DOCUMENTATIE

-Managementul unui serviciu de remorcari-

Nume: Ghinet Rares-Andrei-George

Grupa: 251

Anul: 2

CTI

Cuprins

I

1. Prezentarea modelului si a regulilor acestuia……………………………3

2a. Diagrama entitate-relatie…………………………………………………5

2b. Descrierea entitatilor, atributelor, cheilor, relatiilor si a cardinalitatilor...5

3a. Diagrama conceptuala…………………………………………………..11

3b. Descrierea constrangerilor de integritate……………………………….11

3c. Schemele relationale (descrierea constrangerilor *on delete cascade*)…..12

II

1a. Crearea tabelelor......................................................................................13

1b. Introducerea datelor.................................................................................16

II

1) Prezentarea modelului si a regulilor acestuia

Serviciul de remorcari AutoBUC SRL, localizat in Bucuresti, doreste sa implementeze o baza de date in care sa se tina evidenta angajatilor care lucreaza in fiecare centru si a remorcilor pe care le detine, precum si pe cea a masinilor remorcate, a proprietarilor acestora si a amenzilor pe care acestia trebuie sa le plateasca.

Regulile dupa care functioneaza modelul:

*Angajatul*:

-apartine unui singur centru si poate avea un singur job in cadrul acelui centru

-acesta poate sa apartina sau nu unei echipe care tracteaza masini

*Jobul*:

-un job poate fi detinut de zero sau mai multi angajati

*Centrul de remorcare*:

-poate avea zero sau mai multe remorci in cadrul sau

-poate avea unul sau mai multi angajati

-in cadrul sau pot exista zero sau mai multe masini remorcate

-din fiecare centru pot opera zero sau mai multe echipe care remorcheaza automobile

*Remorca*:

-poate apartine unui singur centru

*Echipa*:

-opereaza dintr-un singur centru

-are doi sau mai multi angajati (un sofer si un asistent tehnic de preferat)

-poate sa fi remorcat zero sau mai multe masini

-poate sa duca o masina si in alt centru decat in cel din care opereaza

*Masina remorcata*:

-are un singur proprietar

-este localizata intr-un singur centru

*Soferul*:

-detine o singura masina

-are de platit o singura amenda

*Amenda*:

-este asociata unui singur sofer

Observatii:

-daca exista un centru cu una sau mai multe echipe, dar fara remorci se presupune ca acestea se inchiriaza de la o firma specializata (acelasi lucru se intampla si daca exista mai multe echipe decat remorci)

-daca intr-un centru nu exista angajati cu un anumit job, se presupune ca sarcinile vor fi preluate de alti angajati (de preferat aceia care au competentele necesare)

-daca exista un centru cu un singur angajat, acela va fi administratorul

-in general, soferul este un angajat cu mai multa experienta decat asistentul si poate fi platit mai bine

2a) Diagrama entitate-relatie

O imagine care conține text, hartă

Descriere generată automat

2b) Descrierea entitatilor, atributelor, cheilor, relatiilor si a cardinalitatilor

*ANGAJATI*:

Atribute: id\_angajat(PK), nume, prenume, email, telefon, salariu, nume\_job(FK), id\_centru(FK), numar\_echipa(FK)

-id\_angajat: un numar unic prin care se identifica un salariat in baza de date, fiind astfel Primary Key pentru acest tabel

-nume: numele angajatului (NOT NULL)

-prenume: prenumele angajatului (NOT NULL)

-email: un angajat trebuie in mod obligatoriu sa isi faca un email, pentru a putea fi contactat (NOT NULL)

-telefon: un angajat trebuie sa ofere si un numar de telefon, pe langa email (NOT NULL)

-salariu: cat castiga angajatul pe luna, depinde de job si este cuprins intre salariul minim si cel maxim din JOBURI (NOT NULL)

-nume\_job: numele jobului pe care acesta il detine in prezent (FK - nu poate fi NULL)

