

PROIECT BAZE DE DATE

Airplane Database

1) Descrierea modelului real.

Un site de zboruri oferă bilete pentru zboruri de la mai multe companii (ex. RyanAir, TAROM, WizzAir, BlueAir etc.). Utilizatori își pot crea un cont alcătuit din email, parolă și date personale. Utilizatorul poate căuta zboruri cu locații și date fixe sau flexibile. Zborurile pot fi sortate și filtrate. Clientul poate alege mai multe opțiuni pentru bilet. Poate opta să aiba anulare gratuită pentru următoarele luni, să aiba prioritate la îmbarcare și își poate selecta un număr de bagaje de cală. Clientul poate vedea locurile disponibile și tipul locurilor (ex. regular, large etc) și poate alege pe care îl dorește.

2) Prezentarea constrângerilor.

- Pentru un email poate exista un singur cont de utilizator.
- Un utilizator poate cumpăra mai multe bilete.
- Un client poate avea un singur bagaj normal și până la maxim șase bagaje de cală.
- Zborurile cu escala sunt trecute pe același bilet.
- Un utilizator poate cumpăra bilete pentru același zbor pentru mai multe persoane distincte (au CNP diferit).
- Pe un rând din avion toate scaunele sunt de același tip.

3) Descrierea entităților.

ENTITATE	CHEIE PRIMARA	OBSERVAȚII
COMPANY	company_id	O companie poate avea mai multe adrese de email sau numere de telefon.
FLIGHT	flight_id	
AIRPLANE	airplane_id	
SEAT	airplane_id, row_id, column_id	Tipul scaunului este determinat exclusiv de avion și rând.
AIRPORT	airport_id	Un aeroport poate fi ori aeroport de sosire, ori aeroport de plecare pentru un zbor.
LOCATIONS	location_id	Locația reprezintă orașul căruia îi servește aeroportul, astfel într-o locație pot fi mai multe aeroporturi.
TICKET	ticket_id	Un bilet poate fi și Flexible și Luggage simultan sau poate să nu fie niciunul din ele.
USER	user_id	

4) Descrierea relațiilor.

RELATIE	CARDINALITATE	OBSERVAȚII
Operează	COMPANY – FLIGHT One-to-many	O companie poate opera mai multe zboruri. Un zbor poate fi operat de o singură companie.
Operează	FLIGHT – AIRPLANE Many-to-one	Un zbor operează un singur avion. Un avion poate fi operat pentru mai multe zboruri.
Deține/ Aparține	AIRPLANE – SEAT One-to-many	Un avion deține mai multe locuri. Un loc aparține de un singur avion.
Aterizează &decolează	FLIGHT – AIRPORT One-to-many	Un zbor poate ateriza și decola pe câte un aeroport. (unul la decolare, unul la aterizare) Pe un aeroport pot decola și ateriza mai multe zboruri.
Există	AIRPORT – LOCATIONS Many-to-one	Pentru un aeroport există o singură locație. Pentru o locație pot exista mai multe aeroporturi.
Are	FLIGHT – TICKET Many-to-many	Un zbor poate avea mai multe bilete. Un bilet poate avea mai multe zboruri.
Cumpără	TICKET – USER Many-to-one	Un bilet poate fi cumpărat de un singur utilizator. Un utilizator poate cumpăra mai multe bilete.

5) Descrierea atributelor.

COMPANY

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
name	string	20	Not null	Poate fi PK
phone_no	string	13	Not null	Poate avea mai multe numere de telefon
email	string	40	Not null	Poate avea mai multe email-uri.
date_start	date		Not null	Data la care s-a semnat contractul.
date_end	date		>date_start, Not null	Data la care expiră contractul.

