Bejan Topse Lucian-Cosmin [211] Mate Info

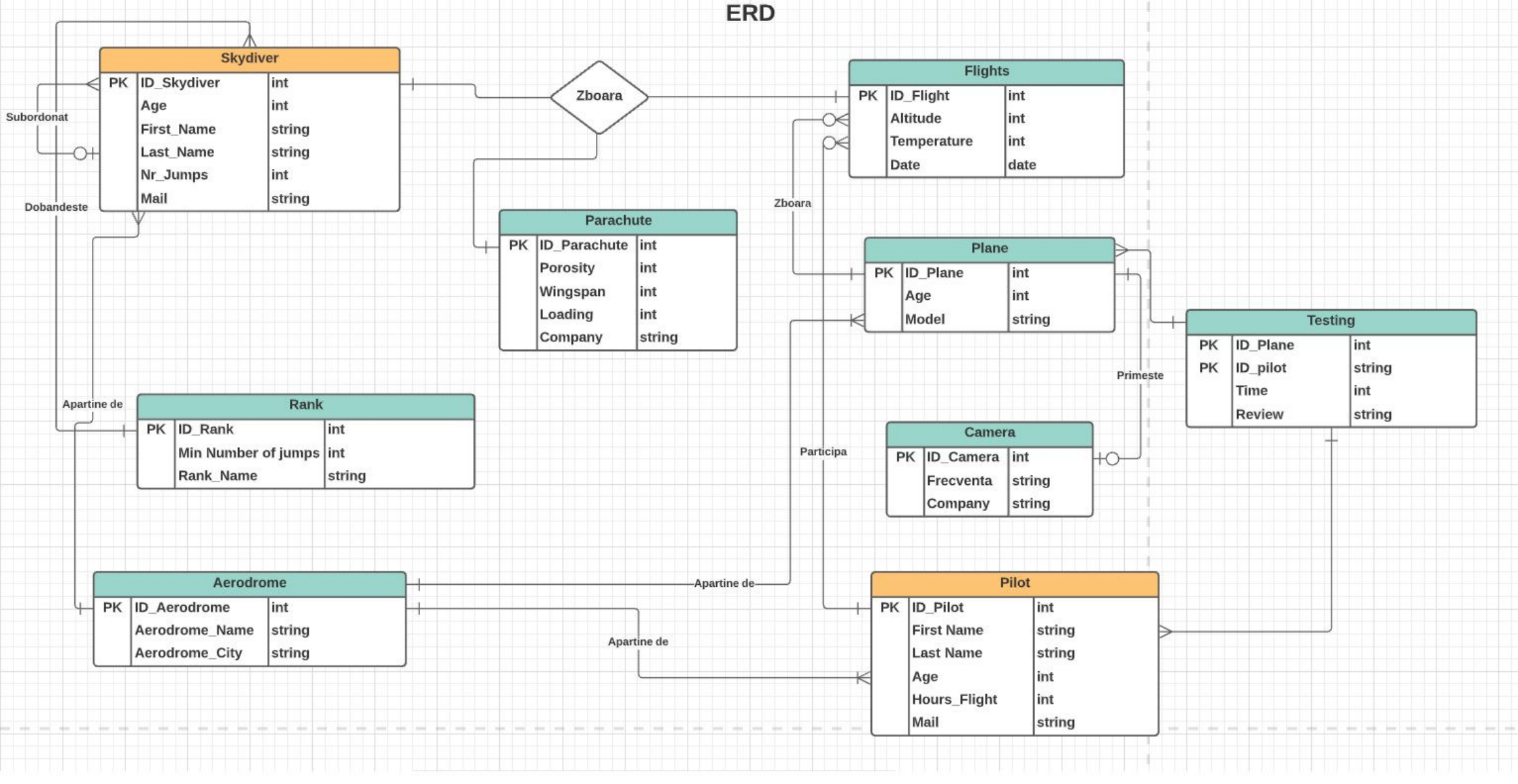
Eveniment Parasutisti

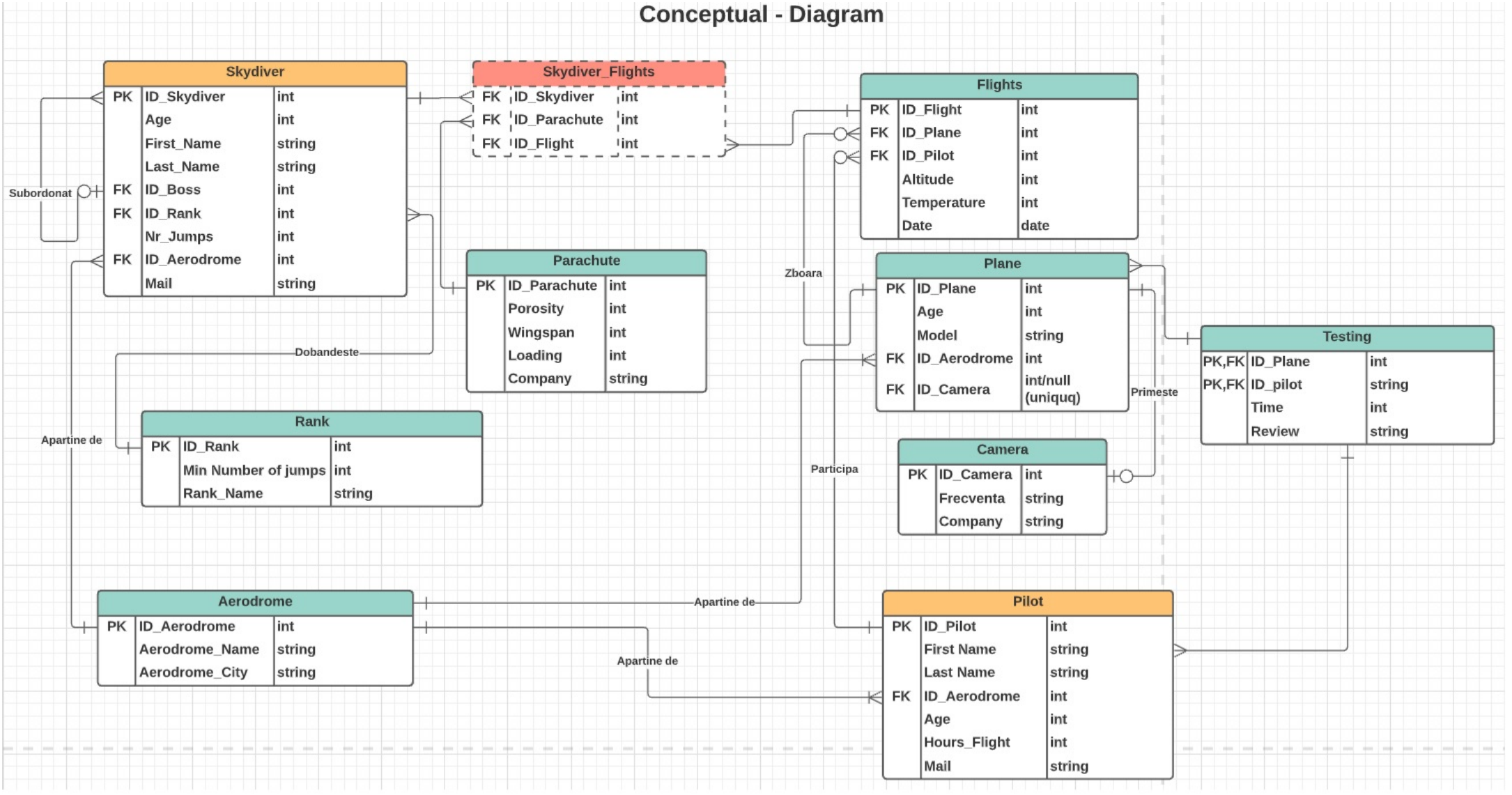
Intrunirea anuala de parasutisti are loc pe 4.11.2021 in Bucuresti. Aceasta presupune o adunare a tuturor parasutistilor, indifferent de rank in bucuresti, impreuna cu pilotii si avionele de la toate aerodromurile din Romania. Pentru fiecare Parasutist se stie Numele, Prenumele, Rank-ul, cine este superiorul lui, numarul de salturi, aerodromul de unde vine, adresa de mail si varsta. Fiecare rank are un nume, si pentru a putea sa il dobandesti iti trebuie un numar minim de salturi. Fiecare Aerodrom are un nume si un oras de resedinta. In fiecare aerodrom se afla un anumit numar de avioane, pentru care se stie varsta si modelul, si un anumit numar de piloti, pentru care se stie numele, prenumele, varsta, adresa de mail si totalul orelor de zbor. Parasutistii au la dispozitie un numar de parasute pentru care se cunoaste porozitate, anvergura, incarcarea si compania care a produs-o.