-id\_centru: numarul centrului in care lucreaza angajatul (FK - nu poate fi NULL)

-numar\_echipa: daca angajatul participa la activitatea de remorcare a masinilor, se va scrie si numarul echipei din care face parte (FK - poate sa fie NULL)

Relatii:

ANGAJATI-JOBURI: M(0)-1(1)

-un angajat are un singur job la un anumit moment in timp, dar este posibil sa existe un job pentru care nu exista angajati in baza de date

ANGAJATI-CENTRE-REMORCARE: M(1)-1(1)

-un angajat lucreaza intr-un singur centru, dar centrul poate avea unul sau mai multi angajati. In centrul cu un singur angajat va lucra doar administratorul

ANGAJATI-ECHIPE: M(2)-1(1)

-angajatul poate sau nu sa lucreze intr-o singura echipa care tracteaza masini, iar echipa este alcatuita din doi sau mai multi oameni (acestia sunt de regula un sofer cu unul sau mai multi asistenti tehnici). Soferul este considerat de obicei ca fiind un angajat cu mai multa experienta

*JOBURI*:

Atribute: nume\_job(PK), salariu\_minim, salariu\_maxim

-nume\_job: un job se identifica prin numele sau (NOT NULL)

-salariu\_minim: suma minima incasata de un salariat pentru jobul pe care acesta il are (NOT NULL)

-salariu\_maxim: suma minima incasata de un salariat pentru jobul pe care acesta il are; salariul efectiv este cuprins intre aceste doua valori (NOT NULL)

Relatii:

JOBURI-ANGAJATI: 1(1)-M(0)

-fiecare angajat are un singur job, dar pot exista joburi care sa nu fie avute de vreun angajat (in situatia in care un nou job a fost introdus si inca se cauta angajati)

*CENTRE\_REMORCARE*:

Atribute: id\_centru(PK), adresa, sector, data\_deschidere

-id\_centru: un numar prin care se identifica fiecare centru, acesta fiind Primary Key

-adresa: daca o persoana doreste sa gaseasca un centru anume, este nevoie de adresa acestuia (NOT NULL)

-sector: daca intamplator adresele a doua centre sunt identice, persoana se poate uita la numarul sectorului (NOT NULL)

-data\_deschidere: este necesara in cazul in care se doreste analiza activitatii unui centru de la deschidere pana in prezent, pentru a identifica centrele care nu au avut rezultate satisfacatoare (NOT NULL)

Relatii:

CENTRE\_REMORCARE-ANGAJATI: 1(1)-M(1)

-un centru are cel putin un angajat (daca este unul singur acesta va fi administratorul), iar un angajat poate sa lucreze intr-un singur centru la un anumit moment in timp

CENTRE\_REMORCARE-REMORCI: 1(1)-M(0)

-intr-un centru pot fi localizate zero sau mai multe remorci (care apartin acestuia), iar remorca trebuie neaparat sa apartina unui centru

CENTRE\_REMORCARE-ECHIPE: 1(1)-M(0)

-dintr-un centru pot sa opereze mai multe echipe care se ocupa cu tractarea masinilor, minimul fiind zero. De asemenea, o echipa trebuie sa aiba neaparat un centru din care sa isi desfasoare activitatea

*Remorci*:

Atribute: nr\_remorca(PK), numar\_inmatriculare, marca, greutate\_maxima, id\_centru(FK)

-nr\_remorca: numarul unic prin care se identifica o remorca, cee ace inseamna ca va fi Primary Key

-numar\_inmatriculare: pentru a circula legal, dar si pentru a putea fi identificata, este nevoie de numarul de inmatriculare al remorcii (NOT NULL)

-marca: in cazul in care este nevoie de asistenta din exterior, trebuie sa fie cunoscut si fabricantul remorcii (NOT NULL)