AIRPORT

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
name	string	40	Not null	Poate fi PK

LOCATIONS

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
name	string	40	Not null	Poate fi PK
time_zone	integer	2	[-11, 14]	

AIRPLANE

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
model	string	20	Not null	
no_seats	integer	3	[50, 600]	

FLIGHT

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
co2	integer	4	[50,3000]	
airport_start_id	integer		Not null	Aeroportul de pe care decolează
airport_end_id	integer		Not null	Aeroportul pe care aterizează
date_start	datetime		Not null	Data și ora (locală) la care decolează.
date_end	datetime		>date_start, Not null	Data și ora (locală) la care aterizează.
price	integer	5		Prețul unui zbor (fără bagaje și anulare gratuită)
luggage_price	integer	3		Prețul unui bagaj de cală
month_cancel_price	integer	3		Prețul unei luni de anulare gratuită.

SEAT

Atribut	Tip	Dimensiune/ precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
s_type	string	20	Not null	
s_row	integer	2	[1, 99]	
s_column	integer	2	[1, 12]	
cnp	string	19		Opțional

TICKET

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
price	integer	5	[1,99999]	
cnp	string	19	Not null	Este vorba de cnp-ul clientului, care poate fi diferit de cel al user-ului
name	string	20	Not null	
surname	string	20	Not null	
birthday	date		Not null	
nationality	string	20	Not null	
disabled	string	3	Yes/no	
priority	string	3	Yes/no	
basic_luggage	string	3	Yes/no	

FLEXIBLE TICKET

Atribut	Tip	Dimensiune/ precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
no_months_cancel	integer	2	[1,24]	Numarul de luni în care își poate anula zborul de la data cumpărării biletului.

