUN-21

ÎNSOȚITORI:

	✧ ID_INSOTITOR_DE_ZBOR	✧ COD_AEROPORT	✧ NUME	✧ PRENUME	✧ TELEFON	✧ DATA_ANGAJĂRII	✧ SALARIU
1	10	OTP	Voiculescu	Alina	0745678146	04-OCT-18	33000
2	20	OTP	Moisescu	Denisa	0792218597	13-MAR-19	40000
3	30	OTP	Ionescu	Florin	0726688610	02-APR-12	90000
4	40	CLJ	Georgescu	Calin	0768216950	01-JUL-16	80000
5	50	CLJ	Antonie	Mihaela	0783717731	06-DEC-20	15000
6	60	TSR	Voica	Laura	0766138430	31-JAN-21	13000
7	70	TSR	Enache	Maria	0778767173	25-OCT-15	40000
8	80	IAS	Matache	Irina	0762706994	20-MAR-15	45000
9	90	IAS	Mirea	Mihai	0785072934	07-JUL-19	27000
10	100	CND	Lazar	Nicoleta	0718538567	09-NOV-18	33000
11	110	CND	Florea	Stefania	0784725925	23-SEP-09	80000
12	120	CRA	Miulescu	Ana	0758812546	06-JUN-20	12000
13	130	SBZ	Diaconescu	Daniela	0734456845	27-APR-18	16000

PASAGERI:

	✧ ID_PASAGER	✧ NUME	✧ PRENUME	✧ EMAIL	✧ TELEFON	✧ DATA_NĂȘTERII
1	1	Ionita	Alexandra	aionita@yahoo.com	0741974529	03-MAY-01
2	2	Moldoveanu	Victor	victor_m@yahoo.com	0785483431	08-MAR-80
3	3	Stoica	Valentin	valistoica@gmail.com	0791459679	14-DEC-96
4	4	Dabijsa	Adrian	dabadi@yahoo.com	0760523629	06-JUL-90
5	5	Barbu	Dumitru	dumitruB@yahoo.com	0746888320	13-MAY-00
6	6	Teodorescu	Eugen	eugen_t@yahoo.com	0770522210	23-NOV-65
7	7	Lupu	Maria	maria_lupu@yahoo.com	0757241888	03-DEC-77
8	8	Sava	Florentina	savaflo@gmail.com	0766560779	29-JUN-02
9	9	Nistor	Mariana	mariananistor@gmail.com	0745582866	03-MAY-01
10	10	Barbu	Ana Maria	anamariab@yahoo.com	0788821557	31-JAN-92
11	11	Dumitrescu	Andreea	andreea_dumi@yahoo.com	0749267946	07-JUL-79
12	12	Gheorghiu	Mihaela	mihaela.g@yahoo.com	0727774485	24-NOV-69
13	13	Pop	Rodica	rodica_pop@yahoo.com	0791567138	16-APR-99
14	14	Constantinescu	Roxana	roxana.c@yahoo.com	0743479564	12-DEC-82
15	15	Dobre	Gabriela	gabi_dobre@gmail.com	0765870667	28-FEB-97
16	16	Ciobanu	Adriana	a.ciobanu@yahoo.com	0732936930	03-NOV-60
17	17	Toma	Claudia	claudia.toma@yahoo.com	0736630215	16-SEP-95
18	18	Stanescu	Constantin	ctin_stanescu@gmail.com	0756729015	08-AUG-55
19	19	Dima	Gheorghe	dima.gh@gmail.com	0774034758	16-APR-68
20	20	Dumitrescu	Cristina	cristinaD@gmail.com	0784550324	15-MAY-00

CLASE:

	COD_CLASA	NUME	PRET_ADITIOAL	FACILITATI
1	10	Economica	(null)	masuta retractabila
2	20	Business	300	scaune ajustabile, o masa calda
3	30	First class	500	lounge, meniu variat, paturi
4	40	Business Premium	800	business + lounge
5	50	First Class Premium	1000	first class + meniu lux

AVIOANE_AU_CLASE:

ÎNSOȚESC:

	COD_CLASA	COD_AVION	CAPACITATE		ID_INSOSITOR_DE_ZBOR	COD_CALATORIE	SEF_CABINA
1	10	1000	100	1	10	1001	20
2	20	1000	49	2	20	1001	(null)
3	10	1001	141	3	30	1001	20
4	10	1002	132	4	10	1003	(null)
5	10	1003	52	5	30	1003	10
6	10	1004	80	6	10	1004	20
7	20	1004	61	7	20	1004	(null)
8	10	1005	126	8	40	1005	(null)
9	10	1006	100	9	40	1006	(null)
10	20	1006	40	10	50	1006	40
11	30	1006	49	11	80	1007	90
12	10	1007	160	12	90	1007	(null)
13	20	1007	26	13	90	1008	(null)
14	10	1008	100	14	120	1009	(null)
15	20	1008	100	15	120	1010	(null)
16	30	1008	20	16	60	1011	(null)
17	40	1008	10	17	70	1011	60
18	10	1009	185	18	60	1012	70
19	10	1010	162	19	70	1012	(null)
				20	20	1013	(null)
				21	30	1013	20

COMPANII:

	COD_COMPANIE	NUME
1	120	Tarom
2	121	Blue Air
3	122	Wizz Air
4	123	KLM
5	124	Ryanair
6	125	Lufthansa

AVIOANE:

	COD_AVION	MODEL	COD_COMPANIE	CAPACITATE	VITEZA_DE_CROAZIERA
1	1000	BOEING 737-700	120	149	870
2	1001	BOEING 737-300	120	141	800
3	1002	AIRBUS A318-111	120	132	850
4	1003	ATR 42-500	120	52	550
5	1004	BOEING 737-300	121	141	800
6	1005	BOEING 737-500	121	126	785
7	1006	BOEING 737-800	121	189	823
8	1007	AIRBUS A320-200	122	186	900
9	1008	AIRBUS A321-200	122	230	840
10	1009	AIRBUS A330-200	123	185	913
11	1010	BOEING 737-800	124	162	946

REZERVĂ:

	COD_CLASA	COD_CALATORIE	ID_PASAGER	COD_REZERVARE	PRET_BILET
1	30	1001	2	100001	300
2	30	1001	3	100002	300
3	10	1001	15	100003	300
4	10	1010	7	100004	200
5	10	1010	10	100005	200
6	10	1010	11	100006	200
7	20	1006	5	100007	250
8	40	1006	12	100008	250
9	40	1006	13	100009	250
10	30	1006	14	100010	250
11	20	1012	1	100011	250
12	10	1012	17	100012	250
13	10	1012	18	100013	250
14	10	1012	20	100014	250
15	10	1004	4	100015	270
16	10	1004	6	100016	270
17	10	1003	16	100017	150
18	20	1003	8	100018	150
19	20	1003	9	100019	150
20	10	1013	19	100020	150
21	10	1007	10	100021	240
22	10	1007	11	100022	240
23	10	1011	2	100023	350
24	10	1011	3	100024	350
25	10	1008	19	100025	275
26	10	1008	17	100026	275
27	10	1008	18	100027	275

11 Cereri SQL

11.1 Exercițiul 1

Sa se afiseze numele pasagerilor, pretul total al biletelor rezervate de acestia, aeroportul de plecare si data si ora plecarii pentru pasagerii ce pleaca dintr-un oras din Muntenia sau Oltenia. Pretul total al biletului reprezinta pretul biletului plus pretul aditional al clasei, daca este cazul. Sa se ordoneze rezultatele dupa data plecarii.

```
SELECT p.num "Nume pasager", r.pret_bilet + NVL(cl.pret_aditional,0) "Pret total
bilet", c.cod_aeroport_plecare "Aeroport plecare", to_char(c.data_plecarii
'dd-mm-yyyy hh24:mi') "Data si ora plecarii"
FROM PASAGERI p JOIN REZERVA r USING (id_pasager) JOIN CALATORII c USING
(cod_calatorie) JOIN CLASE cl USING (cod_clasa)
WHERE c.cod_aeroport_plecare IN (SELECT cod_IATA
                                FROM AEROPORTURI a JOIN ORASE o USING (cod_postal)
                                JOIN REGIUNI re USING (cod_regiune)
                                WHERE INITCAP(re.num) = 'Muntenia' or
                                INITCAP(re.num) = 'Oltenia')

ORDER BY c.data_plecarii;
```

Se foloseste functia NVL pentru a putea calcula pretul total: acolo unde pretul aditional pentru clasa este NULL se va aduna valoarea 0. In subcererea nesincronizata sunt selectate aeroporturi care se afla in regiunile Muntenia sau Oltenia. Functia pentru siruri de caractere INITCAP se foloseste pentru a verifica numele regiunilor.