-greutate\_maxima: remorca nu poate ridica masini mai grele decat greutatea permisa, dar poate fi modificata corespunzator (NOT NULL)

-id\_centru: centrul in care este localizata remorca (NOT NULL)

Relatii:

REMORCI-CENTRE\_REMORCARE: M(0)-1(1)

-o remorca trebuie neaparat sa apartina unui centru, dar un centru poate avea zero sau mai multe remorci

*ECHIPE*:

Atribute: numar\_echipa(PK), id\_centru(FK)

-numar\_echipa: este un unic prin care se identifica o echipa, deci va fi Primary Key

-id\_centru: centrul din care opereaza echipa respective

Relatii:

ECHIPE-ANGAJATI: 1(0)-M(2)

-este nevoie de doi sau mai multi angajati pentru a ridica o masina, dar un angajat care nu are competenta necesara nu poate sa plece pe teren (de exemplu un functionar)

ECHIPE-CENTRE\_REMORCARE: M(0)-1(1)

-dintr-un centru de remorcare pot opera zero sau mai multe echipe, iar o echipa trebuie neaparat sa apartina unui centru

ECHIPE-MASINI REMORCATE: 1(1)-M(0)

-este posbil ca o echipa sa fi ridicat zero sau mai multe masini, dar o masina a fost ridicata de o singura echipa si dusa intr-un centru

*MASINI\_REMORCATE*:

Atribute: id\_masina(PK), numar\_masina, marca\_masina, culoare, id\_centru(FK), numar\_echipa(FK)

-id\_masina: numarul unic prin care identifica o masina remorcata in baza de date, deci va fi Primary Key

-numar\_masina: numarul de inmatriculare al masinii trebuie retinut pentru a identifica mai usor vehiculul (NOT NULL)

-marca\_masina: daca numerele de inmatriculare a doua masini coincid, marca masinii va fi urmatorul criteriu de identficare (NOT NULL)

-culoare: daca si marcile coincid, ne vom orienta dupa culoare (NOT NULL)

-id\_centru: centrul in care a fost dusa masina (NOT NULL)

-numar\_echipa: echipa care a dus masina in centrul respectiv (NOT NULL)

Relatii:

MASINI\_REMORCATE-CENTRE\_REMORCARE: M(0)-1(1)

-intr-un centru se pot afla zero sau mai multe masini, iar o masina remorcata trebuie neaparat sa fie localizata intr-un centru

MASINI\_REMORCATE-ECHIPE: M(0)-1(1)

-masina a fost ridicata de catre o singura echipa, dar o echipa poate avea zero sau mai multe masini ridicate

MASINI\_REMORCATE-SOFERI: 1(1)-1(1)

-o masina ridicata are un singur proprietar, iar un proprietar poate avea o singura masina ridicata in baza de date

*SOFERI*:

Atribute: cnp(PK), nume\_sofer, prenume\_sofer, telefon\_sofer, email\_sofer, id\_masina(FK), cod\_amenda(FK)

-cnp: un sofer poate fi identificat usor dupa cnp-ul sau, deci acesta poate fi trecut ca Primary Key

-nume\_sofer: este nevoie de numele de familie pentru ca acesta sa fie identificat (NOT NULL)

-prenume\_sofer: este necesar si prenumele acestuia (NOT NULL)

-telefon\_sofer: proprietarul masinii trebuie sa ofere numarul sau de telefon ca sa poata fi contactat la nevoie (NOT NULL)

-email\_sofer: daca doreste, soferul poate sa ofere o adresa de email ca metoda alternativa de a fi contactat (poate sa fie NULL)

-id\_masina: numarul de identificare al masinii asociate unui sofer (NOT NULL+UNIQUE)

-cod\_amenda: codul amenzii pe care soferul trebuie sa o plateasca (NOT NULL+UNIQUE)

Relatii:

SOFERI-MASINI\_REMORCATE: 1(1)-1(1)

