

Universitatea din București
Facultatea de Matematică și Informatică
Calculatoare și Tehnologia Informației

Baze de Date

Profesor coordonator:

Vasile Silviu-Laurențiu

Student:

Albei Liviu-Andrei
Grupa 261

2021

Magazin de ceasuri de mână

Profesor coordonator:

Vasile Silviu-Laurențiu

Student:

Albei Liviu-Andrei
Grupa 261

2021

CUPRINS

1. Prezentarea modelului și a regulilor acestuia.....	5
2. Regulile modelului.....	5
3. Diagrama Entitate-Relație.....	6
3.1. Reprezentarea diagramei	6
3.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților.....	7
3.2.1. Descrierea entităților, atributelor și a cheilor	7
3.2.2. Descrierea relațiilor și a cardinalităților	10
4. Diagrama Conceptuală.....	13
4.1. Reprezentare diagramă	13
4.2. Descrierea constrângerilor de integritate.....	14
4.2.1. Tabelul LOCAȚIE	14
4.2.2. Tabelul MAGAZIN	14
4.2.3. Tabelul DETINE_CEAS	15
4.2.4. Tabelul CEAS.....	15
4.2.5. Tabelul FURNIZEAZA_CEAS.....	15
4.2.6. Tabelul BRAND_PRODUCATOR.....	15
4.2.7. Tabelul PLASAT.....	16
4.2.8. Tabelul COMANDA	16
4.2.9. Tabelul FIRMA_DE_LIVRAT	16
4.2.10. Tabelul CLIENT.....	16
4.2.11. Tabelul ORAS	17
4.3. Scheme relaționale	17
4.3.1. Schemele relaționale.....	17
4.3.2. Descrierea constângerilor ON DELETE	18
5. Scriptul SQL	19
5.1. Introducere	19
5.2. Etapa de DROP	19

5.3. Crearea tabelelor și a constrângerilor inclusiv	20
5.3.1. Crearea tabelului Magazin	20
5.3.3. Crearea tabelului Brand_producator	20
5.3.4. Crearea tabelului Oras	21
5.3.5. Crearea tabelului Client	21
5.3.6. Crearea tabelului Comanda	21
5.3.7. Crearea tabelului Firma_de_Livrat	22
5.3.8. Crearea tabelului Plasat	22
5.3.9. Crearea tabelului Locatie	22
5.3.10. Crearea tabelului Furnizeaza_Ceas	22
5.3.11. Crearea tabelului Detine_ceas	22
5.4. Introducerea datelor în baza de date	23
5.4.1. Introducerea datelor în tabelul Magazin	23
5.4.2. Introducerea datelor în tabelul Locatie	23
5.4.3. Introducerea datelor în tabelul Ceas	23
5.4.4. Introducerea datelor în tabelul Detine_ceas	23
5.4.5. Introducerea datelor în tabelul Brand_producator	24
5.4.6. Introducerea datelor în tabelul Furnizeaza_ceas	24
5.4.7. Introducerea datelor în tabelul Oras	24
5.4.8. Introducerea datelor în tabelul Client	24
5.4.9. Introducerea datelor în tabelul Comanda	24
5.4.10. Introducerea datelor în tabelul Firma_de_livrat	25
5.4.11. Introducerea datelor în tabelul Plasat	25

1. Prezentarea modelului și a regulilor acestuia

Tema aleasă pentru proiectul la disciplina “Baze de Date” este “Magazine de ceasuri de mână”. Consider faptul că proiectul face referire la niște magazine fictive, care nu există, pe care le voi inventa.

Bazele de date fac parte din viața noastră cotidiană fără să ne dăm seama neapărat, același lucru se aplică și în cazul magazinelor. Fără aceste baze de date, aproape nimic nu poate funcționa(mai puțin pentru un călugăr, de exemplu).