11.2 Exercițiul 2

Sa se afiseze angajatii din regiunea TRANSILVANIA care au salariul peste media tuturor angajatilor din acea regiune. Pentru fiecare angajat se va afisare id-ul, numele, prenumele, salariul si jobul (pilot sau insotitor) si sa se ordoneze crescator dupa salariu

```
WITH angajati_transilvania
AS ((SELECT p.id_pilot "ID_ANGAJAT", p.num, p.prenume, p.salariu "SALARIU",
'pilot' AS "JOB"
FROM PILOTI p JOIN AEROPORTURI a ON (a.cod_IATA = p.cod_aeroport)
WHERE a.cod_postal IN (SELECT cod_postal
                        FROM ORASE o JOIN REGIUNI r USING(cod_regiune)
                        WHERE UPPER(r.num) = 'TRANSILVANIA'))

UNION

(SELECT i.id_insotitor_de_zbor "ID_ANGAJAT", i.num, i.prenume, i.salariu
```



```

        "SALARIU", 'insotitor' AS "JOB"
    FROM INSOTITORI i JOIN AEROPORTURI a ON (a.cod_IATA = i.cod_aeroport)
    WHERE a.cod_postal IN (SELECT cod_postal
                           FROM ORASE o JOIN REGIUNI r USING (cod_regiune)
                           WHERE UPPER(r.ume) = 'TRANSILVANIA'))

SELECT id_angajat, nume, prenume, salariu, job
FROM angajati_transilvania
WHERE salariu > (SELECT AVG(salariu)
                 FROM angajati_transilvania)
ORDER BY salariu;

```

Se va crea tabelul angajati_transilvania folosind clauza WITH pentru a cuprinde atat pilotii cat si insotitorii care lucreaza la un aeroport din Transilvania. Se foloseste operatorul UNION pentru a combina cele 2 categorii. Se foloseste o subcerere sincronizata pentru a compara salariile din noul tabel cu salariul mediu, calculat folosind functia grup AVG.

11.3 Exercițiul 3

Să se afișeze rezervările al căror preț este sub media prețurilor biletelor și să se ordoneze după numele clientului. Pentru fiecare rezervare se va afișa codul rezervării, numele și prenumele pasagerului pe aceeași coloană, numele clasei de zbor pe care acesta a ales-o, prețul biletului și dacă pasagerul are vârsta sub sau peste 25 de ani

```

SELECT r.cod_rezervare "Cod rezervare", CONCAT(CONCAT(p.ume, ' '),p.prenume)
"Nume si prenume pasager", c.ume "Nume clasa", r.pret_bilet "Pret bilet",
CASE WHEN MONTHS_BETWEEN(SYSDATE, data_nasterii)/12 > 25 THEN 'Peste 25 de ani'
ELSE 'Sub 25 de ani'
END AS "Varsta"
FROM PASAGERI p JOIN REZERVA r USING (id_pasager) JOIN CLASE c USING (cod_clasa)
WHERE pret_bilet < (SELECT AVG(pret_bilet)
                   FROM REZERVA)
ORDER BY p.ume;

```

Se folosește expresia CASE pentru a afișa una dintre cele 2 variante de vârstă, adică sub sau peste 25 de ani. Se va face JOIN între cele 3 tabele pentru a putea accesa toate datele și se va folosi funcția grup AVG pentru a calcula prețul mediu al unui bilet.

11.4 Exercițiul 4

Sa se afiseze codul IATA, numele aeroportului, numarul de piloti angajati si salariul maxim din acel aeroport pentru aeroporturile care au mai mult de un pilot.

```
SELECT a.cod_IATA "Cod IATA", a.numa "Nume",  
COUNT(*) "Numar piloti", MAX(p.salariu) "Salariu maxim"  
FROM AEROPORTURI a JOIN PILOTI p ON (cod_IATA = cod_aeroport)  
GROUP BY a.cod_IATA, a.numa  
HAVING COUNT(*) > 1;
```

Se folosesc functiile grup COUNT și MAX pentru a număra piloții angajați și pentru a afla salariul maxim al pilotilor din respectivul aeroport. Se face JOIN intre tabelele AEROPORTURI si PILOTI. Prin expresia HAVING se testeaza daca numarul de piloti angajati din respectivul aeroport este mai mare decat unu.