In decursul acestui eveniment se vor desfasura mai multe zboruri pentru care se va retine numarul, pilotul, avionul, altitudinea si parasutistii care au participat. In timpul acestor zboruri, unele avione primesc o camera proprie, pentru care se stie firma, care face poze la un anumit interval de timp. Se stie clar ca nu vor apuca toate avioanele si toti pilotii sa participe la aceste zboruri.

De asemenea, cu o zi inainte de inceperea evenimentului toti pilotii vor testa toate avioanele, printr-un mic zbor de 15-30 de minute dupa care vor lasa un review.

ERD


Skydiver
PK ID_Skydiver int
Age int
First_Name string
Last_Name string
Nr_Jumps int
Mail string
Subordonat
Dobandeste
Zboara
Flights
PK ID_Flight int
Altitude int
Temperature int
Date date
Zboara
Parachute
PK ID_Parachute int
Porosity int
Wingspan int
Loading int
Company string
Plane
PK ID_Plane int
Age int
Model string
Testing
PK ID_Plane int
PK ID_pilot string
Time int
Review string
Primeste
Apartine de
Rank
PK ID_Rank int
Min Number of jumps int
Rank_Name string
Participa
Camera
PK ID_Camera int
Frecventa string
Company string
Aerodrome
PK ID_Aerodrome int
Aerodrome_Name string
Aerodrome_City string
Apartine de
Apartine de
Pilot
PK ID_Pilot int
First Name string
Last Name string
Age int
Hours_Flight int
Mail string


Conceptual - Diagram
Skydiver
PK ID_Skydiver int
Age int
First_Name string
Last_Name string
FK ID_Boss int
FK ID_Rank int
Nr_Jumps int
FK ID_Aerodrome int
Mail string
Skydiver_Flights
FK ID_Skydiver int
FK ID_Parachute int
FK ID_Flight int
Flights
PK ID_Flight int
FK ID_Plane int
FK ID_Pilot int
Altitude int
Temperature int
Date date
Parachute
PK ID_Parachute int
Porosity int
Wingspan int
Loading int
Company string
Plane
PK ID_Plane int
Age int
Model string
FK ID_Aerodrome int
FK ID_Camera int/null (uniquq)
Testing
PK,FK ID_Plane int
PK,FK ID_pilot string
Time int
Review string
Rank
PK ID_Rank int
Min Number of jumps int
Rank_Name string
Camera
PK ID_Camera int
Frecventa string
Company string
Aerodrome
PK ID_Aerodrome int
Aerodrome_Name string
Aerodrome_City string
Pilot
PK ID_Pilot int
First Name string
Last Name string
FK ID_Aerodrome int
Age int
Hours_Flight int
Mail string
Subordonat
Dobandeste
Zboara
Primeste
Apartine de
Participa
Apartine de
Apartine de

8. Scheme Relationale :

- Skydiver (ID_Skydiver, Age, First_Name, Last_Name, Nr_Jumps, ID_Boss)
- Aerodrome (ID_Aerodrome, Name, City)
- Pilot (ID_Pilot, First_Name, Last_Name, Age)
- Plane (ID_Plane, Age, Model)
- Camera (ID_Camera, Color, Company)
- Rank (ID_Rank, Min_Jumps, Rank_Name)
- Flights (ID_Flight, Altitude, Temperature, Date)
- Parachute (ID_Parachute, Porosity, Wingspan, Loading, Company)

Bejan Topse Lucian-Cosmin [211] Mate Info

Normalizari

Non-FN1

O relatie este in FN1 daca domeniile pe care sunt definite atributele relatiei sunt constituite numai din valori atomice.