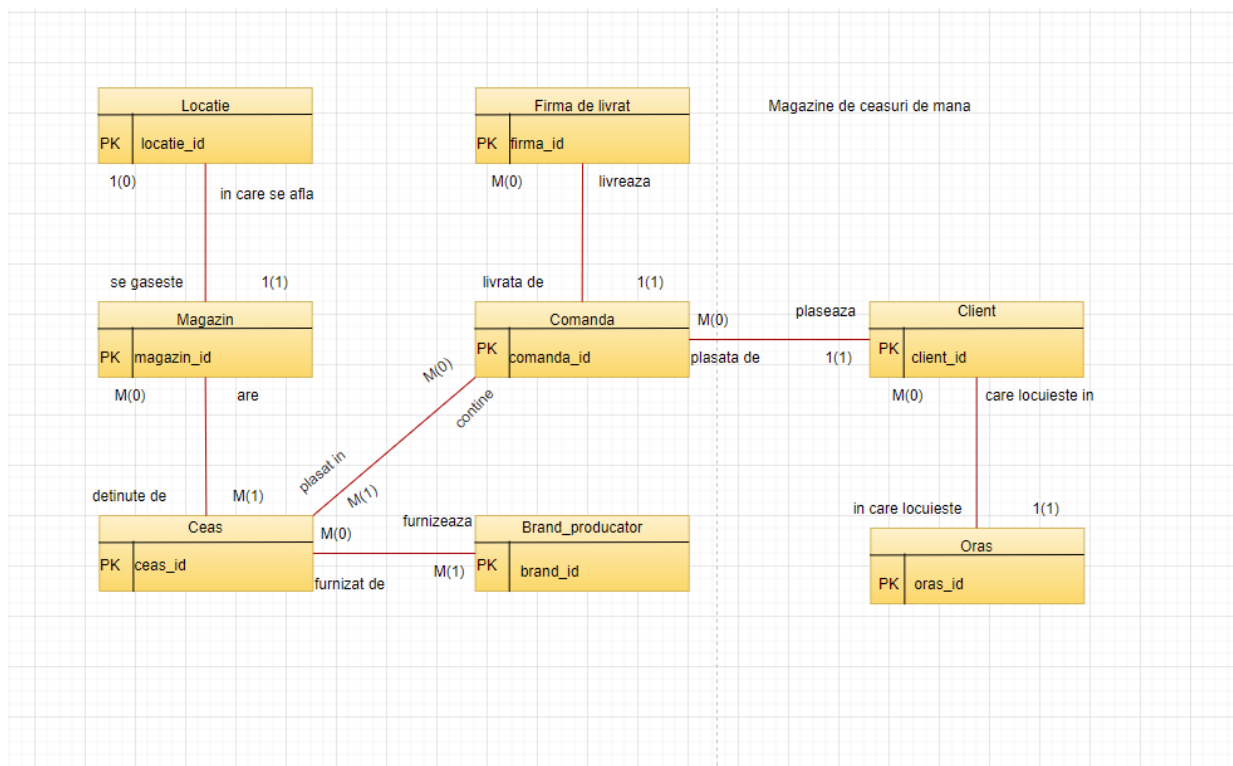
Timpul este foarte prețios, dar într-un mod diferit pentru fiecare persoană. Astfel, un ceas ne poate ajuta să nu pierdem noțiunea timpului și să îl planificăm pe acesta cât mai bine. Datorită acestui lucru dar și datorită pasiunii mele pentru ceasuri, am ales acest proiect de implementat.

2. Regulile modelului

- un magazin se găsește într-o locație fizică sau poate să fie online;
- un magazin are în stoc mai multe ceasuri(modele) și cel puțin unul;
- un brand poate să furnizeze un model de ceas sau mai multe;
- într-o comandă este plasat cel puțin un ceas;
- o comandă sau mai multe sunt livrate de o firmă de livrat, dar firma poate respectivă poate să nu livreze nicio comandă;
- un client plasează cel puțin o comandă sau niciuna;
- un client locuiește într-un singur oraș;

3. Diagrama Entitate-Relație

3.1. Reprezentarea diagramei



3.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

3.2.1. Descrierea entităților, atributelor și a cheilor

3.2.1.1. Tabelul LOCAȚIE

Tabelul LOCAȚIE stochează datele elementare despre locațiile în care se află magazinele din baza de date. Structura tabelului LOCAȚIE este:

Cheie	Denumire atribut	Descriere	Tip date
PK	locatie_id	identificatorul unic al locației	VARCHAR2
FK	magazin_id	identificatorul unic al magazinului care se află în locație	VARCHAR2
	oraș	adresă	VARCHAR2
	stradă	adresă	VARCHAR2

3.2.1.2. Tabelul MAGAZIN

Tabelul MAGAZIN memorează informațiile generale ale unui magazin, precum numele și orarul de funcționare. Fiecare magazin are un cod unic de indentificare. Structura tabelului MAGAZIN este:

Cheie	Denumire atribut	Descriere	Tip date
PK	magazin_id	identificatorul unic al magazinului	VARCHAR2
	nume	numele magazinului	VARCHAR2
	program	programul de lucru/orarul	VARCHAR2
	nr_telefon	numărul de telefon/contact	VARCHAR2
	email	contact	VARCHAR2
	site_web	pentru cumpărături online	VARCHAR2

3.2.1.3. Tabelul CEAS

Tabelul CEAS stochează informațiile generale, specifice ale unui ceas, precum modelul, prețul sau tipul mecanismului. Structura tabelului CEAS este:

Cheie	Denumire atribut	Descriere	Tip date
PK	ceas_id	identificatorul unic al fiecărui ceas	VARCHAR2
	nume	numele ceasului	VARCHAR2
	model	modelul ceasului	VARCHAR2
	preț	prețul de achiziție	NUMBER
	tip_mecanism	quartz/mecanic	BOOL
	tip_curea	din ce este făcută cureaua	VARCHAR2
	dimensiune	ce dimensiune are geamul	NUMBER
	culoare	ce culoare are ceasul	VARCHAR2
	afișaj	digital/analog	BOOL
	an_fabricație	în ce an a fost construit modelul	NUMBER

3.2.1.4. Tabelul Brand_producator

Tabelul BRAND_PRODUCĂTOR stochează informațiile despre brand-ul ce furnizează ceasurile, precum și țara lor de origine. Structura tabelului Brand_producator este:

Cheie	Denumire atribut	Descriere	Tip date
PK	brand_id	identificatorul unic al brandului	VARCHAR2
	nume	numele producătorului	VARCHAR2
	tara_origine	adresa	VARCHAR2

3.2.1.5. Tabelul Comandă

Tabelul COMANDĂ memorează informațiile în legătură cu o comanda plasată de către un client. Structura tabelului COMANDĂ este:

Cheie	Denumire atribut	Descriere	Tip date
-------	------------------	-----------	----------

PK	comanda_id	identificatorul uni al comenzii	VARCHAR2
	data_amplasare	data la care a fost amplasată comanda	DATE
	data_onorare_comanda	data la care se preconizează onorarea acesteia	DATE
FK	client_id	identificatorul unic al clientului ce a plasat comanda	VARCHAR2