-fiecare sofer este proprietarul unei singure masini, iar o masina apartine unui singur sofer

SOFERI-AMENZI: 1(1)-1(1)

-fiecare sofer are asociata o amenda, iar fiecare amenda trebuie sa fie platita de un singur sofer

*AMENZI*:

Atribute: cod\_amenda(PK), valoare\_amenda, observatii, data\_emitere

-cod\_amenda: un cod alcatuit din litere si cifre, prin care se identifica fiecare amenda, reprezentand Primary Key-ul tabelului

-valoare\_amenda: suma de bani pe care soferul trebuie sa o plateasca pentru a-si recupera vehiculul (NOT NULL)

-observatii: informatii aditionale referitoare la contextul in care masina a ajuns sa fie ridicata (poate sa fie NULL)

-data\_emitere: data la care a fost emisa amenda si la care a fost ridicata masina (NOT NULL)

3a) Diagrama conceptuala

O imagine care conține cer, masă, alb, text

Descriere generată automat

3b) Descrierea constrangerilor de integritate

*ANGAJATI*:

-nume, prenume, email, telefon, salariu, nume\_job si id\_centru au la constrangere de integritate NOT NULL pentru ca sunt obligatorii pentru fiecare angajat (nume\_job, id\_centru si numar\_echipa sunt FOREIGN KEY)

*JOBURI*:

-salariu\_minim si salariu\_maxim trebuie stiute pentru fiecare job in parte, deci vor fi NOT NULL

*CENTRE\_REMORCARE*:

-adresa, sector si data\_deschidere trebuie stiute pentru fiecare centru, deci vor fi NOT NULL

-sectorul va fi un numar de la 1 la 6, deci vom pune o conditie CHECK(sector>0 and sector<7)

*REMORCI*:

-numar\_inmatriculare, marca, greutate\_maxima si id\_centru vor fi NOT NULL pentru ca sunt obligatorii (id\_centru este FOREIGN KEY)

-greutatea maxima pe care o poate ridica o remorca este 4000kg, deci vom adauga o constrangere CHECK(greutate\_maxima<4000)

*ECHIPE*:

-id\_centru este un camp obligatoriu, deci va fi NOT NULL (este si FOREIGN KEY)

*MASINI\_REMORCATE*:

-numar\_masina, marca\_masina, culoare, id\_centru, numar\_echipa sunt obligatorii pentru identificarea vehiculului, deci vor fi NOT NULL (id\_centru si numar\_echipa vor fi FOREIGN KEY)

*SOFERI*:

-nume\_sofer, prenume\_sofer, telefon\_sofer, id\_masina, cod\_amenda sunt necesarea pentru identificarea proprietarului unei masini si pentru contactarea acestuia, deci vor fi NOT NULL (id\_masina si cod\_amenda sunt FOREIGN KEY)

-id\_masina si cod\_amenda vor fi si UNIQUE, pentru ca amenda asociata unui sofer are un cod unic, iar acesta este proprietarul unei singure masini

*AMENZI*:

-valoare\_amenda si data\_emitere trebuie stiute de catre sofer si de catre persoana responsabila pentru eliberarea masinii, deci vor fi NOT NULL

3c) Schemele relationale (descrierea constrangerilor *on delete cascade*)

*ANGAJATI*:

-se va aplica ON DELETE CASCADE pentru nume\_job si id\_centru: daca un centru este inchis sau jobul respectiv nu mai este figurat, angajatii vor fi

concediati

-se va aplica ON DELETE SET NULL pentru numar\_echipa: daca o echipa dispare, angajatii vor putea fi transferati in alte echipe

*REMORCI*:

-se va aplica ON DELETE CASCADE pentru id\_centru: daca este inchis centrul, vor disparea si remorcile

*ECHIPE*:

-se va aplica ON DELETE CASCADE pentru id\_centru: daca este inchis centrul, echipele vor fi si ele sterse

*MASINI\_REMORCATE*:

-se va aplica ON DELETE CASCADE pentru id\_centru: daca este inchis centrul, masinile vor fi sterse

-se va aplica ON DELETE SET NULL pentru numar\_echipa: daca o echipa este desfiintata, valoarea din campul corespunzator va deveni NULL

*SOFERI*:

-se va aplica ON DELETE SET NULL pentru id\_masina: daca o masina este dusa la casat sau nu mai poate fi recuperata, aceasta va fi stearsa, dar amenda si soferul vor ramane in baza de date pana la plata contraventiei

-se va aplica ON DELETE CASCADE pentru cod\_amenda: daca o amenda e platita, aceasta va fi stearsa din baza de date, odata cu proprietarul masinii. Automobilul ramane pana in momentul in care proprietarul intra in posesia lui

II

1a) Crearea tabelelor

O imagine care conține captură de ecran

Descriere generată automat

Baza de date a fost creata in Oracle Database o19c.

*ANGAJATI*:

create table ANGAJATI

(

id\_angajat NUMBER(4) PRIMARY KEY,

nume CHAR(10) NOT NULL,

prenume CHAR(15) NOT NULL,

email CHAR(40) NOT NULL,

telefon CHAR(10) NOT NULL,

salariu NUMBER(4) NOT NULL,

nume\_job CHAR(15) NOT NULL,

id\_centru NUMBER(3) NOT NULL,

numar\_echipa NUMBER(4),

FOREIGN KEY(nume\_job) REFERENCES JOBURI(nume\_job) ON DELETE CASCADE,

FOREIGN KEY(id\_centru) REFERENCES CENTRE\_REMORCARE(id\_centru) ON DELETE CASCADE,

FOREIGN KEY(numar\_echipa) REFERENCES ECHIPE(numar\_echipa) ON DELETE SET NULL

);

*JOBURI*:

create table JOBURI

(

nume\_job CHAR(15) PRIMARY KEY,

salariu\_minim NUMBER(6) NOT NULL,

salariu\_maxim NUMBER(6) NOT NULL

);

*CENTRE\_REMORCARE*:

create table CENTRE\_REMORCARE

(

id\_centru NUMBER(3) PRIMARY KEY,

adresa CHAR(60) NOT NULL,

sector NUMBER(1) NOT NULL,

data\_deschidere DATE NOT NULL

);

alter table CENTRE\_REMORCARE

add constraint valid\_sector check (sector>0 and sector<7);

*REMORCI*:

create table REMORCI

(

nr\_remorca NUMBER(4) PRIMARY KEY,

numar\_inmatriculare CHAR(19) NOT NULL,

marca CHAR(17) NOT NULL,

greutate\_maxima NUMBER(4) NOT NULL,

id\_centru NUMBER(3) NOT NULL,

FOREIGN KEY(id\_centru) REFERENCES CENTRE\_REMORCARE(id\_centru) ON DELETE CASCADE

);

alter table REMORCI

add constraint valid\_greutate\_maxima check (greutate\_maxima<4000);

*ECHIPE*:

create table ECHIPE

(

numar\_echipa NUMBER(4) PRIMARY KEY,

id\_centru NUMBER(3) NOT NULL,

FOREIGN KEY(id\_centru) REFERENCES CENTRE\_REMORCARE(id\_centru) ON DELETE CASCADE

);

*MASINI\_REMORCATE*:

create table MASINI\_REMORCATE

(

id\_masina NUMBER(4) PRIMARY KEY,

numar\_masina CHAR(19) NOT NULL,

marca\_masina CHAR(17) NOT NULL,

culoare CHAR(15) NOT NULL,

id\_centru NUMBER(3) NOT NULL,

numar\_echipa NUMBER(4) NOT NULL,

FOREIGN KEY(id\_centru) REFERENCES CENTRE\_REMORCARE(id\_centru) ON DELETE CASCADE,