Pilot

ID_Pilot	First_Name	Last_name	ID_Aerodrome	Age	Hours_Flight	Mail
607104	Andrei	Mocanu	10567	45	20 000	a_mocanu@gmail.com
607106	Marian	Georgescu	10567	44	18 000	m_geor@gmail.com marian_g@yahoo.com
607107	Matei	Andronic	10568	50	26 000	mateiandr5@gmail.com

FN1

Pilot

ID_Pilot	First_Name	Last_name	ID_Aerodrome	Age	Hours_Flight	Mail
607104	Andrei	Mocanu	10567	45	20 000	a_mocanu@gmail.com
607106	Marian	Georgescu	10567	44	18 000	m_geor@gmail.com
607106	Marian	Georgescu	10567	44	18 000	marian_g@yahoo.com
607107	Matei	Andronic	10568	50	26 000	mateiandr5@gmail.com

Plane
ID_Plane
Age
Model
ID_Camera
Color(camera)
Company (camera)
ID_Aerodrome
Aerodrome_name
Aerodrome_city
ID_Pilot
First_Name(pilot)
Last_Name(pilot)
Age(pilot)
Hours_flight(pilot)
Mail(pilot)
Time(test)
Review(test)

Non-FN1

Aducerea relatiilot in FN1 presupune eliminarea atributelor compuse si a celor repetitive. In exemplul urmator se vor elimina grupurile repetitive prin construirea de noi relatii.

Etape:

1. Se construieste cate o relatie pt fiecare grup repetitive
2. In schema fiecarei noi relatii obtinute la pasul 1 se introduce si cheia primara a relatiei Plane nenormalizata
3. Cheia primara a fiecarei noi relatii va fi compusa din atributul chei ale relatiei Plane, plus unul sau mai multe attribute propria.

FN1

Plane
ID_Plane
Age (plane)
Model (plane)
ID_Camera
Color (camera)
Company (camera)
ID_Aerodrome
Aerodrome_name
Aerodrome_city

Non-FN2

Testing
ID_Plane
ID_Pilot
Time(test)
Review(test)
First_Name(pilot)
Last_Name(pilot)
Age(pilot)
Hours_flight(pilot)
Mail(pilot)

Bejan Topse Lucian-Cosmin [211] Mate Info

FN2 Non-FN3

Plane
ID_Plane
Age (plane)
Model (plane)
ID_Camera
Color (camera)
Company (camera)
ID_Aerodrome
Aerodrome_name
Aerodrome_city

Pilot
ID_Pilot
First_Name
Last_Name
Age
Hours_flight
Mail

Testing
ID_Plane
ID_Pilot
Time
Review

O relatie se afla in FN2 daca:

- Se afla in FN1
- Fiecare atribut care nu este cheie este dependet de intreaga cheie primara

Etape FN1 -> FN2:

1. Se identificare posibila cheie primara
2. Se identifica toate dependentele dintre atributele relatiei cu exceptia acelora in care sursa este un atribut component al cheii primare.
3. Se identifica toate dependentele care au ca sursa un atribut sau subansamblu de attribute din cheia primara
4. Pentru fiecare atribut (sau subansamblu) al cheii de la pasul III se creează o relaţie care va avea cheia primară atributul (subansamblul) respectiv, iar celelalte atribute vor fi cele care apar ca destinaţie în dependenţele de la etapa III.

In cazul nostru, se observa ca First_Name, Last_Name, Age, Hours_flight si Mail depinde doar de ID_Pilot, nu si de ID_plane.

Bejan Topse Lucian-Cosmin [211] Mate Info

FN3

Pilot
ID_Pilot
First_Name
Last_Name
Age
Hours_flight
Mail

Plane
ID_Plane
Age
Model

Camera
ID_Camera
Color
Company

Aerodrome
ID_Aerodrome
Aerodrome_name
Aerodrome_city

Testing
ID_Plane
ID_Pilot
Time
Review

O relatie se afla in FN3 daca:

- Se afla in FN2
- Fiecare atribut care nu este cheie depinde direct de cheia primara

Etape FN2 -> FN3:

1. Se identifica toate atributele ce nu fac parte din cheia primara si sunt surse ale unor dependente funcitonale
2. Pentru aceste atribute, se construieşte cate o relatie în care cheia primara va fi atributul respectiv, iar celelalte atribute, destinaţiile din DF considerate;
3. Din relatia de la care s-a pornit se elimina atributele destinatie din DF identificata la pasul I, si se pastreaza atributele surse

In cazul nostru, in relatia Plane, se observa ca ID_Camera si ID_Aerodrome sunt surse ale unor dependente functionale, si nu fac parte din cheia primara.

Se va crea o relatie in care cheia primara este ID_Camera, iar celalalte attribute destinatiile din dependeta functionala, adica: Color si Company.

Exact la fel si pentru ID_Aerodrome cu dependetele Aerodrome_Name si Aerodrome_City.