11.5 Exercițiul 5

Sa se afiseze salariul maxim al pilotilor in aeroporturi cu codul OTP, CLJ si TSR si salariul mediu din Romania al pilotilor

```
SELECT  
MAX(DECODE(cod_aeroport, 'OTP', salariu)) "Maxim OTP",  
MAX(DECODE(cod_aeroport, 'CLJ', salariu)) "Maxim CLJ",  
MAX(DECODE(cod_aeroport, 'TSR', salariu)) "Maxim TSR",  
ROUND(AVG(salariu)) "Salariu mediu pe tara"  
FROM PILOTI;
```

Se folosesc functiile grup AVG și MAX pentru a calcula salariul mediu si cel maxim. Comanda DECODE este folosita pentru a selecta codurile aeroporturilor.

12 Actualizarea și suprimarea datelor

Se vor realiza 3 operații de actualizare sau suprimare a datelor folosind subcereri

1. Să se mărească salariul piloților care au fost cel puțin o dată piloți principali în cadrul unui zbor cu 20% din valoarea actuală.

```
UPDATE PILOTI
SET salariu = salariu * 1.2
WHERE id_pilot IN (SELECT DISTINCT id_pilot_principal
                  FROM CALATORII);
```

2. Să se crească prețul biletelor cu 100 de lei pentru rezervările făcute de pasagerii născuți după anul 2000, inclusiv

```
UPDATE REZERVA
SET pret_bilet = pret_bilet + 100
WHERE id_pasager IN (SELECT id_pasager
                    FROM PASAGERI
                    WHERE EXTRACT(YEAR FROM data_nasterii) >= 2000);
```

3. Să se șteargă companiile care nu operează niciun avion folosit într-o călătorie

```
DELETE FROM COMPANII
WHERE cod_companie NOT IN (SELECT DISTINCT cod_companie
                          FROM AVIOANE
                          WHERE cod_avion IN (SELECT DISTINCT cod_avion
                                              FROM CALATORII));
```

13 Secvențe pentru înregistrări

```
--- SECVENTA PENTRU INCREMENTAREA ATRIBUTULUI ID_PILOT DIN TABELUL PILOTI
CREATE SEQUENCE SEQ_PILOTI
INCREMENT by 10
START WITH 100
MAXVALUE 9990
NOCYCLE;
```

```
--- SECVENTA PENTRU INCREMENTAREA ATRIBUTULUI ID_INSOSITOR_DE_ZBOR DIN TABELUL INSOTITORI
CREATE SEQUENCE SEQ_INSOTITORI
INCREMENT by 10
START WITH 10
MAXVALUE 9990
NOCYCLE;
```

```
--- SECVENTA PENTRU INCREMENTAREA ATRIBUTULUI ID_PASAGER DIN TABELUL PASAGERI
CREATE SEQUENCE SEQ_PASAGERI
INCREMENT by 1
START WITH 1
MAXVALUE 999999
NOCYCLE;
```

```
--- SECVENTA PENTRU INCREMENTAREA ATRIBUTULUI COD_CALATORIE DIN TABELUL CALATORII
CREATE SEQUENCE SEQ_CALATORII
INCREMENT by 1
START WITH 1000
MAXVALUE 999999
NOCYCLE;
```

16 Cerere outer-join

Sa se listeze pentru fiecare regiune numele insotitorilor de zbor si numelor aeroportului la care acestia lucreaza. Lista se va ordona alfabetic dupa numele regiunilor.

```
SELECT r.num "Nume regiune", i.num "Nume insotitor", a.num "Nume aeroport"
FROM INSOTITORI i FULL OUTER JOIN AEROPORTURI a ON (cod_IATA = cod_aeroport)
FULL OUTER JOIN ORASE o ON (a.cod_postal = o.cod_postal)
FULL OUTER JOIN REGIUNI r ON (r.cod_regiune = o.cod_regiune)
ORDER BY r.num;
```