3.2.1.6. Tabelul Firmă_de_livrat

Tabelul FIRMĂ_DE_LIVRAT stochează informațiile despre firma de livrat ce urmează să livreze comenzile amplasate de clienți. Structura tabelului FIRMĂ_DE_LIVRAT este:

Cheie	Denumire atribut	Descriere	Tip date
PK	firma_id	identificatorul unic al firmei	VARCHAR2
	nume	numele firmei	VARCHAR2
FK	comanda_id	identificatorul unic al comenzii ce urmează să fie livrat de către firmă	VARCHAR2

3.2.1.7. Tabelul Client

Tabelul CLIENT stochează informații despre clienții ce au plasat o comandă la unul din magazinele prezente în baza de date. Informațiile sunt generale, în legătura cu numele, numărul de telefon sau email-ul. Structura tabelului CLIENT este:

Cheie	Denumire atribut	Descriere	Tip date
PK	client_id	identificatorul unic al clientului	VARCHAR2
	nume	numele clientului	VARCHAR2
	prenume	prenumele	VARCHAR2

	nr_telefon	numărul de telefon al acestuia	VARCHAR2
	email	email-ul clientului	VARCHAR2
	strada	adresa	VARCHAR2
FK	oras_id	identificatorul unic al oraşului unde clientul doreşte să ajungă comanda	VARCHAR2

3.2.1.8. Tabelul ORAŞ

Tabelul ORAŞ stochează informaţiile necesare în legătură cu locaţia în care urmează să fie livrată comanda. Informaţiile ce se găsesc în acest tabel sunt ţara sau strada. Structura tabelului ORAŞ este:

Cheie	Denumire atribut	Descriere	Tip date
PK	oras_id	identificatorul unic al oraşului	VARCHAR2
	nume	numele oraşului	VARCHAR2
	tara	ţara în care se găseşte oraşul	VARCHAR2
	cod_postal	adresa	VARCHAR2

3.2.2. Descrierea relaţiilor şi a cardinalităţilor

3.2.2.1. LOCAŢIE – MAGAZIN

Relaţia: MAGAZINELE se găsesc/se află într-o LOCAŢIE specifică.

Cele două entităţi, “MAGAZIN” şi “LOCAŢIE”, sunt într-o relaţie de tipul “one-to-one”. Un magazin se poate afla într-o locaţie sau poate să nu aibă nicio locaţie(magazin de tip online) iar o locaţie aparţine unui singur magazin.

Câte LOCAŢII poate să aibă un MAGAZIN? => 1

Câte MAGAZINE pot fi într-o LOCAŢIE? => 1

Câte MAGAZINE trebuie să depindă de o LOCAŢIE? => 0

Câte MAGAZINE trebuie să apară într-o LOCAŢIE? => 1

3.2.2.2. MAGAZIN – CEAS

Relația: CEASURILE sunt deținute și vândute mai departe de către MAGAZINE.

Cele două entități, “MAGAZIN” și “CEAS”, sunt într-o relație de tipul “many-to-many”.

ATENȚIE! Datorită faptului că cele două entități se află într-o relație de tipul “many-to-many”, o să avem nevoie de un tabel intermediar/asociativ între cele două. Astfel, rezultă tabelul “DEȚINE_CEAS” ce va face legătura între tabelele “MAGAZIN” și “CEAS”.

Astfel, un magazin poate deține un model de ceas sau mai multe iar un model de ceas poate fi deținut de mai multe magazine și cel puțin într-unul.

Câte CEASURI se găsesc într-un MAGAZIN? => Multe

Câte MAGAZINE pot să aibă același CEAS? => Multe

Câte CEASURI trebuie să dețină un MAGAZIN? => 1 (Minim)

De câte MAGAZINE trebuie să depindă un CEAS? => 0

3.2.2.3. CEAS – BRAND_PRODUCATOR

Relația: CEASURILE sunt produse de către un BRAND_PRODUCATOR.

Cele două entități, “CEAS” și “BRAND_PRODUCATOR”, sunt într-o relație de tipul “many-to-many”. Așadar, apare tabela intermediară “FURNIZEAZA_CEAS”. Un ceas poate să fie produs de către un brand producător iar un brand producător furnizează cel puțin un ceas.

Câte CEASURI sunt furnizate de un BRAND? => Multe

Câte BRAND-uri furnizează CEAS-uri? => Multe

De câte BRAND-uri depinde un CEAS? => 1

Câte CEAS-uri trebuie să furnizeze un BRAND? => 0

3.2.2.4. CEAS – COMANDA

Relația: CEASUL/CEASURILE sunt amplasate într-o COMANDĂ/ mai multe.

Cele două entități, “CEAS” și “COMANDĂ”, sunt într-o relație de tipul “many-to-many”.

Așadar, apare tabela “PLASAT”. Într-o comandă se găsește cel puțin un ceas iar un ceas poate să aparțină de mai multe comenzi.

Câte CEASURI trebuie să conțină o COMANDĂ? => Multe

Câte COMENZI trebuie să dețină CEASURI? => Multe

De câte CEASURI depinde o COMANDĂ? => 1 (Minim)

Câte COMENZI nu dețin CEAS-uri? => 0

3.2.2.5. FIRMĂ_DE_LIVRAT – COMANDĂ

Relația: O FIRMĂ DE LIVRAT onorează o COMANDĂ sau niciuna.