Bejan Topse Lucian-Cosmin [211] Mate Info

Non-BCNF

BCNF

Pentru a exemplifica voi folosi un model fictiv. In aceast idee, in cadrul evenimentului, se vor executa si niste salturi in care fiecare parasustist vine cu parasuta lui persoanala unica.

Skydiver_flight
ID_flight
Skydiver_id
Parachute_id
review

Skydiver_flight
ID_flight
Skydiver_ID
review

Parachute_skydiver
Parachute_ID
Skydiver_ID

K1 = ID_flight

K2 = Skydiver_ID

X = Parachute _ID

Y = Review

[K1]-> X,Y

[X] - > K2

Dependente:

(K1,K2) - > X

X->K2

Bejan Topse Lucian-Cosmin [211] Mate Info

Non-FN4

Pilot		
ID_Pilot	Mail	Nr_masina
607105	andrei@gmail.com	BC 05 EHI
607105	andrei_marian@yahoo.com	BC 05 EHI
607105	andrei_marian@yahoo.com	BC 17 MRT
607105	andrei@gmail.com	BC 17 MRT
607106	matei@gmail.com	MM 87 ERP
607107	marius@yahoo.com	GL 65 TYU

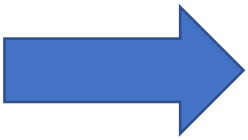
Pentru a exemplifica vom folosi un exemplu fictive. Presupunem ca in tabelul Pilot, pe langa campurile obisnuie avem si un camp Nr_Masina_Personala. Putem declara un tabel exemplu cu campurile ID Pilot, Nr_masina, si mail .

FN4

Pilot_masina	
ID_Pilot	Nr_Masina
607105	BC 05 EHI
607105	BC 17 MRT
607106	MM 87 ERP
607107	GL 65 TYU

Pilot_Mail	
ID_Pilot	Mail
607105	andrei@gmail.com
607105	andrei_marian@yahoo.com
607106	matei@gmail.com
607107	marius@yahoo.com

Pilot
ID_Pilot
Mail
Nr_Masina



Pilot_mail
ID_Pilot
Mail

Pilot_mail
ID_Pilot
Nr_masina

Bejan Topse Lucian-Cosmin [211] Mate Info

Non-FN5

Skydiver_Plane_parachute		
ID_Skydiver	ID_Plane	ID_Parachute
101	11	1001
101	12	1001
101	11	1002
102	12	1002
103	11	1003
104	11	1004
104	12	1003

S-P-P
ID_skydiver
ID_plane
ID_parachute

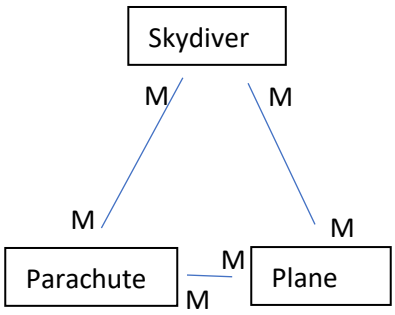
FN5

Se vor elimina relatiile N:M dependente. Astfel informatiile redundate se vor inlatura prin impartirea tabelului initial in 3 sau mai multe tabele.

Skydiver_plane	
ID_Skydiver	ID_Plane
101	11
101	12
102	12
103	11
104	11
104	12

Skydiver_parachute	
ID_Skydiver	ID_parachute
101	1001
101	1002
102	1002
103	1003
104	1003
104	1004

Parachute_plane	
ID_parachute	ID_plane
1001	11
1001	12
1002	11
1002	12
1003	11
1003	12
1004	11