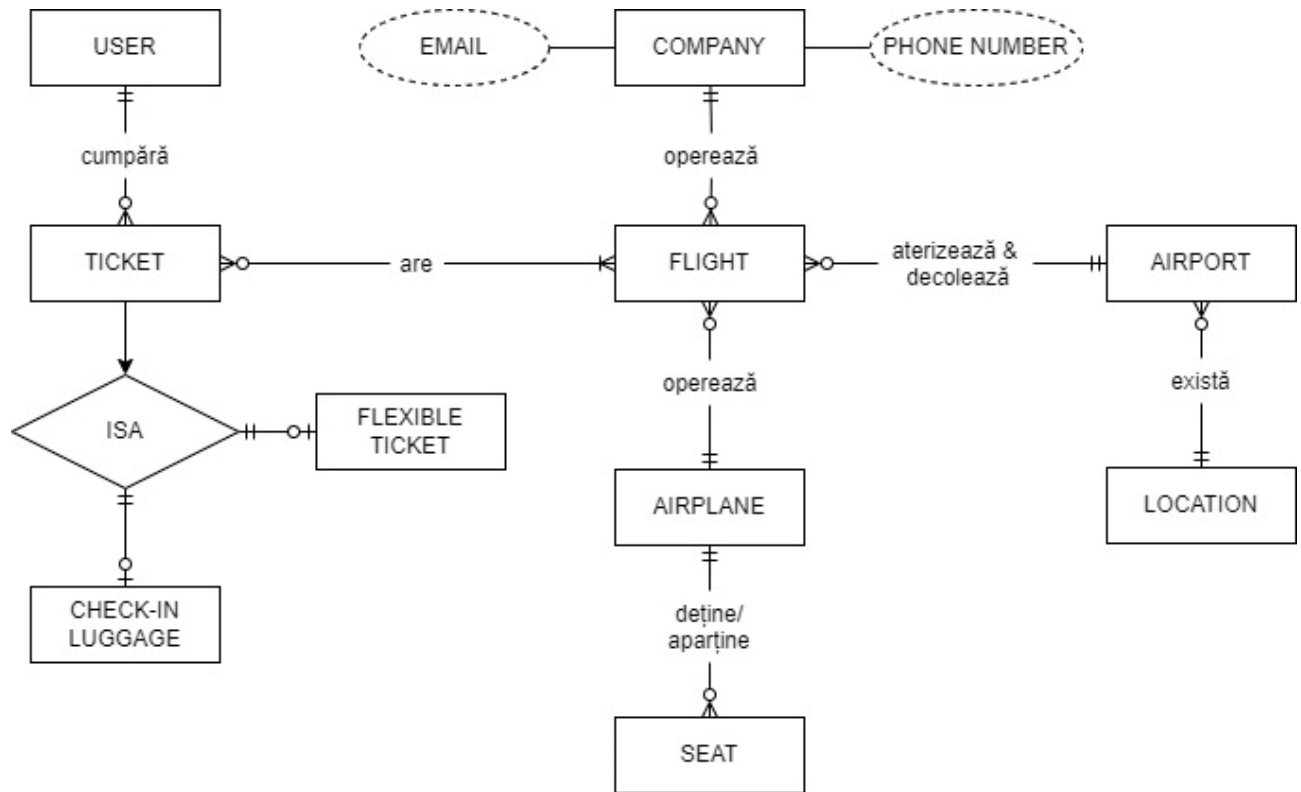
CHECK-IN LUGGAGE

Atribut	Tip	Dimensiune/ precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
no_luggages	integer	1	[1,6]	

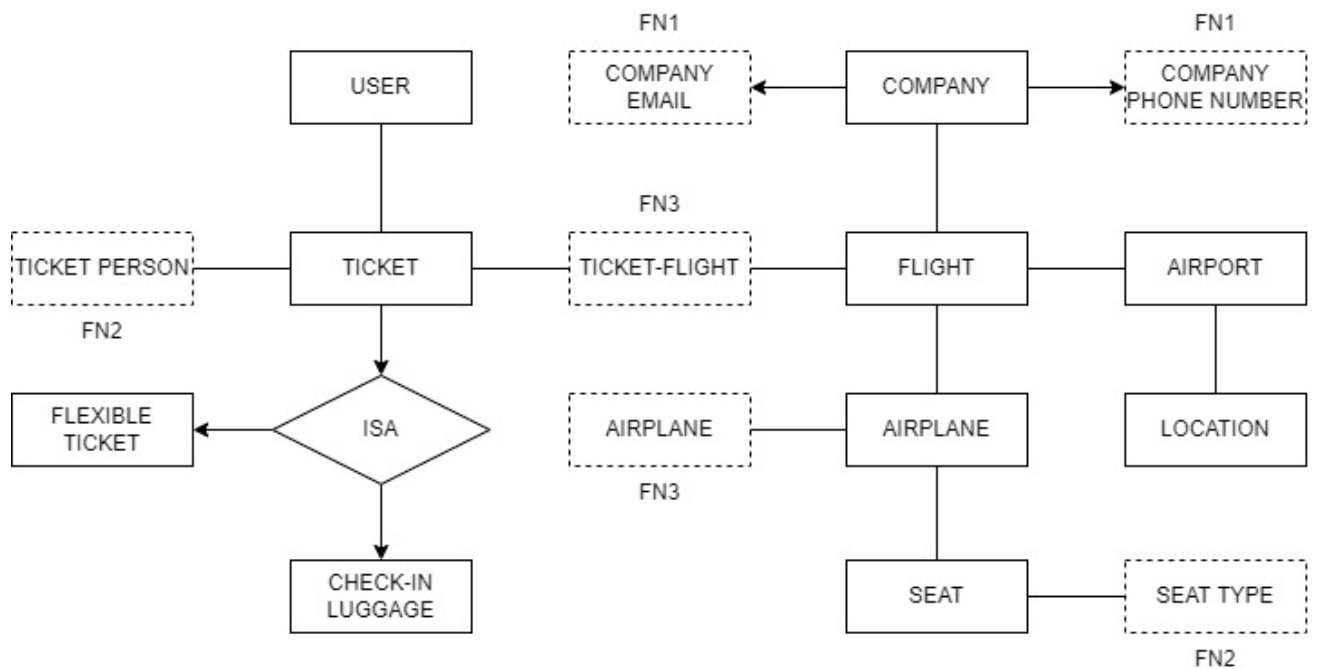
USER SITE

Atribut	Tip	Dimensiune / precizie	Valori posibile si valori default	Observații, obligatoriu/opțional
email	string	20	Trebuie să conțină @	Obligatoriu
pssword	string	20	Minim 6 caractere	Obligatoriu
name	string	20	Not null	
surname	string	20	Not null	
birthday	date		Un utilizator trebuie să aibă 18 ani.	
nationality	string	20	Not null	
date_signed	date			Reține data la care s-a creat contul

6) Diagramei entitate-relație.