FOREIGN KEY(numar\_echipa) REFERENCES ECHIPE(numar\_echipa) ON DELETE SET NULL

);

*SOFERI*:

create table SOFERI

(

cnp CHAR(13) PRIMARY KEY,

nume\_sofer CHAR(10) NOT NULL,

prenume\_sofer CHAR(15) NOT NULL,

telefon\_sofer CHAR(10) NOT NULL,

email\_sofer CHAR(40),

id\_masina NUMBER(4) NOT NULL,

cod\_amenda CHAR(4) NOT NULL,

FOREIGN KEY(id\_masina) REFERENCES MASINI\_REMORCATE(id\_masina) ON DELETE SET NULL,

FOREIGN KEY(cod\_amenda) REFERENCES AMENZI(cod\_amenda) ON DELETE CASCADE

);

alter table SOFERI

ADD UNIQUE(id\_masina);

alter table SOFERI

ADD UNIQUE(cod\_amenda);

*AMENZI*:

create table AMENZI

(

cod\_amenda CHAR(4) PRIMARY KEY,

valoare\_amenda NUMBER(5) NOT NULL,

observatii CHAR(40),

data\_emitere DATE NOT NULL

);

1b) Introducerea datelor

*ANGAJATI*:

INSERT INTO ANGAJATI VALUES

(

1,

'Berceanu',

'Constantin',

'berceanu.constantin@gmail.com',

'0742233500',

4500,

'Administrator',

1,

NULL

);

INSERT INTO ANGAJATI VALUES

(

2,

'Vladeanu',

'Cristina',

'vladeanu.cris@yahoo.com',

'0742247512',

2650,

'Functionar',

1,

NULL

);

*JOBURI*:

INSERT INTO JOBURI VALUES

( 'Mecanic',

1900,

2600

);

INSERT INTO JOBURI VALUES

(

'Functionar',

2400,

3000

);

*CENTRE\_REMORCARE*:

INSERT INTO CENTRE\_REMORCARE VALUES

( 1,

'Bulevardul Pipera Nr.22',

1,

TO\_DATE('23-04-2010','DD-MM-YYYY')

);

INSERT INTO CENTRE\_REMORCARE VALUES

( 2,

'Calea Giulesti Nr.43',

6,

TO\_DATE('21-05-2011','DD-MM-YYYY')

);

*REMORCI*:

INSERT INTO REMORCI VALUES

( 221,

'B-67-XYZ',

'BMW',

2000,

1

);

INSERT INTO REMORCI VALUES

( 446,

'B-99-MMT',

'Volkswagen',

2500,

1

);

*ECHIPE*:

INSERT INTO ECHIPE VALUES

( 45,

1

);

INSERT INTO ECHIPE VALUES

( 255,

4

);

*MASINI\_REMORCATE*:

INSERT INTO MASINI\_REMORCATE VALUES

(

14,

'B-77-UPS',

'Dacia',

'Rosu',

4,

255

);

INSERT INTO MASINI\_REMORCATE VALUES

(

12,

'B-43-REM',

'Renault',

'Alb',

12,

31

);

*SOFERI*:

INSERT INTO SOFERI VALUES

(

'1850225236511',

'Stoica',

'Mihai',

'0743234100',

'mihai.stoica@yahoo.com',

42,

'AQ40'

);

INSERT INTO SOFERI VALUES

(

'1760218246712',

'Petrescu',

'Alin-Georgian',

'0740233110',

NULL,

170,

'T56W'

);

*AMENZI*:

INSERT INTO AMENZI VALUES

(

'R48J',

2400,

NULL,

TO\_DATE('31-12-2018','DD-MM-YYYY')

);

INSERT INTO AMENZI VALUES

(

'G45A',

2400,

NULL,

TO\_DATE('23-04-2019','DD-MM-YYYY')

);

O imagine care conține captură de ecran, monitor, ecran

Descriere generată automat