Cele două entități, “FIRMĂ_DE_LIVRAT” și “COMANDĂ”, sunt într-o relație de tipul “one-to-many”. O firmă de livrat poate să livreze o anumită comandă dar o comandă trebuie să fie livrată neapărat de către o firmă de livrat.

Câte COMENZI pot avea aceeași FIRMĂ DE LIVRAT? => Multe

De câte FIRME DE LIVRAT poate fi livrată o COMANDĂ? => 1

Câte COMENZI trebuie să depindă de mai multe FIRME? => 0

Câte FIRME DE LIVRAT depind de o COMANDĂ? => 1

3.2.2.6. COMANDĂ – CLIENT

Relația: Un CLIENT amplacează cel puțin o COMANDĂ sau niciuna.

Cele două entități, “COMANDĂ” și “CLIENT”, sunt într-o relație de tipul “one-to-many”.

Astfel, un client poate plasa cel puțin o comandă sau niciuna dar o comandă trebuie să aparțină de un singur client.

Câte COMENZI poate să aibă un CLIENT? => Multe

Câți CLIENTI au aceeași COMANDĂ? => 1

Câte COMENZI trebuie să aibă același CLIENT? => 0

Câți CLIENTI trebuie să aibă o COMANDĂ? => 1

3.2.2.7. CLIENT – ORAȘ

Relația: CLIENTUL locuiește într-un ORAȘ (unde se livrează comanda).

Cele două entități, “CLIENT” și “ORAȘ”, sunt într-o relație de tipul “one-to-many”. Un client locuiește într-un singur oraș (în acel oraș/adresă va ajunge comanda, nu neapărat locuiește acolo) iar într-un oraș pot să locuiască mai mulți clienți.

Câți CLIENTI pot să locuiască în același ORAȘ? => Mulți

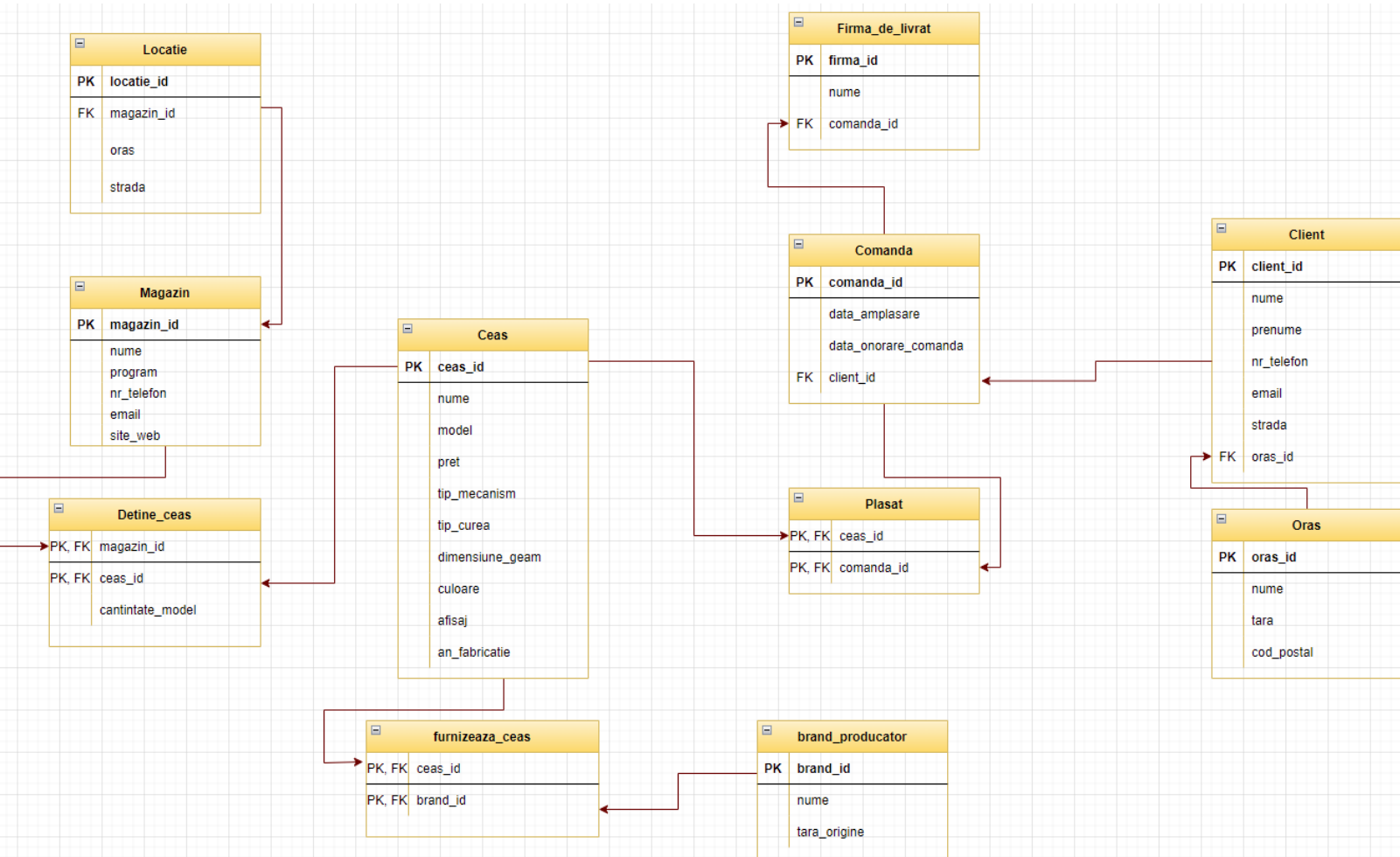
Câte ORAȘE au aceiași CLIENTI? => 1

Câți CLIEȚI trebuie să locuiască în același ORAȘ? => 0

Câte ORAȘE trebuie să fie locuite de un CLIENT? => 1.