7) Diagrama conceptuală în FN3.



8) Schemele relaționale.

COMPANY(company_id#, name, date_start, date_end)

COMPANY EMAIL(company_id#, email#)

COMPANY PHONE NUMBER(company_id#, phone_no#)

FLIGHT(flight_id#, airplane_id, company_id, airport_start_id, airport_end_id, co2, date_start, date_end, price, luggage_price, month_cancel_price)

AIRPORT(airport_id#, location_id, name)

LOCATION(location_id#, name, time_zone)

AIRPLANE(airplane_id#, model)

AIRPLANE MODEL(model#, no_seats)

SEAT(flight_id#, s_row#, s_column#, cnp)

SEAT TYPE(model#, st_row#, type)

TICKET(ticket_id#, cnp, user_id, price, priority, basic_luggage)

TICKET FLIGHT(ticket_id#, flight_id#)

TICKET PERSON(cnp#, name, surname, birthday, nationality, disabled)

FLEXIBLE TICKET(ticket_id#, no_months_cancel)

CHECK-IN LUGGAGE(ticket_id#, no_luggages)

USER SITE(user_id#, email, password, name, surname, birthday, nationality, date_signed)

9) Normalizare

I. Non-FN1:

- Enitatea COMPANY are două attribute multiple: email, phone_no. Acestea se transformă în două tabele în felul urmator.

Din tabelul initial:

COMPANY

COMPANY_ID	NAME	DATE_START	DATE_END	EMAIL	PHONE_NO
1	RyanAir	21.05.2021	21.05.2023	ryanair@gmail.com ryanair@yahoo.com	+40792 313 456 +35879 384 932
2	WizzAir	27.06.2021	27.06.2025	wizz@gmail.com	+74992 024 283 +38399 292 385

Obținem următoarele tabele:

COMPANY

COMPANY_ID	NAME	DATE_START	DATE_END
1	RyanAir	21.05.2021	21.05.2023
2	WizzAir	27.06.2021	27.06.2025

COMPANY EMAIL: (company_id, email) PK

COMPANY_ID	EMAIL
1	ryanair@gmail.com
1	ryanair@yahoo.com
2	wizz@gmail.com

COMPANY PHONE NUMBER: (company_id, phone_no) PK

COMPANY_ID	PHONE_NO
1	+40792 313 456
1	+35879 384 932
2	+74992 024 283
2	+38399 292 385

II. Non-FN2:

- Pentru entitatea TICKET avem:

TICKET(flight_id#, user_id, price, name, surname, cnp#, birthday, nationality, disabled, priority, basic_luggage)

Atunci:

$K1 = \text{FLIGHT_ID}$

$K2 = \text{CNP}$

$X = \text{USER_ID}, \text{PRICE}, \text{PRIORITY},$
 BASIC_LUGGAGE

$Y = \text{NAME}, \text{SURNAME}, \text{BIRTHDAY},$
 $\text{NATIONALITY}, \text{DISABLED},$

Obținem:

Relațiile de dependență:

$K2 \rightarrow Y$

$K1, K2 \rightarrow X$

TICKET(flight_id#, user_id, price, cnp#, priority, basic_luggage)

TICKET PERSON(cnp#, name, surname, birthday, nationality, disabled)

III. Non-FN3:

- Pentru entitatea AIRPLANE avem:

AIRPLANE(airplane_id#, model, no_seats)

Atunci:

$K = \text{AIRPLANE_ID}$

$X = \text{MODEL}$

$Y = \text{NO_SEATS}$

Obținem:

AIRPLANE(airplane_id#, model)

AIRPLANE MODEL(model#, no_seats)