Skydiver_plane
ID_skydiver
ID_plane

Skydiver_parachute
ID_skydiver
ID_parachute

Plane_parachute
ID_plane
ID_parachute

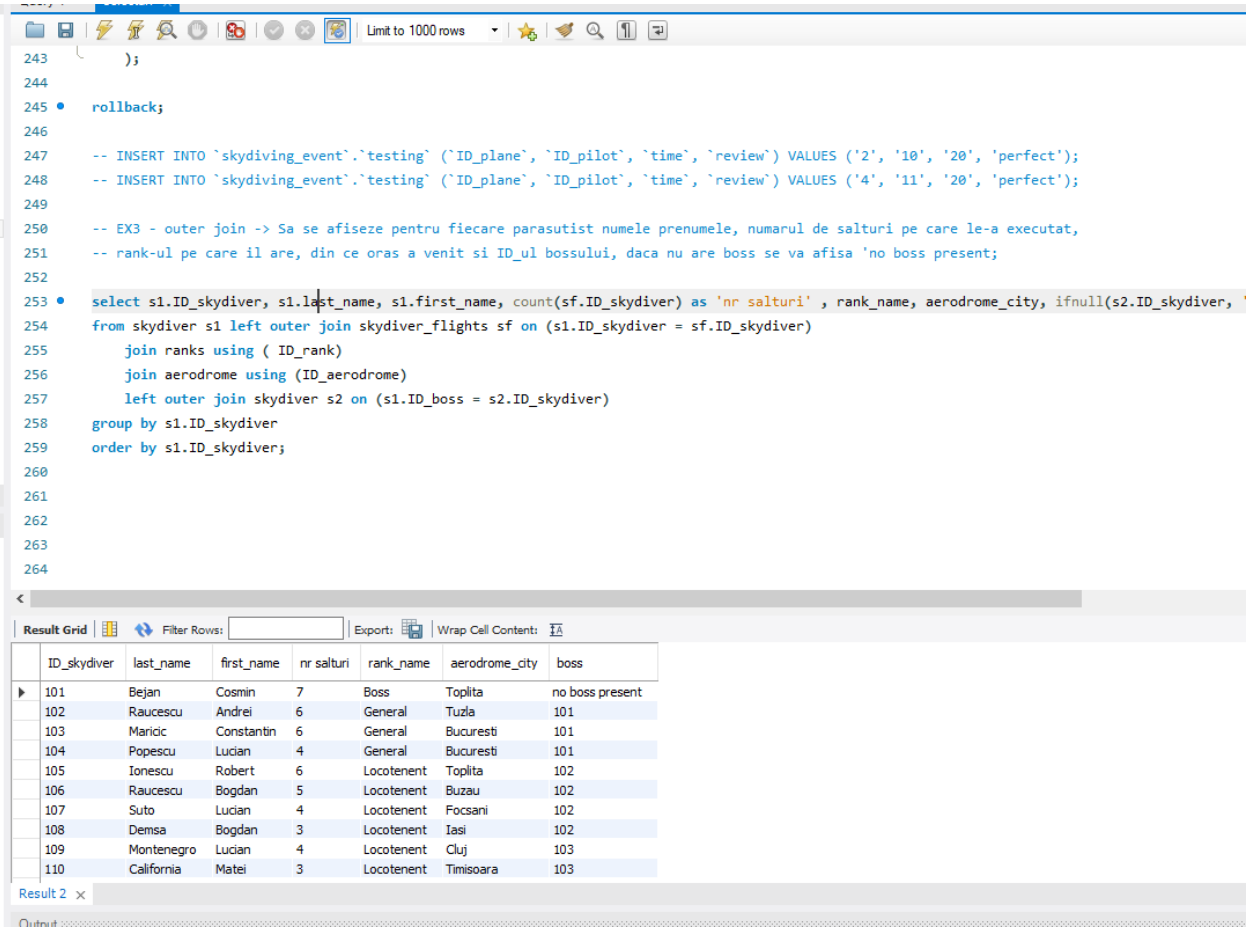
Bejan Topse Lucian-Cosmin [211] Mate Info

Denormalizari

In principia, denormalizarea este realizata cand se doreste cresterea performantei sau simplificarea programului de manipulare a datelor.

Pilot		
ID_Pilot	Mail	Nr_masina
607105	andrei@gmail.com	BC 05 EHI
607105	andrei_marian@yahoo.com	BC 05 EHI
607105	andrei_marian@yahoo.com	BC 17 MRT
607105	andrei@gmail.com	BC 17 MRT
607106	matei@gmail.com	MM 87 ERP
607107	marius@yahoo.com	GL 65 TYU