4. Diagrama Conceptuală

4.1. Reprezentare diagramă



4.2. Descrierea constrângerilor de integritate

4.2.1. Tabelul LOCAȚIE

- **Constrângeri de tip PRIMARY KEY**
 - LOCATIE_locatieId_PK, pentru atributul locatie_id;
- **Constrângeri de tip FOREIGN KEY**
 - LOCATIE_magazinId_FK, pentru atributul magazin_id;
 - *face legătura cu tabelul MAGAZIN(magazin_id).*
- **Constrângeri de tip NOT NULL**
 - LOCATIE_oras_NN, pentru atributul oras;
 - *orașul trebuie să aibă un nume.*
 - LOCATIE_strada_NN, pentru atributul strada;
 - *strada trebuie să aibă un nume.*

4.2.2. Tabelul MAGAZIN

- **Constrângeri de tip PRIMARY KEY**
 - MAGAZIN_magazinId_PK, pentru atributul magazin_id;
- **Constrângeri de tip NOT NULL**
 - MAGAZIN_nume_NN, pentru atributul nume;
 - MAGAZIN_program_NN, pentru atributul program;
 - MAGAZIN_nrTlf_NN, pentru atributul nr_telefon;
 - MAGAZIN_email_NN, pentru atributul email;
 - MAGAZIN_siteWeb_NN, pentru atributul site_web.
- **Constrângeri de tip UNIQUE**
 - MAGAZIN_nrTlf_U, pentru atributul nr_telefon;
 - *nu pot exista două numere de telefon identice.*
 - MAGAZIN_email_U, pentru atributul email;
 - *nu pot exista două email-uri identice.*
 - MAGAZIN_siteWeb_U, pentru atributul site_web.
 - *nu pot exista două site-uri web identice.*
- **Constrângeri de tip CHECK**
 - MAGAZIN_nrTlf_C, pentru atributul nr_telefon.
 - *prin funcția LENGTH(), verificăm dacă numărul de telefon are 10 cifre.*

4.2.3. Tabelul DETINE_CEAS

- **Constrângeri de tip PRIMARY KEY**
 - DETINEceas_PK, pentru attributele (magazin_id, ceas_id).
- **Constrângeri de tip FOREIGN KEY**
 - DETINEceas_magazinId_FK, pentru atributul magazin_id;
 - *face legătura cu tabelul MAGAZIN (magazin_id).*
 - DETINEceas_ceasId_FK, pentru atributul ceas_id;
 - *face legătura cu tabelul CEAS (ceas_id).*

4.2.4. Tabelul CEAS

- **Constrângeri de tip PRIMARY KEY**
 - CEAS_ceasId_PK, pentru atributul ceas_id.
- **Constrângeri de tip NOT NULL**
 - CEAS_nume_NN, pentru atributul nume;
 - CEAS_model_NN, pentru atributul model;
 - CEAS_pret_NN, pentru atributul pret;
 - CEAS_tipMecanism_NN, pentru atributul tip_mecanism;
 - CEAS_tipCurea_NN, pentru atributul tip_curea;
 - CEAS_dimensiuneGeam_NN, pentru atributul dimensiune_geam;
 - CEAS_culoare_NN, pentru atributul culoare;
 - CEAS_afisaj_NN, pentru atributul afisaj;
 - CEAS_anFabricatie_NN, pentru atributul an_fabricatie.

4.2.5. Tabelul FURNIZEAZA_CEAS

- **Constrângeri de tip PRIMARY KEY**
 - FURNIZEAZA_CEAS_PK, pentru attributele (ceas_id, brand_id).
- **Constrângeri de tip FOREIGN KEY**
 - FURNIZEAZA_CEAS_ceasId_FK, pentru atributul ceas_id;
 - *face legătura cu tabelul CEAS.*
 - FURNIZEAZA_CEAS_brandId_FK, pentru atributul brand_id;
 - *face legătura cu tabelul BRAND_PRODUCATOR.*

4.2.6. Tabelul BRAND_PRODUCATOR

- **Constrângeri de tip PRIMARY KEY**
 - BP_brandId_PK, pentru atributul brand_id.
- **Constrângeri de tip NOT NULL**
 - BP_nume_NN, pentru atributul nume;
 - BP_taraOrigine_NN, pentru atributul tara_origine.

4.2.7. Tabelul PLASAT

- **Constrângeri de tip PRIMARY KEY**
 - PLASAT_PK, pentru attributele (ceas_id, comanda_id).
- **Constrângeri de tip FOREIGN KEY**
 - PLASAT_ceasId_FK, pentru atributul ceas_id;
 - *face legătura cu tabelul CEAS (ceas_id).*
 - PLASAT_comandaId_FK, pentru atributul comanda_id;
 - *face legătura cu tabelul COMANDA (comanda_id).*

4.2.8. Tabelul COMANDA

- **Constrângeri de tip PRIMARY KEY**
 - COMANDA_comandaId_PK, pentru atributul comanda_id.
- **Constrângeri de tip FOREIGN KEY**
 - COMANDA_clientId_FK, pentru atributul client_id.
- **Constrângeri de tip NOT NULL**
 - COMANDA_dataAmplasare_NN, pentru atributul data_amplasare;
 - COMANDA_dataOnorare_NN, pentru atributul data_onorare_comanda.