Relațiile de dependență:

$K \rightarrow X$

$X \rightarrow Y$

10) Creare și inserare

```
CREATE TABLE flight(  
    flight_id NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),  
    airplane_id NUMBER NOT NULL,  
    company_id NUMBER NOT NULL,  
    airport_start_id NUMBER NOT NULL,  
    airport_end_id NUMBER NOT NULL,  
    co2 NUMBER(4) CONSTRAINT co2_value CHECK(co2 >= 50 AND co2 <= 3000),  
    date_start TIMESTAMP NOT NULL,  
    date_end TIMESTAMP NOT NULL,  
    price NUMBER(5),  
    luggage_price NUMBER(3),  
    month_cancel_price NUMBER(3),  
    CONSTRAINT check_date_flight CHECK(date_start < date_end)  
    CONSTRAINT flight_pk PRIMARY KEY(flight_id)  
);
```

```
CREATE TABLE user_site(  
    user_id NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),  
    email VARCHAR2(20) NOT NULL CONSTRAINT email_valid CHECK(email LIKE '%@%'),  
    password VARCHAR2(20) NOT NULL CONSTRAINT password_length CHECK(LENGTH(password) >= 6),  
    name VARCHAR2(20) NOT NULL,  
    surname VARCHAR2(20) NOT NULL,  
    date_signed DATE DEFAULT SYSDATE,  
    birthday DATE NOT NULL,  
    nationality VARCHAR2(20) NOT NULL,  
    CONSTRAINT user_pk PRIMARY KEY(user_id)  
);
```



```
ALTER TABLE user_site
ADD CONSTRAINT birthday_valid CHECK(birthday>=TO_DATE('1900/01/01', 'yyyy/mm/dd') AND TRUNC((date_signed-birthday)/365)>=18);
```

```
ALTER TABLE flight
ADD(
  CONSTRAINT flight_ariport_start_fk FOREIGN KEY(airport_start_id) REFERENCES airport(airport_id),
  CONSTRAINT flight_airport_end_fk FOREIGN KEY(airport_end_id) REFERENCES airport(airport_id),
  CONSTRAINT flight_unique_start UNIQUE(airplane_id, airport_start_id, date_start)
);
```

```
INSERT INTO airplane (model) VALUES ('Airbus A321neo');
INSERT INTO airplane (model) VALUES ('Boeing 737');
INSERT INTO airplane (model) VALUES ('Boeing 737');
INSERT INTO airplane (model) VALUES ('Airbus A350');
INSERT INTO airplane (model) VALUES ('Boeing 737');
INSERT INTO airplane (model) VALUES ('Boeing 777');
INSERT INTO airplane (model) VALUES ('Airbus A319');
INSERT INTO airplane (model) VALUES ('Boeing 737');
INSERT INTO airplane (model) VALUES ('Boeing 777');
```

11) Select-uri

- I. Să se afișeze compania care operează zborul, tipul avionului, indicii locurilor și numele complet al clienților care au plătit mai mult de 100 euro pentru bilet. Dacă există clienți care au cumpărat bilete mai scumpe de 100 euro, dar nu au selectat locul, se va afișa acest lucru.

```
SELECT c.name "COMPANY NAME", d.rand, d.coloana, d.price, d.full_name "FULL NAME"
FROM company c JOIN
(
  SELECT f.company_id company,
  CASE
    WHEN s.cnp IS NULL THEN 'clientul nu a selctat un loc'
    ELSE CAST(s.s_row AS VARCHAR2(2))
  END AS rand,
  CASE
    WHEN s.cnp IS NULL THEN 'clientul nu a selctat un loc'
    ELSE CAST(s.s_column AS VARCHAR(2))
  END AS coloana,
  t.price price, CONCAT(CONCAT(tp.name, ' '), tp.surname) full_name
  FROM ticket t JOIN ticket_flight tf ON (t.ticket_id = tf.ticket_id)
  JOIN flight f ON (tf.flight_id = f.flight_id)
  JOIN seat s ON (s.flight_id = f.flight_id)
  RIGHT JOIN ticket_person tp ON (s.cnp = tp.cnp)
  WHERE t.price > 100 AND t.cnp = tp.cnp
  ORDER BY tp.name) d ON(d.company = c.company_id);
```