In acest exemplu, decat sa am 2 tabele separate cu [ID_pilot, Mail] si [ID_pilot, Nr_masina], daca stiu ca am nevoie constant de toate datele, decat sa fac join intre cele 2 tabele, ar fi mai efficient sa denormalizam si sa lucram intr-un singur tabel.



```
243     );
244
245 •  rollback;
246
247     -- INSERT INTO `skydiving_event`.`testing` (`ID_plane`, `ID_pilot`, `time`, `review`) VALUES ('2', '10', '20', 'perfect');
248     -- INSERT INTO `skydiving_event`.`testing` (`ID_plane`, `ID_pilot`, `time`, `review`) VALUES ('4', '11', '20', 'perfect');
249
250     -- EX3 - outer join -> Sa se afiseze pentru fiecare parasutist numele prenumele, numarul de salturi pe care le-a executat,
251     -- rank-ul pe care il are, din ce oras a venit si ID_ul bossului, daca nu are boss se va afisa 'no boss present;
252
253 •  select s1.ID_skydiver, s1.last_name, s1.first_name, count(sf.ID_skydiver) as 'nr salturi' , rank_name, aerodrome_city, ifnull(s2.ID_skydiver, '
254     from skydiver s1 left outer join skydiver_flights sf on (s1.ID_skydiver = sf.ID_skydiver)
255         join ranks using ( ID_rank)
256         join aerodrome using (ID_aerodrome)
257         left outer join skydiver s2 on (s1.ID_boss = s2.ID_skydiver)
258     group by s1.ID_skydiver
259     order by s1.ID_skydiver;
260
261
262
263
264
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ID_skydiver	last_name	first_name	nr salturi	rank_name	aerodrome_city	boss
101	Bejan	Cosmin	7	Boss	Toplita	no boss present
102	Raucescu	Andrei	6	General	Tuzla	101
103	Maricic	Constantin	6	General	Bucuresti	101
104	Popescu	Lucian	4	General	Bucuresti	101
105	Ionescu	Robert	6	Locotenent	Toplita	102
106	Raucescu	Bogdan	5	Locotenent	Buzau	102
107	Suto	Lucian	4	Locotenent	Focsani	102
108	Demsa	Bogdan	3	Locotenent	Iasi	102
109	Montenegro	Lucian	4	Locotenent	Cluj	103
110	California	Matei	3	Locotenent	Timisoara	103

Result 2

```sql
DROP TABLE IF EXISTS `aerodrome`;

CREATE TABLE `aerodrome` (
  `ID_aerodrome` int NOT NULL,
  `Aerodrome_name` varchar(45) NOT NULL,
  `Aerodrome_city` varchar(45) NOT NULL,
  PRIMARY KEY (`ID_aerodrome`),
  UNIQUE KEY `Aerodrome_name_UNIQUE` (`Aerodrome_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

LOCK TABLES `aerodrome` WRITE;

INSERT INTO `aerodrome` VALUES
  (1001,'SuperWings','Tuzla'),
  (1002,'CoolWings','Bucuresti'),
  (1003,'CoolerWings','Bucuresti'),
  (1004,'AirTurbo','Toplita'),
  (1005,'Eagles','Buzau'),
  (1006,'NorthPower','Focsani'),
  (1007,'PlaneMax','Iasi'),
  (1008,'Courage','Cluj'),
  (1009,'WesternClouds','Timisoara'),
  (1010,'BoysFromArad','Arad'),
  (1011,'BeachBreeze','Constanta'),
  (1012,'CloudOcean','Constanta'),
  (1013,'Aladin','Arad');

UNLOCK TABLES;
```

Output

Action Output

#	Time	Action	Message
3	14:37:53	select s1.ID_skydiver, s1.last_name, s1.first_name, count(sf.ID_skydiver) as 'nr salturi' , rank_name, aerodrome_city, ifnull(s2.ID_skydiver, 'no boss...	31 row(s) returned
4	14:38:47	use skydiving_shop	0 row(s) affected
5	14:39:15	DROP TABLE IF EXISTS `aerodrome`	0 row(s) affected, 1 warning(s): 1051 Unknown table 'skydiving_shop.aerodrome'
6	14:39:15	CREATE TABLE `aerodrome` (`ID_aerodrome` int NOT NULL, `Aerodrome_name` varchar(45) NOT NULL, `Aerodrome_city` varchar(45) NOT...	0 row(s) affected
7	14:39:15	LOCK TABLES `aerodrome` WRITE	0 row(s) affected
8	14:39:15	INSERT INTO `aerodrome` VALUES (1001,'SuperWings','Tuzla'), (1002,'CoolWings','Bucuresti'), (1003,'CoolerWings','Bucuresti'), (1004,'AirT...	13 row(s) affected Records: 13 Duplicates: 0 Warnings: 0
9	14:39:15	UNLOCK TABLES	0 row(s) affected

Am dat refresh, si a aparut tabelul in schema.