4.2.9. Tabelul FIRMA_DE_LIVRAT

- **Constrângeri de tip PRIMARY KEY**
 - FIRMALIVRAT_firmaId_PK, pentru atributul firma_id.
- **Constrângeri de tip FOREIGN KEY**
 - FIRMALIVRAT_comandaId_FK, pentru atributul comanda_id
 - *face legătura cu tabelul COMANDA (comanda_id).*
- **Constrângeri de tip NOT NULL**
 - FIRMALIVRAT_numa_NN, pentru atributul nume.

4.2.10. Tabelul CLIENT

- **Constrângeri de tip PRIMARY KEY**
 - CLIENT_clientId_PK, pentru atributul client_id.
- **Constrângeri de tip FOREIGN KEY**
 - CLIENT_orasId_PK, pentru atributul oras_id.
 - *face legătura cu tabelul ORAS (oras_id).*
- **Constrângeri de tip NOT NULL**
 - CLIENT_numa_NN, pentru atributul nume;
 - CLIENT_prenume_NN, pentru atributul prenume;
 - CLIENT_nrTlf_NN, pentru atributul nr_telefon;
 - CLIENT_email_NN, pentru atributul email;
 - CLIENT_strada_NN, pentru atributul strada.

- **Constrângeri de tip CHECK**
 - CLIENT_nrTlf_U, pentru atributul nr_telefon.
 - *numărul de telefon trebuie să aibă 10 cifre.*
- **Constrângeri de tip UNIQUE**
 - CLIENT_nrTlf_C, pentru atributul nr_telefon;
 - numărul de telefon este unic.
 - CLIENT_email_C, pentru atributul email.
 - email-ul este unic.

4.2.11. Tabelul ORAS

- **Constrângeri de tip PRIMARY KEY**
 - ORAS_orasId_PK, pentru atributul oras_id.
- **Constrângeri de tip NOT NULL**
 - ORAS_nume_NN, pentru atributul nume.
 - ORAS_tara_NN, pentru atributul tara.
 - ORAS_codPostal_NN, pentru atributul cod_postal.
- **Constrângeri de tip UNIQUE**
 - ORAS_codPostal_U, pentru atributul cod_posta.
 - codul poștal al fiecărui oraș este unic.

4.3. Scheme relaționale

4.3.1. Schemele relaționale

Schemele relaționale atașate diagramei conceptuale sunt:

- LOCATIE (locatie_id#, magazin_id <FK>, oras, strada);
- MAGAZIN (magazin_id#, nume, program, nr_telefon, email, site_web);
- DETINE_CEAS (magazin_id# <FK>, ceas_id# <FK>, cantitate_model);
- CEAS (ceas_id#, nume, model, pret, tip_mecanism, tip_curea, dimensiune_geam, culoare, afisaj, an_fabricatie);
- FURNIZEAZA_CEAS (ceas_id# <FK>, brand_id# <FK>);
- BRAND_PRODUCATOR (brand_id#, nume, tara_origine);
- PLASAT (ceas_id# <FK>, comanda_id# <FK>);
- COMANDA (comanda_id#, data_amplasare, data_onorare_comanda, client_id <FK>);
- FIRMA_DE_LIVRAT (firma_id#, nume, comanda_id <FK>);

- CLIENT (client_id#, nume, prenume, nr_telefon, email, strada, oras_id <FK>);
- ORAS (oras_id#, nume, tara, cod_postal).

4.3.2. Descrierea constrângerilor ON DELETE

Există 10 constrângeri de tip ON DELETE specifice bazei mele de date, câte una pentru fiecare tip de constrângere de tip FOREIGN KEY. Acestea sunt:

- LOCATIE_magazinId_FK, **ON DELETE SET NULL**
 - când o locație este ștearsă din baza de date, magazinele care aveau acea locație nu vor fi șterse, locația va fi NULL astfel încât magazinul o să rămână de tip online.
- DETINEceas_magazinId_FK, **ON DELETE CASCADE**
 - dacă un magazin este șters, ceasurile pot fi șterse din tabel.
- DETINEceas_ceasId_FK, **ON DELETE CASCADE**
 - dacă un ceas este șters, magazinele o să rămână dar nu vor deține ceasul respectiv.
- FURNIZEAZA_CEAS_ceasId_FK, **ON DELETE CASCADE**
 - dacă un ceas este șters, brandul producător poate fi șters.
- FURNIZEAZA_CEAS_brandId_FK, **ON DELETE CASCADE**
 - dacă un brand este șters, ceasul produs de către acesta va fi șters.
- PLASAT_ceasId_FK, **ON DELETE CASCADE**
 - dacă un ceas este șters din comandă, comanda poate fi ștearsă dacă rămâne goală sau ramâne valabilă dacă mai conține produse.
- PLASAT_comandaId_FK, **ON DELETE CASCADE**
 - dacă o comandă este ștearsă, ceasurile rămân în stocul magazinului respectiv.
- COMANDA_clientId_FK, **ON DELETE CASCADE**
 - dacă este șters clientul, comanda acestuia urmează să fie ștearsă la rândul ei.
- FIRMALIVRAT_comandaId_FK, **ON DELETE CASCADE**

- dacă o comandă este ștearsă, firma de livrat poate livra altele.
- CLIENT_orasId_PK, [ON DELETE SET NULL](#)
 - dacă un oraș este șters, clientul va avea setat orașul drept NULL, urmând să ii fie livrată comanda în altul.