COMPANY NAME	RAND	COLOANA	PRICE	FULL NAME
Blue Air	2	2	210	Popescu Alexandra
Blue Air	2	1	180	Popescu Andrei

- II. Să se afișeze numele și prenumele clienților care au cumpărat bilete pentru zborurile cu cele mai multe escale.

```
WITH escala_temp (max_escale) AS
  (SELECT max(escale)
   FROM (
     SELECT t.ticket_id ticket, t.cnp, COUNT(tf.flight_id) AS escale
     FROM ticket_person tp JOIN ticket t ON (tp.cnp = t.cnp)
                        JOIN ticket_flight tf ON (t.ticket_id = tf.ticket_id)
     GROUP BY t.ticket_id, t.cnp
     ORDER BY t.ticket_id) f
  )
SELECT CONCAT(CONCAT(f.name, ' '), f.surname) Name, f.ticket
FROM (
  SELECT tp.name, tp.surname, t.ticket_id ticket, t.cnp, COUNT(tf.flight_id) AS escale
  FROM ticket_person tp JOIN ticket t ON (tp.cnp = t.cnp)
                        JOIN ticket_flight tf ON (t.ticket_id = tf.ticket_id)
  GROUP BY tp.name, tp.surname, t.ticket_id, t.cnp
  ORDER BY t.ticket_id) f
JOIN escala_temp t ON (f.escale = t.max_escale);
```

NAME	TICKET
Popescu Andrei	1
Popescu Alexandra	2
Raducea Andreea	4
Elodi Lorelin	5
Dinu Valeria	6

III. Să se afișeze câți utilizatori au parola formată din cel puțin opt caractere.

```
SELECT COUNT(user_id) "NO USERS", LENGTH(password)
FROM user_site
HAVING LENGTH(password)>8
GROUP BY LENGTH(password);
```

NO USERS	LENGTH(PASSWORD)
2	10
3	9

IV. Să se afișeze locurile de tip large din avioanele care au decolat acum o săptămâna. Pentru acestea să se menționeze dacă au fost ocupate. Dacă da, să se afișeze cnp-ul persoanei respective.

```
SELECT NVL(s.cnp, 'Nu a fost ocupat'), s.s_row, s.s_column, f.flight_id
FROM flight f
      JOIN seat s ON (s.flight_id = f.flight_id)
      JOIN airplane a ON (a.airplane_id = f.airplane_id)
      JOIN seat_type st ON (st.model = a.model and s.s_row = st.st_row)
      LEFT JOIN ticket_person tp ON(s.cnp = tp.cnp),
      (SELECT SYSTIMESTAMP current_date
       FROM DUAL) d
WHERE EXTRACT(day FROM f.date_start-d.current_date)=-7 and st.st_type='large';
```

NVL(S.CNP,'NUAFOSTOCUPAT')	S_ROW	S_COLUMN	FLIGHT_ID
Nu a fost ocupat	1	1	1
Nu a fost ocupat	1	2	1
Nu a fost ocupat	1	3	1
Nu a fost ocupat	1	4	1
Nu a fost ocupat	1	5	1
Nu a fost ocupat	1	6	1

- V. Pentru zborurile care decolează de pe data de 06-03-2022 afișați numele și tipul companiei.

```
SELECT a.name, DECODE(c.name, 'Quatar Airways', 'Premium',  
                              'American', 'Bussines',  
                              'Avianca', 'Bussines',  
                              'Singapore Airlines', 'Premium',  
                              'Economy') "Company type"  
FROM flight f JOIN airport a ON (f.airport_start_id = a.airport_id)  
              JOIN company c ON (c.company_id = f.company_id)  
WHERE CAST(f.date_start AS DATE) LIKE TO_DATE('06/03/2022', 'MM/DD/YYYY');
```