5. Scriptul SQL

5.1. Introducere

Scriptul SQL, salvat cu numele **SQLscript_Albei_Liviu-Andrei_261.sql**, a fost scris în aplicația Oracle SQL Developer.

Scriptul SQL rulează fără probleme, singura excepție fiind în prima etapă de rulare a acestuia deoarece atunci are loc procesul de DROP pentru tabele. Astfel, sunt afișate niște erori specifice datorită faptului ca ele nu există în primă instanță, dar după, crearea tabelor și popularea acestora se va face fără erori.

ATENȚIE! Script-ul prezentat în această documentație prezintă mici erori datorită neatenției când l-am scris inițial dar fișierul .sql conține și modificările corespunzătoare([ALTER TABLE](#)).

5.2. Etapa de DROP

[drop table](#) magazin;

[drop table](#) ceas;

[drop table](#) brand_producator;

[drop table](#) oras;

[drop table](#) client;

[drop table](#) comanda;

[drop table](#) firma_de_livrat;

[drop table](#) plasat;

[drop table](#) locatie;

[drop table](#) furnizeaza_ceas;

[drop table](#) detine_ceas;

5.3. Crearea tabelelor și a constrângerilor inclusiv

5.3.1. Crearea tabelului Magazin

`create table magazin`

```
( magazin_id number(5) constraint magazin_magazinId_PK primary key,  
  nume varchar2 (20) constraint magazin_nume_nn not null,  
  program varchar2 (20) constraint magazin_program_nn not null,  
  nr_telefon varchar2 (11) constraint magazin_nrTlf_nn not null constraint magazin_nrTlf_U  
unique constraint magazin_nrTlf_C check (length(nr_telefon)=10),  
  email varchar2 (50) constraint magazin_email_nn not null constraint magazin_email_u  
unique,  
  site_web varchar2 (50) constraint magazin_siteWeb_nn not null constraint  
magazin_siteWeb_U unique  
);
```

5.3.2. Crearea tabelului Ceas

`create table ceas`

```
( ceas_id number(6) constraint ceas_ceasId_PL primary key,  
  nume varchar2 (10) constraint ceas_nume_nn not null,  
  model varchar2 (15) constraint ceas_model_nn not null,  
  pret number (6,2) constraint ceas_pret_nn not null,  
  tip_mecanism char(8) constraint ceas_tipMecanism_nn not null constraint  
ceas_tipMecanism_c check (tip_mecanism in ('Quartz','Mecanic')),  
  tip_curea char (10) constraint ceas_tipCurea_nn not null constraint ceas_tipCurea_c check  
(tip_curea in ('Silicon','Material','Zale','Piele')),  
  dimensiune_geam number (2) constraint ceas_dimensiuneGeam_nn not null,  
  culoare varchar2 (20) constraint ceas_culoare_nn not null,  
  afisaj char (10) constraint ceas_afisaj_nn not null constraint ceas_afisaj_c check (afisaj in  
( 'Analogic','Digital')),  
  an_fabricatie number (4) constraint ceas_anFabricatie_nn not null  
);
```

5.3.3. Crearea tabelului Brand_producator

`create table Brand_producator`

```
( brand_id number (6) constraint bp_brandId_PK primary key,
  nome varchar2 (15) constraint bp_nome_nn not null,
  tara_origine varchar2 (20) constraint bp_taraOrigine_nn not null
);
```

5.3.4. Crearea tabelului Oras

```
create table Oras
```

```
( oras_id number (6) constraint oras_orasId_PK primary key,
  nome varchar2 (15) constraint oras_nome_nn not null,
  tara varchar2 (15) constraint oras_tara_nn not null,
  cod_postal number (6) constraint oras_codPostal_nn not null constraint oras_codPostal_u
unique
);
```

5.3.5. Crearea tabelului Client

```
create table Client
```

```
( client_id number(6) constraint client_clientId_PK primary key,
  nome varchar2 (20) constraint client_nome_nn not null,
  prenume varchar2 (20) constraint client_prenume_nn not null,
  nr_telefon number (11) constraint client_nrTlf_nn not null constraint client_nrTlf_c check
(length(nr_telefon)=10) constraint client_nrTlf_u unique,
  email varchar2 (30) constraint client_email_nn not null constraint client_email_u unique,
  strada varchar2 (50) constraint client_strada_nn not null,
  oras_id number (6) constraint client_orasId_fk references Oras (oras_id) on delete set null
);
```

5.3.6. Crearea tabelului Comanda

```
create table comanda
```

```
( comanda_id number (6) constraint comanda_comandaId_PK primary key,
  data_amplasare date constraint comanda_dataAmplasare_nn not null,
  data_onorare_comanda date constraint comanda_dataOnorare_nn not null
);
```

5.3.7. Crearea tabelului Firma_de_Livrat

create table firma_de_livrat

```
( firma_id number (6) constraint firmalivrat_firmaId_PK primary key,  
  nume varchar2 (15) constraint firmalivrat_nume_nn not null,  
  comanda_id number (6) constraint firmalivrat_comandaId_fk references Comanda  
(comanda_id) on delete cascade  
);
```