NAME	Company type
Henri Coandă	Economy

12) Actualizare folosind subcereri

- Actualizați locurile de pe rândul unu să fie locuri 'large' pt avioanele de tip Airbus

```
UPDATE seat_type st  
SET st.st_type = 'large'  
WHERE st.st_row = 1 AND  
st.model IN (SELECT a.model  
             FROM airplane a  
             WHERE a.model LIKE 'Airbus%');
```

- Pentru biletele care aparțin de useri mai vechi de doi ani, modificați (daca este cazul) tipul biletului la priority.

```
UPDATE ticket t  
SET t.priority = 'yes'  
WHERE t.priority = 'no'  
AND t.user_id IN (SELECT u.user_id  
                  FROM user_site u  
                  WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, u.date_signed)/12) >= 2);
```

- Alocării clientului cu cnp-ul cel mai mic primul loc neocupat în avionul cu id-ul unu. Locul trebuie sa fie de tip 'normal'.

```

UPDATE seat s
SET s.cnp = (SELECT MIN(t.cnp)
            FROM ticket t
            JOIN ticket_flight tf ON (t.ticket_id = tf.ticket_id)
            JOIN seat ss ON (ss.flight_id = tf.flight_id)
            WHERE ss.flight_id = s.flight_id)
WHERE s.s_row = (SELECT MIN(ss.s_row)
                FROM seat ss
                JOIN flight ff ON(ss.flight_id = ff.flight_id)
                JOIN airplane aa ON(ff.airplane_id = aa.airplane_id)
                JOIN airplane_model am ON(aa.model = am.model)
                JOIN seat_type st ON(am.model = st.model AND st.st_row = ss.s_row)
                WHERE st.st_type = 'normal' AND ss.cnp IS NULL)
AND s.s_column = (SELECT MIN(ss.s_column)
                  FROM seat ss
                  WHERE ss.cnp IS NULL
                  AND ss.s_row = (SELECT MIN(ss.s_row)
                                  FROM seat ss
                                  JOIN flight ff ON(ss.flight_id = ff.flight_id)
                                  JOIN airplane aa ON(ff.airplane_id = aa.airplane_id)
                                  JOIN airplane_model am ON(aa.model = am.model)
                                  JOIN seat_type st ON(am.model = st.model AND st.st_row = ss.s_row)
                                  WHERE st.st_type = 'normal' AND ss.cnp IS NULL))
AND s.flight_id = 1;

```

13) Crearea unei secvențe utilizată la inserări

```

CREATE SEQUENCE boeing_seat_column_seq
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 6
NOCACHE
CYCLE;

CREATE SEQUENCE boeing_seat_row_seq
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 150
NOCYCLE;

ALTER SEQUENCE boeing_seat_column_seq RESTART START WITH 1;
ALTER SEQUENCE boeing_seat_row_seq RESTART START WITH 1;

INSERT INTO seat (flight_id, s_row, s_column)
SELECT 1, TRUNC((boeing_seat_row_seq.nextval-1)/6)+1, boeing_seat_column_seq.nextval
FROM DUAL
CONNECT BY LEVEL<=150;

```

14) Crearea unei vizualizări compuse.

```
CREATE OR REPLACE VIEW flight_airplane_view AS
SELECT f.flight_id, a.airplane_id, f.company_id, f.airport_start_id, f.airport_end_id, f.co2, f.date_start, f.date_end
FROM airplane a JOIN flight f ON(a.airplane_id = f.airplane_id);
```

- Operație LMD permisă

```
REM Operatie permisa
UPDATE flight_airplane_view fa
SET fa.company_id = 3
WHERE fa.flight_id = 1;
```

- Operație LMD nepermisă:

→se pot introduce date numai în tabelul care ofera o cheie candidat,

```
INSERT INTO flight_airplane_view (airplane_id, company_id, airport_start_id, airport_end_id, co2, date_start, date_end)
VALUES (3, 4, 10, 1, 171, TIMESTAMP '2023-06-03 19:45:00', TIMESTAMP '2023-06-03 21:15:00');
```