5.3.8. Crearea tabelului Plasat

create table plasat

```
( ceas_id number (6) constraint plasat_ceasId_FK references ceas (ceas_id) on delete cascade,  
  comanda_id number (6) constraint plasat_comandaId_FK references comanda (comanda_id)  
on delete cascade,  
  constraint plasat_pk primary key (ceas_id, comanda_id)  
);
```

5.3.9. Crearea tabelului Locatie

create table locatie

```
( locatie_id number (6) constraint locatie_locatieId_PK primary key,  
  oras varchar2 (15) constraint locatie_oras_nn not null,  
  strada varchar2 (15) constraint locatie_strada_nn not null,  
  magazin_id number(5) constraint locatie_magazinId_fk references Magazin (magazin_id) on  
delete set null  
);
```

5.3.10. Crearea tabelului Furnizeaza_Ceas

create table furnizeaza_ceas

```
( ceas_id number (6) constraint furnizeaza_ceas_ceasId_FK references ceas (ceas_id) on delete  
cascade,  
  brand_id number (6) constraint furnizeaza_ceas_comandaId_FK references brand_producator  
(brand_id) on delete cascade,  
  constraint furnizeaza_ceas_pk primary key (ceas_id, brand_id)  
);
```

5.3.11. Crearea tabelului Detine_ceas

create table detine_ceas

```
( magazin_id number (5) constraint detineceas_magazinId_FK references magazin
(magazin_id) on delete cascade,

ceas_id number (6) constraint detineceas_ceasId_FK references ceas (ceas_id) on delete
cascade,

constraint detineceas_pk primary key (magazin_id, ceas_id)
);
```

5.4. Introducerea datelor în baza de date

Voi introduce câte două exemple de populări pentru fiecare tabel, fișierul .sql urmând să aibă un număr de 5-6 populări pentru fiecare tabel.

5.4.1. Introducerea datelor în tabelul Magazin

```
insert into magazin
```

```
values(10000, 'Watchshop', '08-16', '0769123456', 'watchshop@gmail.com', 'watchshop.ro');
```

```
insert into magazin
```

```
values(10001, 'BBShop', '08-20', '0769456123', 'bbshop@gmail.com', 'bbshop.ro');
```

5.4.2. Introducerea datelor în tabelul Locatie

```
insert into locatie
```

```
values(250000, 'Ramnicu Sarat', 'B-dul 1 Dec1918', 10000);
```

```
insert into locatie
```

```
values(250001, 'Buzau', 'Principatelor', 10001);
```

5.4.3. Introducerea datelor în tabelul Ceas

```
insert into ceas
```

```
values(500000, 'Rolex', 'Mad Paris', 4350, 'Mecanic', 'Zale', 40, 'Gri', 'Analogic', 2020);
```

```
insert into ceas
```

```
values(500001, 'Fossil', 'Townsmen', 827.99, 'Mecanic', 'Zale', 42, 'Gri', 'Analogic', 2018);
```

5.4.4. Introducerea datelor în tabelul Detine_ceas

```
insert into detine_ceas
```

```
values(10000, 500000, 2);
```

```
insert into detine_ceas
```

```
values(10001, 500001, 4);
```

5.4.5. Introducerea datelor în tabelul Brand_producator

```
insert into brand_producator
```

```
values(800000, 'Rolex', 'Elvetia');
```

```
insert into brand_producator
```

```
values(800001, 'Fossil', 'Elvetia');
```

5.4.6. Introducerea datelor în tabelul Furnizeaza_cas

```
insert into furnizeaza_cas
```

```
values(500000, 800000);
```

```
insert into furnizeaza_cas
```

```
values(500001, 800001);
```

5.4.7. Introducerea datelor în tabelul Oras

```
insert into oras
```

```
values (900000, 'Bucuresti', 'Romania', 400000);
```

```
insert into oras
```

```
values (900001, 'Cluj-Napoca', 'Romania', 400001);
```

5.4.8. Introducerea datelor în tabelul Client

```
insert into client
```

```
values(150000, 'Serban', 'Cosmin', 'jamboc92@gmail.com', 'Strada Scolii', 900000,  
'0727100200');
```

```
insert into client
```

```
values(150001, 'Sirbu', 'Ionut', 'alexs2001@gmail.com', 'B-dul Iuliu Maniu', 900001,  
'0727200300');
```

5.4.9. Introducerea datelor în tabelul Comanda

```
insert into comanda
```

```
values(850000, '15-JAN-22', '17-JAN-22', 150000);
```


insert into comanda

values(850001, '10-JAN-22', '11-JAN-22', 150001);

5.4.10. Introducerea datelor în tabelul Firma_de_livrat

insert into firma_de_livrat

values (450000, 'FanCourier RO', 850000);

insert into firma_de_livrat

values (450001, 'NemoExpress', 850001);

5.4.11. Introducerea datelor în tabelul Plasa

insert into plasa

values (500000, 850000);

insert into plasa

values (500001, 850001);

NOTĂ! Id-urile prezentate (de exemplu 500000 sau 850000) sunt puse cât mai mari din simplă preferință iar în tabelul Comanda, formatul prezentat DD-MON-YY este cel implicit pe care îl aveam setat SQL Developer-ului.

Pentru constrângeri am aplicat o metodă simplă de numire:

NumeTABEL_NumeColoana_TipCONSTRÂNGERE

Exemplu:

NumeTABEL -> Magazin

NumeColoana -> MagazinId

TipCONSTRÂNGERE -> PK

Legendă:

- **PK** – Primary key;
- **NN** – Not Null;
- **FK** – Foreign key;
- **U** – Unique;
- **C** – Check.