15) Non-BCNF

- Diagrama conceptuală se află deja în BCNF, însă să considerăm cazul ipotetic în care avem entitatea:

CLIENTI(client_id#, airplane_id#, pilot)

Reguli:

- Un client poate apărea la mai multe avioane (a cumpărat bilete pentru zboruri cu aeronave diferite)
- Un avion poate avea mai mulți piloti (de-a lungul contractului)
- Un pilot operează un unic avion (de-a lungul contractului)

Avem:

$K = \text{CLIENT_ID}, \text{AIRPLANE_ID}$

$K \rightarrow X$

$X = \text{PILOT}$




$X \rightarrow K_Y$

$K_Y = \text{AIRPLANE_ID}$

Obținem:

CLIENT PILOT(client_id#, pilot_id#)

PILOT AVION(pilot_id#, pilot, airplane_id)

165	0.00	INSERT INTO ticket_flight (ticket_id, flight_id) VALUES (6,	1 row(s) inserted.	1
166	0.02	INSERT INTO seat_type (model, st_row) SELECT 'Airbus A321ne	25 row(s) inserted.	25
167	0.02	DROP SEQUENCE boeing_seat_column_seq	Sequence dropped.	0
168	0.01	DROP SEQUENCE boeing_seat_row_seq	Sequence dropped.	0
169	0.02	CREATE SEQUENCE boeing_seat_column_seq START WITH 1 INCREMEN	Sequence created.	0
170	0.01	CREATE SEQUENCE boeing_seat_row_seq START WITH 1 INCREMENT B	Sequence created.	0
171	0.01	ALTER SEQUENCE boeing_seat_column_seq RESTART START WITH 1	Sequence altered.	0
172	0.00	ALTER SEQUENCE boeing_seat_row_seq RESTART START WITH 1	Sequence altered.	0
173	0.14	INSERT INTO seat (flight_id, s_row, s_column) SELECT 1, TRU	150 row(s) inserted.	150
174	0.07	UPDATE seat_type st SET st.st_type = 'large' WHERE st.st_row	1 row(s) updated.	1
175	0.05	UPDATE seat s SET s.cnp = (SELECT MIN(t.cnp) FR	1 row(s) updated.	1
176	0.03	UPDATE seat s SET s.cnp = '2432443243' WHERE s.s_row = (SELE	1 row(s) updated.	1
177	0.02	UPDATE seat s SET s.cnp = '2432234423' WHERE s.s_row = (SELE	1 row(s) updated.	1
178	0.03	UPDATE seat s SET s.cnp = '6464644443' WHERE s.s_row = (SELE	1 row(s) updated.	1
179	0.02	UPDATE seat s SET s.cnp = '8658453573' WHERE s.s_row = (SELE	1 row(s) updated.	1
180	0.03	UPDATE seat s SET s.cnp = '5757657355' WHERE s.s_row = (SELE	1 row(s) updated.	1
181	0.01	UPDATE user_site u SET u.date_signed = TO_DATE('2018/11/17',	2 row(s) updated.	2
182	0.01	UPDATE ticket t SET t.priority = 'yes' WHERE t.priority = 'n	1 row(s) updated.	1
183	0.03	CREATE OR REPLACE VIEW flight_airplane_view AS SELECT f.flig	View created.	0
184	0.03	UPDATE flight_airplane_view fa SET fa.company_id = 3 WHERE f	1 row(s) updated.	1
Download				
			row(s) 1 - 184 of 184	
<div><div><div>184</div><div>Statements Processed</div></div><div><div>184</div><div>Successful</div></div><div><div>0</div><div>With Errors</div></div></div>				
<div><div><div><div> mihail.chiribocsa@ts.unibuc.ro</div><div> airplane</div><div> en</div></div><div>Copyright © 1999, 2022, Oracle and/or its affiliates.</div><div>Oracle APEX 22.1.0</div></div></div>				